



# INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO



## Primera Evaluación

**Unidad de Aprendizaje:**

**Visión Artificial**

**Alumno:**

**Espinosa Cruz Emilio Jared**

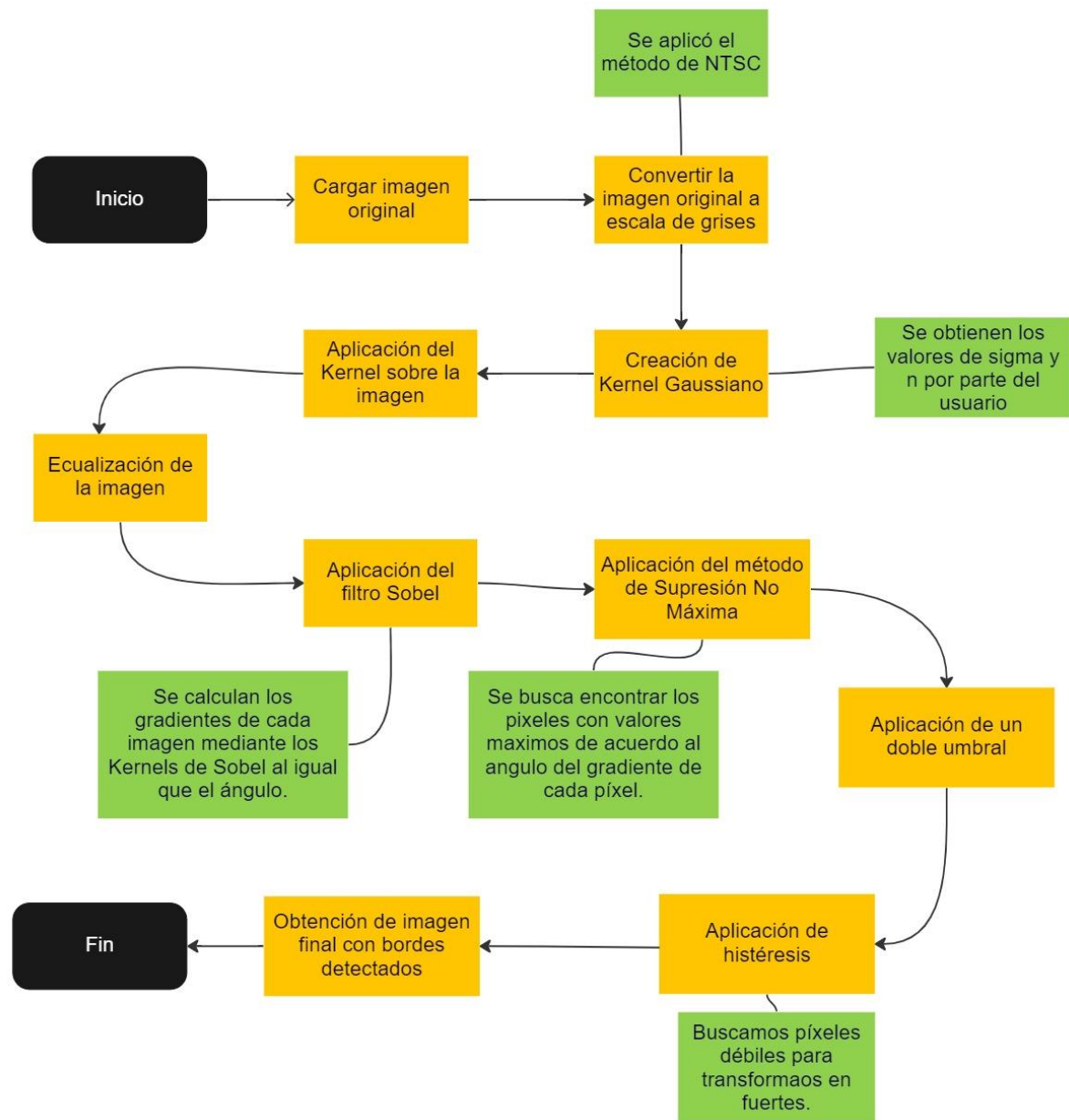
**Profesor:**

**Octavio Sánchez García**

**Grupo:**

**5BM1**

## Diagrama de procesos



## Descripción de procesos

### - **Cargar imagen original**

Para poder realizar todo el proceso, donde buscaremos detectar los bordes mediante el algoritmo de Canny, y donde aplicaremos diferentes técnicas de tratamiento de imagen, como conversión de escala de grises, aplicación de filtro gaussiano, entre otros, necesitamos definir la imagen con la que trabajaremos durante todo el proceso.

Para este caso se empleó la imagen "lena.jpg".



### - **Conversión de escala de grises**

Una vez que tenemos nuestra imagen fuente esta se debe convertir a escala de grises, ya que es primordial para poder realizar los siguientes procesos.

