

Original



Imagen1



Bordes1



Resultante



Observaciones:

1. Imagen con filtro Gaussiano:
 - a. Después de la aplicación de la máscara de 5x5 la imagen si se nota que está un poco más borrosa, sin embargo se retienen sus características principales y es posible la distinción de sus elementos.
2. Imagen de bordes con el operador Laplaciano:
 - a. Los bordes del edificio central fueron captadas casi a la perfección.
 - b. El pasto con sus reflejos causo que el operador viera mucha diferencia entre lo verde, negro y blanco causando que se viera una textura de raíces o "venosa".
3. Imagen resultante:
 - a. La imagen resultante muestra cierto tipo de "relieve", podemos ver que el pasto y las plantas están "tapando" más fuertemente el fondo que en la imagen original donde todo se ve muy "plano". Es como que la parte de los edificios les hicieran un "send to back" y la parte vegetal un "Bring to front".
 - b. El reflejo y lo blanco de las plantas se muestran con más intensidad.
 - c. Estos efectos se lograron de restarle a la imagen con filtro Gaussiano la imagen con operador Laplaciano, ya que al aplicar esta resta los bordes resaltados que tiene la imagen con operador Laplaciano opacan los de la imagen con filtro Gaussiano, esto crea el efecto de nitidez que se nota en los bordes, ya que al aplicar la resta los bordes se opacan y por lo tanto se marcan más en la imagen resultante.
4. ¿Es suficiente los pasos que ejecutamos para hacer que una imagen sea más nítida?
 - a. La imagen resultante es decente, no se pierden detalles de la imagen original, de hecho, resalta algunos. Si nos acercamos lo suficiente, el efecto del filtro Gaussiano (la que difumina la imagen) sí se nota. A la vez, el operador Laplaciano también agrega ciertos artefactos a la

imagen final. Como síntesis, sí pensamos que podrían servir un par de pasos más para eliminar dichos detalles para tener una imagen resultante casi perfecta y nítida.

Métodos que hacen que una imagen más nítida

1. De la librería PIL para Python, usar el método `ImageEnhance.Sharpness()` y el método `enhance()`. En el fondo usa un filtro 3x3 con la estructura $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.
2. Usar un filtro con la estructura: $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$.

Referencias

- Página oficial de documentación:
<https://pillow.readthedocs.io/en/stable/reference/ImageEnhance.html>
- D. .rs, “#004 How to smooth and sharpen an image in OpenCV?,” *Master Data Science*, 11-Nov-2020. [Online]. Available: <http://datahacker.rs/004-how-to-smooth-and-sharpen-an-image-in-opencv/>. [Accessed: 11-Nov-2020].