Matriz a imagen

El tiempo de ejecución es considerablemente más largo fuera del entorno de Colab, principalmente debido a que la matriz dentro de Colab ya está construida, pero aún así es posible comparar el tiempo de cálculo de estadísticas en ambos casos: al no usar ningún tipo de librería para hacer estos cálculos el tiempo que tardan en realizarse es de 2.378 segundos, mientras que al hacer uso de la librería NumPy, estos cálculos se realizan en tan solo 0.017 segundos, además de compactar en gran medida el código. NumPy además de ayudar a hacer cada uno de los cálculos rápidamente y en tan solo una línea, también funciona para la conversión del vector a una matriz.

Convoluciones en excel

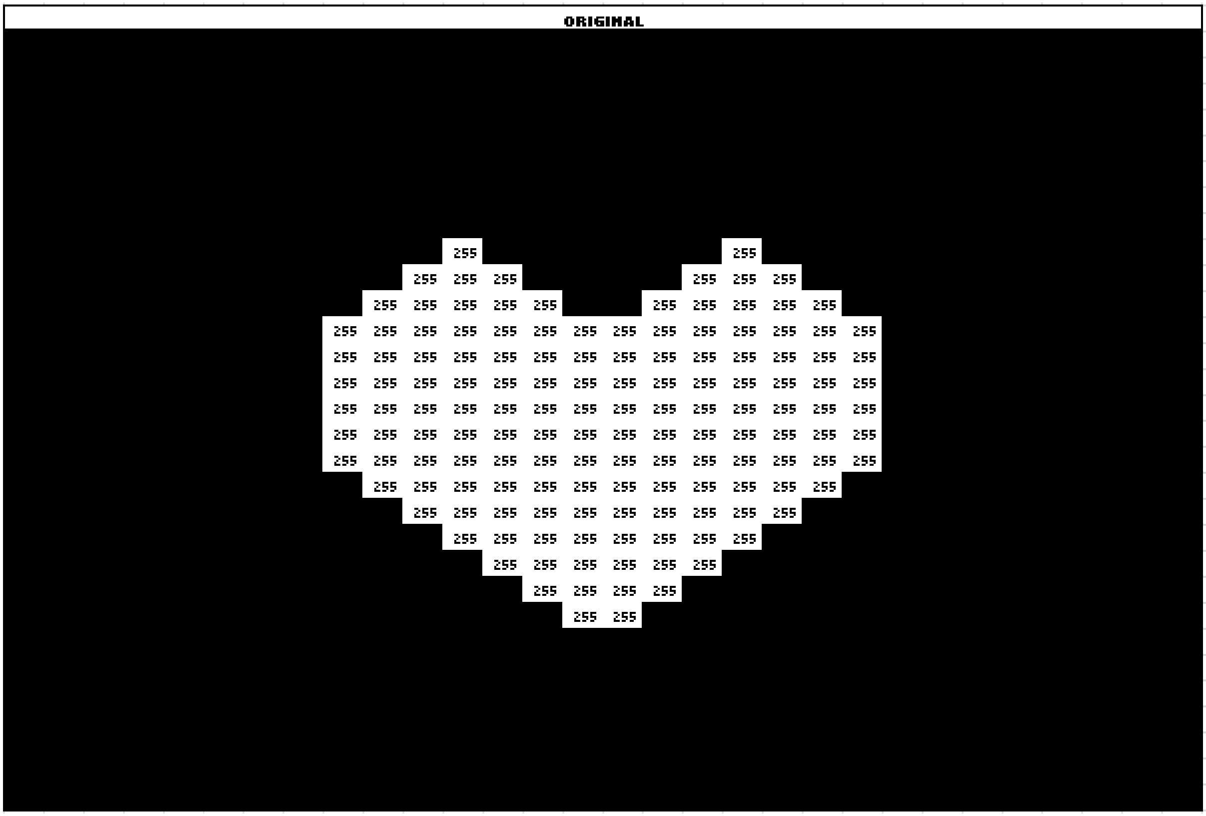
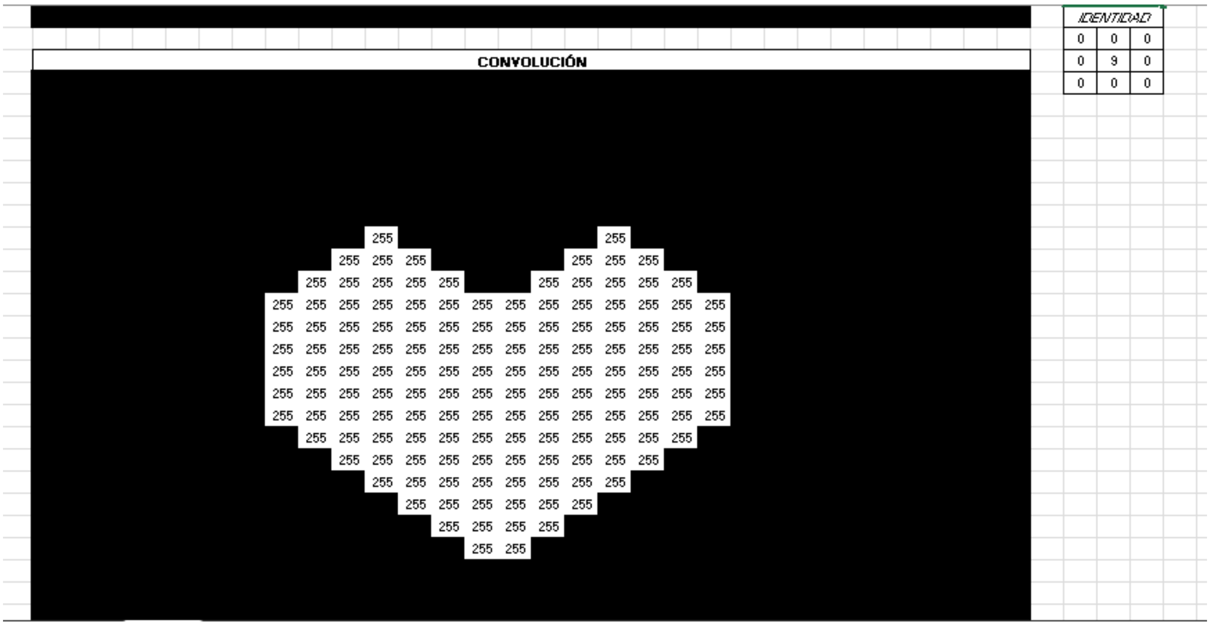


