Trabajo de Visión por Computador

TRABAJO-1: Convolución y Derivadas

Implementación

VALORACIÓN TOTAL: 8 (+4) puntos

Fecha de entrega: 30 de octubre de 2021

Informe a presentar

Para este trabajo, como para los demás proyectos, se debe presentar un informe con las valoraciones y decisiones adoptadas en cada uno de los apartados. También deberá incluirse una valoración sobre la calidad de los resultados encontrados. Dicho informe se presentará de modo separado, como un fichero PDF, en caso de que se opte por NO entregar un Notebook ejecutable en Google Colab incluyendo todas las valoraciones, decisiones y discusión de resultados. Recuérdese que la entrega de código sin informe explicativo no puntúa.

Normas de la entrega de Prácticas:

EL INCUMPLIMIENTO DE ESTAS NORMAS SIGNIFICA PÉRDIDA DIRECTA DE 1 PUNTO CADA VEZ QUE SE DETECTE UN INCUMPLIMIENTO.

1. El código se debe estructurar como un único fichero (.py o .ipynb) que irá llamando de forma secuencial a distintas funciones, una por cada apartado de la práctica.
2. El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
3. Si hay más de un fichero en la entrega (por ejemplo, en caso de entregar separadamente código y memoria en PDF), todos los ficheros se incluirán dentro de un fichero zip/rar.
4. SOLO ENTREGAR EL CODIGO FUENTE. NO INCLUIR IMÁGENES.
5. Los path que se usen en la lectura de imágenes o cualquier fichero de entrada debe ser siempre "imagenes/nombre_fichero"
6. Todos los resultados numéricos serán mostrados por pantalla. No escribir nada en el disco.
7. La práctica deberá poder ser ejecutada de principio a fin sin necesidad de ninguna selección de opciones. Para ello se deberán fijar los parámetros por defecto que se consideren óptimos.
8. Poner puntos de parada para mostrar imágenes o datos por consola y el final de cada apartado

Forma de entrega: A través de PRADO.

Este trabajo de implementación tiene como objetivo principal aprender a implementar filtros de convolución y, particularmente, el cálculo de las derivadas de una imagen. El documento pdf sobre

〔1,2,1〕 que sigma representa?

Hacer una función que acepta dos entradas alternativas, solo una de ellas funciona, o le damos la longitud o le damos el sigma y calculamos que longitud le corresponde.

Gaussian_discretization disponible en PRADO (https://pradogrado2122.ugr.es/mod/folder/view.php?id=96518) puede resultar de gran utilidad para la implementación del ejercicio 1.A.

1.- USANDO SOLO FUNCIONES BÁSICAS DE OPENCV, escribir funciones Python que implementen de forma eficiente el cálculo de los siguientes puntos (4 puntos):

A. Considere la función Gaussiana 1D de media 0 y desviación típica σ. Calcular las máscaras discretas 1D de la función Gaussiana, la derivada de la Gaussiana y la segunda derivada de la Gaussiana. La función implementada debe considerar que tanto el valor de σ como el tamaño de la máscara son posibles entradas alternativas a la función. Represente en ejes cartesianos las máscaras obtenidas como funciones 1D y compare sus formas con las máscaras dadas por la función de OpenCV *getDerivKernels* para los mismos tamaños de máscara. Use los tamaños 5, 7 y 9.

B. Calcule las máscaras discretas 1D de longitud 5 y 7 tanto de alisamiento como de derivada de primer orden generadas a partir de la aproximación binomial de la Gaussiana y la máscara de derivada de longitud 3.. Ejecute la función *cv2.getDerivKernels(0,1,9)*. Observe los vectores de salida y compárelos con los previamente calculados, ¿Qué relación hay? ¿Qué conclusiones extrae sobre la aproximación de OpenCV al cálculo de las máscaras de derivadas de primer orden?

C. Implementar la convolución de una imagen con una máscara 2D de dimensiones inferior a la imagen suponiendo la propiedad de separabilidad de la máscara e imponiendo bordes reflejados. La entrada a la función serán las máscaras 1D descomposición de la máscara 2D. Comparar su funcionamiento con

   a. La salida de *cv2.GaussianBlur* para una máscara de entrada Gaussiana con iguales parámetros en ambos casos. Mostrar ambos resultados.
   b. Usar las máscaras del punto A para calcular la imágenes derivadas respecto de x e y de una imagen dada. Mostrar los resultados.

