

# Escuela de Ciencias de la Computación

## Parcial

### CC462 Sistemas Concurrentes y Distribuidos

Se pide realizar un servidor y sus esclavos para que pueda realizar el proceso tratamiento de imágenes concurrentes y distribuidos, con operaciones de matrices que son las imágenes, con otras matrices que serían los kernels y estos dan resultados que serían nuevas imágenes con los siguientes procedimientos:

- Un servidor que recepcione varios archivos con extensión jpg.
- Cada archivo se transforme a escala de grises.
- Cada archivo en escala de grises multiplicarlo con un kernel y generar un nuevo archivo
- Posteriormente pasar cada archivo que se procesó con el kernel a un archivo jpg.

Pasos:

1. Un servidor que recepcione varios archivos con extensión jpg.

Dataset de imágenes que podrían usar. Tener en cuenta la cantidad y los tamaños de las imágenes que pueda soportar el cluster, ustedes podrán variar el tamaño y la cantidad de imágenes para procesar.

Por ejemplo

<https://www.kaggle.com/datasets?sortBy=relevance&group=featured&search=image>

Existen diferente maneras en java para leer imágenes.

Por ejemplo

<https://alvinalexander.com/blog/post/java/getting-rgb-values-for-each-pixel-in-image-using-java-buffered>

2. Cada archivo se transforme a escala de grises.

<https://en.wikipedia.org/wiki/Grayscale>

<https://onlinejpgtools.com/convert-jpg-to-grayscale>

Pueden pasar con las siguientes

- $Y' = 0.299R + 0.587G + 0.114B$
- $Y' = 0.2126R + 0.7152G + 0.0722B$
- $Y' = 0.2627R + 0.6780G + 0.0593B$
- $Y' = (R + G + B) / 3$

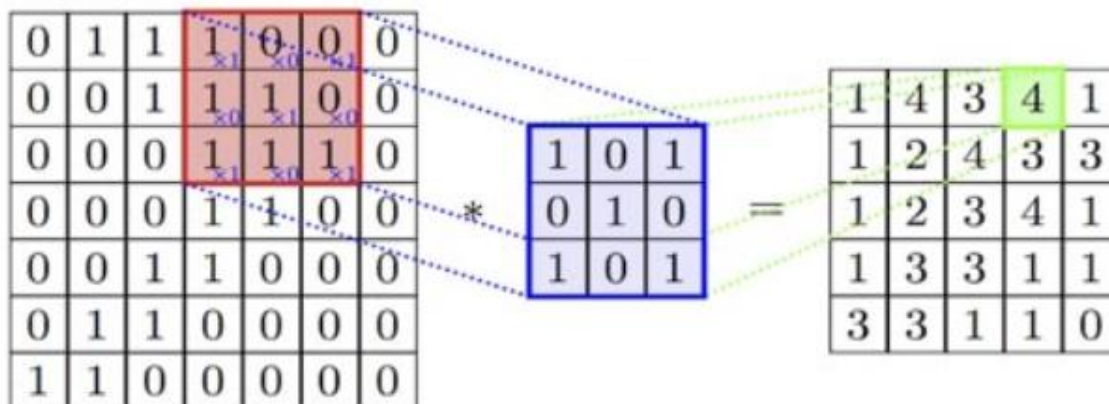
3. Cada archivo en escala de grises multiplicarlo con un kernel y generar un nuevo archivo

Por ejemplo

<http://setosa.io/ev/image-kernels/>

<https://www.saama.com/blog/different-kinds-convolutional-filters/>

La multiplicación con el kernel sería de la siguiente manera, pero de tal manera que pueda corresponder el resultado final con los valores del valor máximo y mínimo de la escala de colores.



Por ejemplo

[https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

<http://aishack.in/tutorials/image-convolution-examples/>

Utilizar 4 de estos Kernels por cada imagen:

- Simple box blur
- Gaussian blur
- Line detection with image convolutions
- Edge detection
- The Sobel Edge Operator
- The laplacian operator
- The Laplacian of Gaussian

4. Posteriormente pasar cada archivo que se procesó con el kernel a un archivo jpg.

Se le pide lo siguiente:

- Se pide escribir un código en java, exponer y redactar un informe de mínimo 3 hojas.
- Al iniciar el servidor este recibirá las imágenes y luego distribuirá las tareas a los diferentes nodos.
- Se tendrá "e" esclavos estos darán los resultados en sus carpetas predeterminadas.
- Tomar como base las explicaciones y el código de clase.
- Exponer y Ejecutar en los clusters y en Java para poder realizar la comparación y resultados.