Imagen original



Convolución con Kernel: IDENTIDAD

La transformación que obtiene la imagen con este kernel, es la misma imagen. Comúnmente, se suele utilizar este kernel como:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Matriz identidad

Pero en este caso se usa en vez del 1, el 9. Esto por que en la formula de excel estamos dividiendo la suma ponderada por 9 (la cantidad de casillas del kernel).

```
=SUMAPRODUCTO(B3:D5;$AG$33:$AI$35)/9
```

Operación de convolución entre imagen original y kernel

El propósito de esto es garantizar que la imagen no se oscurezca u obtenga más brillo, y que los valores se mantengan en (0,255). La analogía es: *"Si sumé 9 cosas, divido entre 9 para quedarme en la misma escala."*

Para un kernel K:

$$\text{resultado} = \frac{\sum(imagen \cdot K)}{\sum K} \quad (\text{si} \sum K > 0)$$

Expresión simple matemática de convolución

$$\text{resultado}(i,j) = \frac{9 \cdot \text{imagen}(i,j)}{9} = \text{imagen}(i,j)$$

Cómo funciona el kernel en una imagen

CONVOLUCIÓN

			28	28	28					28	28	28			
		28	85	113	85	28			28	85	113	85	28		
	28	85	170	198	170	85	28	28	85	170	198	170	85	28	
28	85	170	227	255	227	170	113	113	170	227	255	227	170	85	28
57	142	227	255	255	255	227	198	198	227	255	255	255	227	142	57
85	170	255	255	255	255	255	255	255	255	255	255	255	255	170	85
85	170	255	255	255	255	255	255	255	255	255	255	255	255	170	85
85	170	255	255	255	255	255	255	255	255	255	255	255	255	170	85
85	170	255	255	255	255	255	255	255	255	255	255	255	255	170	85
57	142	227	255	255	255	255	255	255	255	255	255	255	227	142	57
28	85	170	227	255	255	255	255	255	255	255	255	227	170	85	28
	28	85	170	227	255	255	255	255	255	255	227	170	85	28	
		28	85	170	227	255	255	255	255	227	170	85	28		
			28	85	170	227	255	255	227	170	85	28			
				28	85	170	227	227	170	85	28				
					28	85	142	142	85	28					
						28	57	57	28						

LUR (PROMEDIO		
1	1	1
1	1	1
1	1	1

Convolución kernel: BLUR O PROMEDIO

Este kernel saca el promedio de los valores que hay alrededor de un píxel para hallar un valor central (promedio). Por eso da este efecto de “transición suave” en la imagen.

BLUR (PROMEDIO)		
1	1	1
1	1	1
1	1	1

Matriz Blur

1. **Suma del kernel**

$$\sum K = 1+1+1+1+1+1+1+1+1 = 9$$

2. **Producto imagen · kernel**

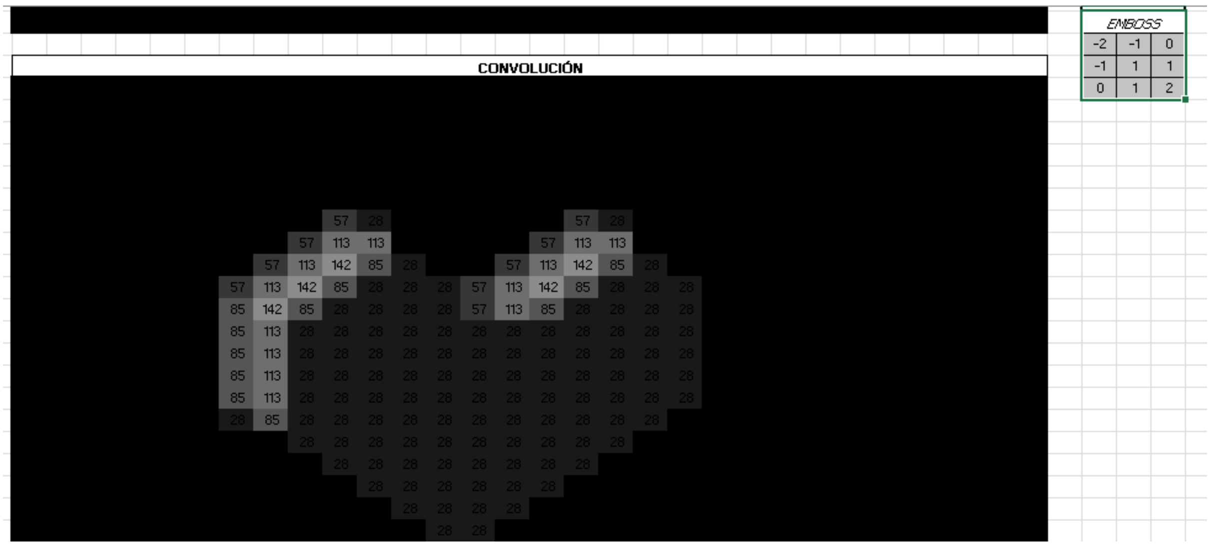
   Como todos los coeficientes valen 1, la suma es simplemente la suma de los 9 píxeles del vecindario 3×3:

$$\sum(\text{imagen} \cdot K) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \text{imagen}(i+m, j+n)$$

3. **Resultado final**

$$\boxed{\text{resultado}(i,j) = \frac{1}{9} \sum_{m=-1}^{1} \sum_{n=-1}^{1} \text{imagen}(i+m, j+n)}$$