D. Usar las implementaciones del punto A para calcular máscaras normalizadas de la Laplaciana de una Gaussiana. Mostrar ejemplos de funcionamiento usando bordes reflejados y dos valores de sigma (1 y 3) con alguna imagen. Compare los resultados con la salida dada por OpenCV con la función Laplaciana. Discuta pros y contras.

La convolucion de una mascara 1d GAUSSIANA es simetrica, si es derivada no serian iguales los dos vectores. Por tanto, deben de ser dos entradas

2.- IMPLEMENTAR funciones para las siguientes tareas (4 puntos). Usar funciones implementadas en el ejercicio 1. No se permite el uso de funciones específicas de OpenCV.

(1.5 puntos) Una función que genere una representación en pirámide Gaussiana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando bordes replicados o reflejados, y justificar la elección de los parámetros. Mostrar todos los niveles de la pirámide en una única imagen. Comparar los resultados con la pirámide obtenida usando las funciones OpenCV.

(1.5 puntos) Una función que genere una representación en pirámide Laplaciana de 4 niveles de una imagen. Mostrar ejemplos de funcionamiento usando bordes. Mostrar todos los niveles de la pirámide en una única imagen. Comparar los resultados con la pirámide obtenida usando las funciones OpenCV.

(1 punto) Verificar el correcto funcionamiento de la pirámide Laplaciana por medio de mostrar su capacidad para recuperar la imagen original. Se debe mostrar el error obtenido en la aproximación, en términos de distancia Euclídea entre los niveles de gris de la imagen original y la imagen reconstruida.

BONUS: Solo se tendrán en cuenta los bonus si se ha logrado al menos el 75% de los puntos en la parte obligatoria. **Imágenes Híbridas** (SIGGRAPH 2006 paper by Oliva, Torralba, and Schyns: https://stanford.edu/class/ee367/reading/OlivaTorralb_Hybrid_Siggraph06.pdf). (4 puntos)

Mezclando adecuadamente una parte de las frecuencias altas de una imagen con una parte de las frecuencias bajas de otra imagen, obtenemos una imagen híbrida que admite distintas interpretaciones a distintas distancias (ver http://olivalab.mit.edu/hybridimage.htm).

Para seleccionar la parte de frecuencias altas y bajas que nos quedamos de cada una de las imágenes usaremos el parámetro sigma del núcleo/máscara de alisamiento Gaussiano. A mayor valor de sigma mayor eliminación de altas frecuencias en la imagen convolucionada. Para una buena implementación, se recomienda elegir dicho valor de forma separada para cada una de las dos imágenes (véanse las recomendaciones dadas en el paper de Oliva et al.). Recuérdese que puede haber una región de posibles valores.

1. (2 puntos) Implementar una función que genere las imágenes de baja y alta frecuencia a partir de las parejas de imágenes (solo en la versión de imágenes de gris) .La imagen a usar para las bajas frecuencias y las altas respectivamente es importante para la calidad final. El valor de sigma (frecuencia de corte) más adecuado para cada imagen en cada pareja hay que encontrarlo por experimentación de prueba y error (el artículo propone una técnica)
    a. Escribir una función que muestre las tres imágenes (alta, baja e híbrida) en una misma ventana. Recordar que las imágenes después de una convolución contienen número flotantes que pueden ser positivos y negativos.
    b. Realizar la composición con, al menos, 3 de las parejas de imágenes.
    c. Construir pirámides Gaussianas de, al menos, 4 niveles con las imágenes resultado. Explicar el efecto que se observa.
2. (1 punto) Realizar todas las parejas de imágenes híbridas en su formato a color. Solo se tendrá en cuenta si la versión de gris es correcta.
3. (1 punto) Realizar una imagen híbrida con, al menos, una pareja de imágenes de su elección que hayan sido extraídas de imágenes más grandes. Justifique la elección y todos los pasos que realiza.