Para este caso se empleo el método de NSTC. La fórmula empleada para realizar este proceso es la siguiente:

$$\text{Grayscale} = 0.299R + 0.587G + 0.114B$$

El resultado obtenido es el siguiente:



### - Creación de Filtro Gaussiano

Para poder realizar el siguiente proceso, el cual es aplicar un filtro gaussiano, necesitamos crear el kernel que nos permitirá llevar a cabo el proceso. Este kernel "gaussiano" se calcula por la siguiente formula:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

El siguiente ejemplo mostrado se especifico un kernel con dimensiones 5x5 y un valor de sigma igual a 1.

```
[0.0029150175, 0.013064203, 0.021539228, 0.013064203, 0.0029150175;  
0.013064203, 0.058549695, 0.096532129, 0.058549695, 0.013064203;  
0.021539228, 0.096532129, 0.15915456, 0.096532129, 0.021539228;  
0.013064203, 0.058549695, 0.096532129, 0.058549695, 0.013064203;  
0.0029150175, 0.013064203, 0.021539228, 0.013064203, 0.0029150175]
```

### - Aplicación Filtro gaussiano

Una vez que se tiene el kernel gaussiano, este ya se puede aplicar sobre la imagen. Para poder realizar esto, sin que haya problemas de índices existentes, se procede a crear una imagen con bordes extras (n-1).



Una vez creada la imagen que nos permitirá realizar este proceso, aplicamos el kernel sobre esta imagen. Como resultado nos da esta imagen.



#### - **Ecualización de la Imagen**

El siguiente paso dentro del proceso, es proceder a ecualizar la imagen obtenida del proceso anterior. Este procedimiento se hace con la intención de balancear los niveles de intensidad de los píxeles en toda la imagen.



#### - **Aplicación Filtro Sobel**

Para poder llevar a cabo este proceso, necesitaremos hacer uso de los siguientes kernels.

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Estos kernels los aplicaremos sobre la imagen ecualizada, y dependiendo de cada kernel obtendremos una imagen diferente para cada uno, las cuales llamaremos  $G_x$  y  $G_y$ . Obtenidas estas imágenes generaremos otra, la cual le aplicaremos la siguiente formula apoyándonos de las imágenes anteriores.

$$|G| = \sqrt{I_x^2 + I_y^2},$$

Aplicada esta fórmula, se obtiene la siguiente imagen.

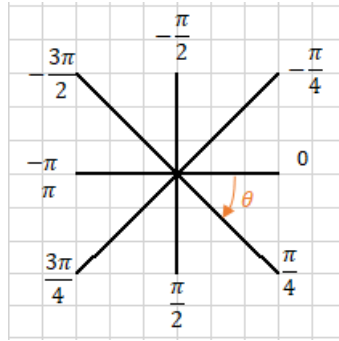


Igualmente, generaremos una matriz donde se guardaron los ángulos del gradiente de cada píxel, la cual se empleará para el método siguiente. Para esto se emplea la siguiente formula.

$$\theta(x,y) = \arctan\left(\frac{I_y}{I_x}\right)$$

#### - **Aplicación del Método de Supresión No Máxima**

Para este proceso se busca poder adelgazar los bordes de la imagen. Para este paso ocupamos la matriz con los valores de los ángulos de los gradientes del método anterior. Esta se empleará para encontrar si algún píxel alineando al eje del ángulo es mayor al valor del pixel actual.



De acuerdo con el eje donde pertenezca el ángulo, será los pixeles que se compararan.

Si alguno de los valores de estos pixeles es mayor al píxel actual su valor cambiará a cero, si no se quedará con el mismo valor.

Aplicado este proceso, se obtiene el siguiente resultado:



#### - Umbralización de la imagen

Realizado el proceso de la Supresión No Máxima, se procede a umbralizar la imagen.

Este proceso consiste en tomar dos valores como umbrales (min, máx), y categorizar los valores de los pixeles de acuerdo a estos dos:

- Si un píxel tiene un valor menor a min, este tomará un valor de 0.
- Si tiene un valor mayor a min y menor a máx, tomará un valor de 25
- Si tiene un valor mayor a máx, tomará un valor de 255.

Aplicado este proceso, se tiene el siguiente resultado:





#### - Histéresis

Como último paso de este proceso, se procede a realizar una histéresis sobre la imagen.

En este paso se busca transformar aquellos pixeles que tiene valores débiles a valores fuertes.

Para esto se evalúa si un píxel tiene el mismo valor a min del proceso anterior. Si es así, se evalúa si algún píxel de sus vecinos es igual a 255, si es así cambia su valor a 255, de lo contrario se queda con el valor de min.

Como resultado de este paso, y resultado final del proceso de detección de bordes se obtiene:





Enlace repositorio: <https://github.com/JaredEspinosa19/Visi-n-Artificial.git>