Cómo funciona el kernel en una imagen



Convolución kernel: EMBOSS

Un kernel **define una dirección** según **cómo se distribuyen los pesos positivos y negativos:**

- **Pesos negativos** → "quitan luz" (sombra)
- **Pesos positivos** → "añaden luz" (iluminación)

Que hacen los distintos pesos

Pero, por que ilumina más hacia donde en el kernel hay pesos negativos? (superior izquierda)

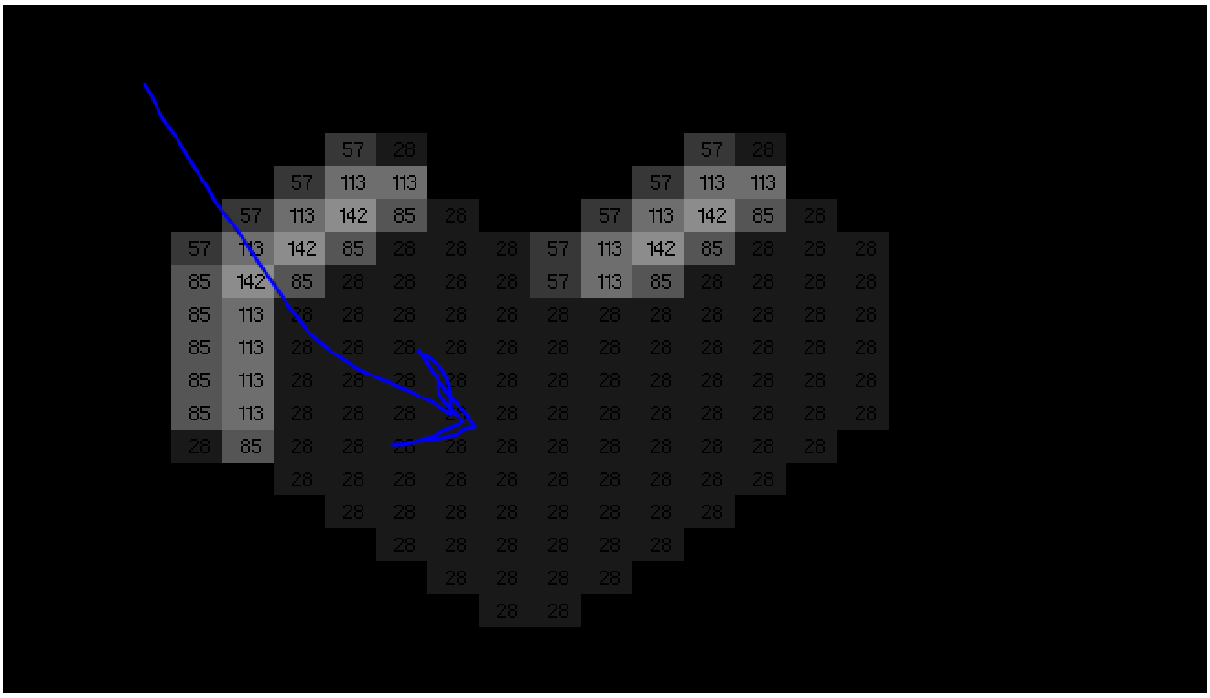


Imagen con pendiente

EMBOSS		
-2	-1	0
-1	1	1
0	1	2

Matriz con pendiente

El kernel hace la operación no solo sobre el píxel en cuestión, sino también con sus píxeles vecinos. Es como una recta que va de la parte superior izquierda a la inferior derecha, si va encontrando un valor más claro hacia dónde se dirige esta recta, lo deja "clarito", si va encontrando un valor más oscuro lo va dejando más "oscurito".

1 Suma del kernel

$$\sum K = (-2 - 1 + 0) + (-1 + 1 + 1) + (0 + 1 + 2) = 1$$

✓ Como $\sum K > 0$, **se puede aplicar la normalización**
y en este caso **no cambia el valor**, porque dividir por 1 no altera el resultado.

---

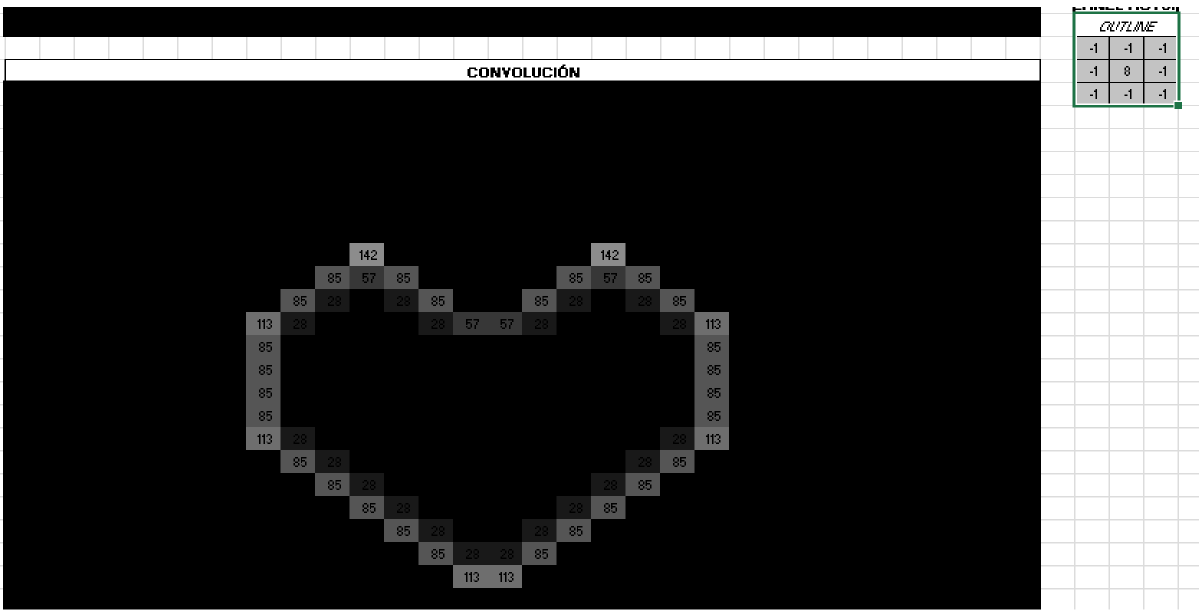
2 Expresión matemática completa

$$\boxed{\text{resultado}(i,j) = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \text{imagen}(i+m, j+n) \cdot K(m,n)}$$

Desarrollado explícitamente:

$$\begin{aligned} \text{resultado}(i,j) = &- 2\, I_{i-1,j-1} - 1\, I_{i-1,j} + 0\, I_{i-1,j+1} \\ &- 1\, I_{i,j-1} + 1\, I_{i,j} + 1\, I_{i,j+1} \\ &+ 0\, I_{i+1,j-1} + 1\, I_{i+1,j} + 2\, I_{i+1,j+1} \end{aligned}$$

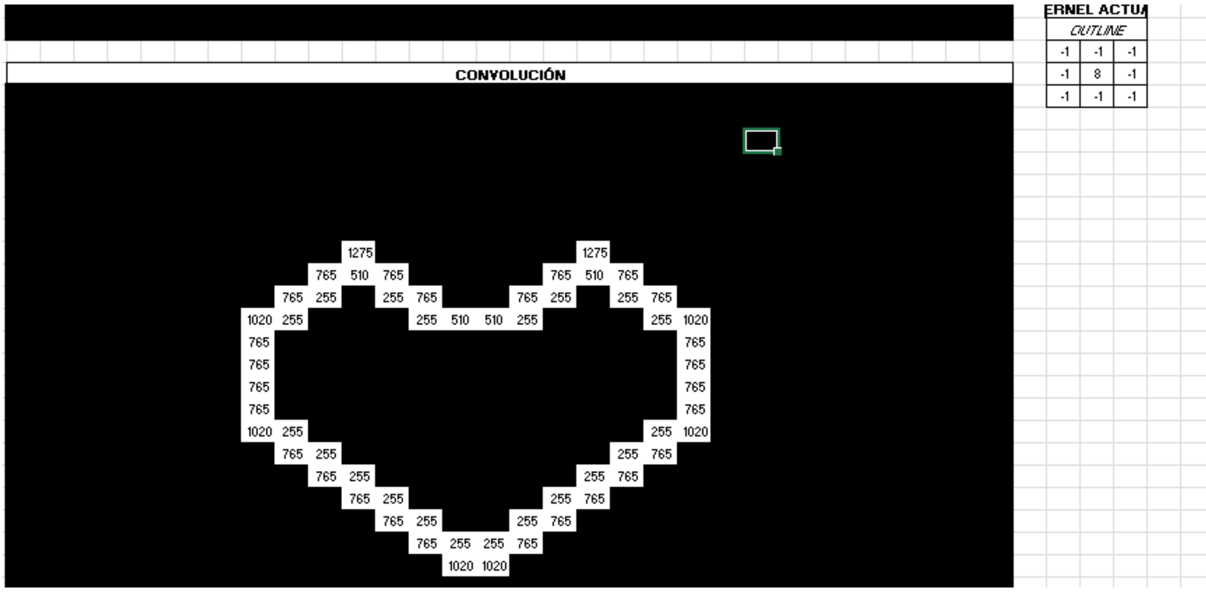
Expresión matemática Emboss



Convolución kernel: OUTLINE

Este kernel resalta los bordes, no resalta las zonas uniformes.
Algo a tener en cuenta es que los valores de este kernel sumados dan 0, por ende esta operación de convolución no es necesario normalizar. Normalizar es dividirla por el tamaño del kernel (3x3=9). Aquí se está usando una fórmula normalizada y por eso el resultado que

obtenemos es el de la transformación a una escala distinta. Si no se estuvieran normalizando los valores, los bordes estuviesen más claros o blancos.



Convolución sin normalizar

Aquí por ejemplo se usa para la convolución otra fórmula y el mismo kernel. Quitando la división, podemos ver que hay valores que se pasan de 255 (se tendría que validar eso) y los bordes ahora si son blancos. Esto se mostró con fines de explicación.

Kernel outline

$$K = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\sum K = 8 - 8 = 0$$

Expresión matemática (convolución en (i, j))

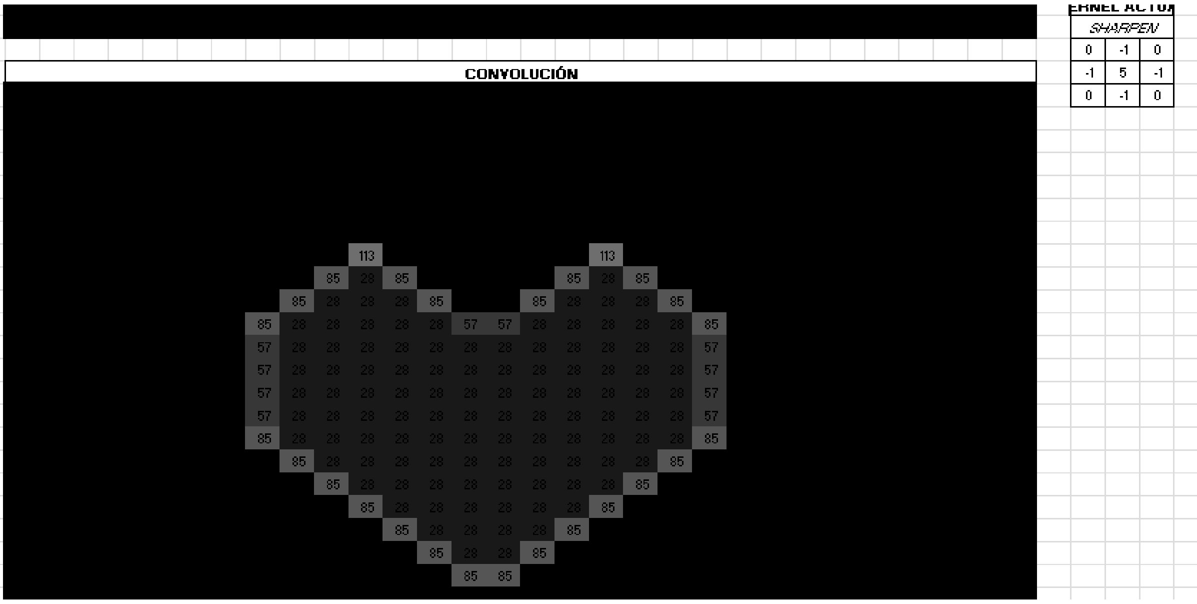
Como $\sum K = 0$, no se puede normalizar dividiendo por $\sum K$. Entonces:

$$\boxed{\text{resultado}(i, j) = \sum(\text{imagen} \cdot K)}$$

Desarrollado explícitamente:

$$\boxed{\begin{aligned} \text{resultado}(i, j) = & -I_{i-1,j-1} - I_{i-1,j} - I_{i-1,j+1} \\ & - I_{i,j-1} + 8I_{i,j} - I_{i,j+1} \\ & - I_{i+1,j-1} - I_{i+1,j} - I_{i+1,j+1} \end{aligned}}$$

Expresión matemática kernel OUTLINE



Convolución kernel: SHARPEN

Al contrario de blur, este resalta los detalles, vuelve los bordes más finos.

Qué hace matemáticamente

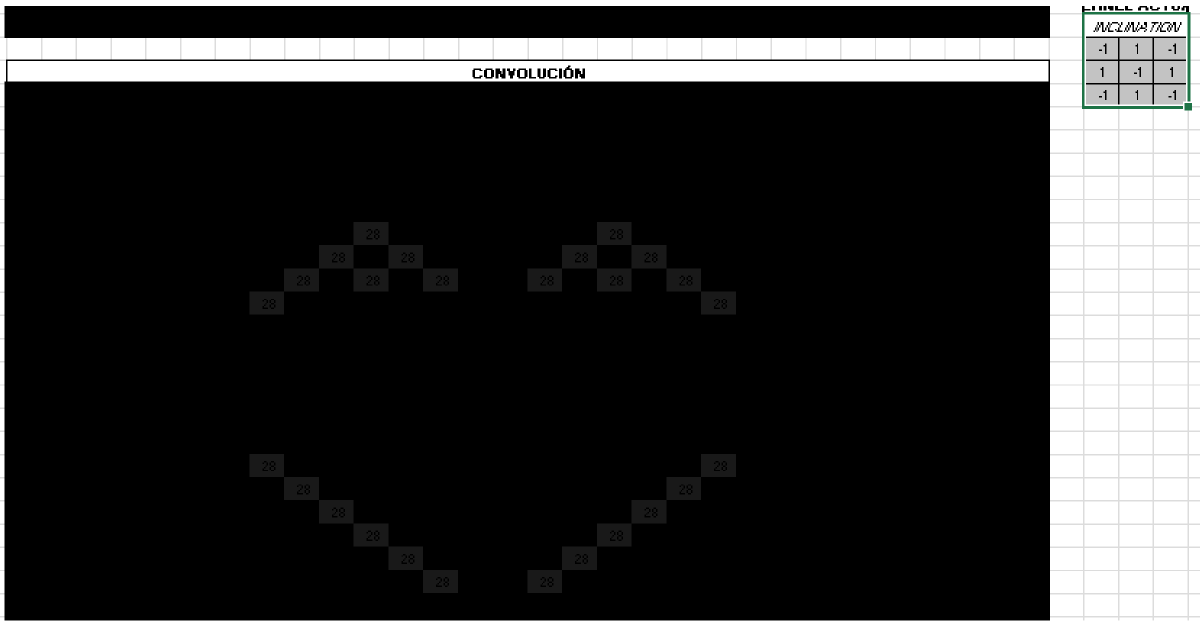
$$\text{resultado}(i,j) = 5I(i,j) - I(i-1,j) - I(i+1,j) - I(i,j-1) - I(i,j+1)$$

Es equivalente a:

$$\boxed{\text{resultado} = \text{imagen} + (\text{imagen} - \text{blur})}$$

O sea: **imagen original + detalles.**

Expresión matemática kernel sharpen



Convolución kernel: INCLINATION (propio)

Este kernel resalta los bordes donde hay inclinación en la imagen, donde hay cambios y hay líneas con pendiente distinta a cero este kernel resalta. Está basado en el kernel blur, pero este resalta más cambios o patrones “ajedrezados”. Como lo es de hecho, la propia matriz kernel.

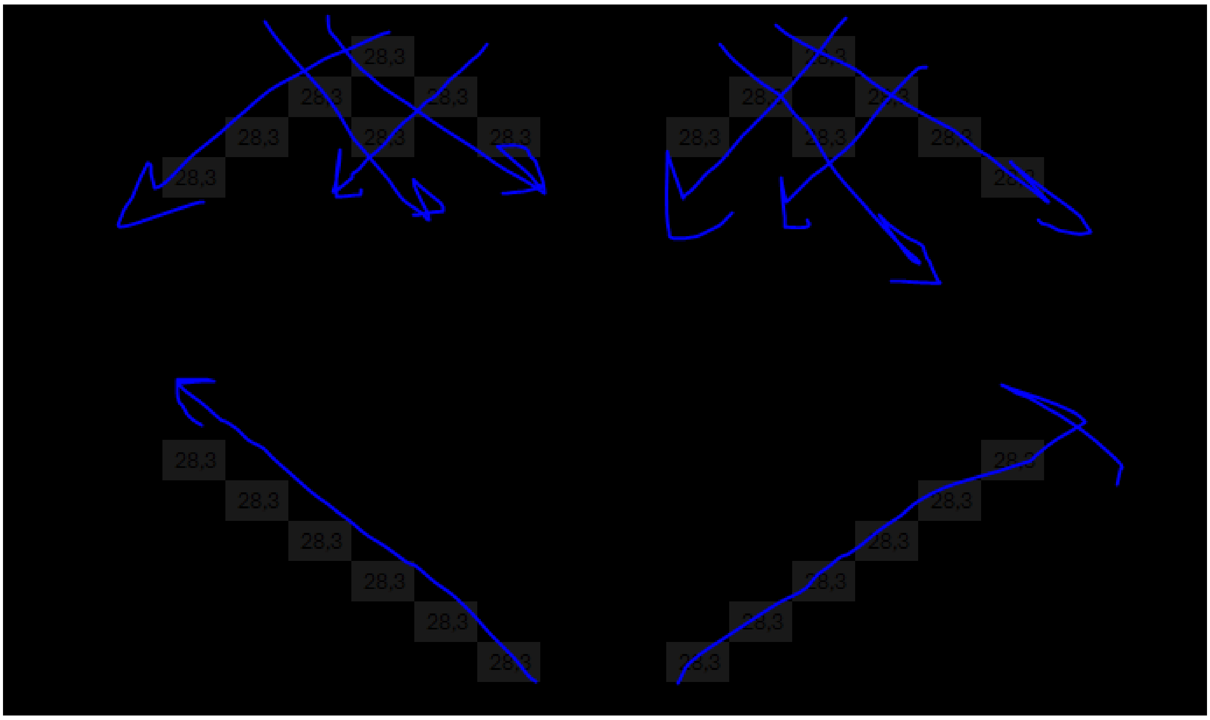


Imagen con diagonales resaltadas

Su suma es:

$$\sum K = (-1 + 1 - 1) + (1 - 1 + 1) + (-1 + 1 - 1) = -1$$

Expresión matemática (con tu estilo)

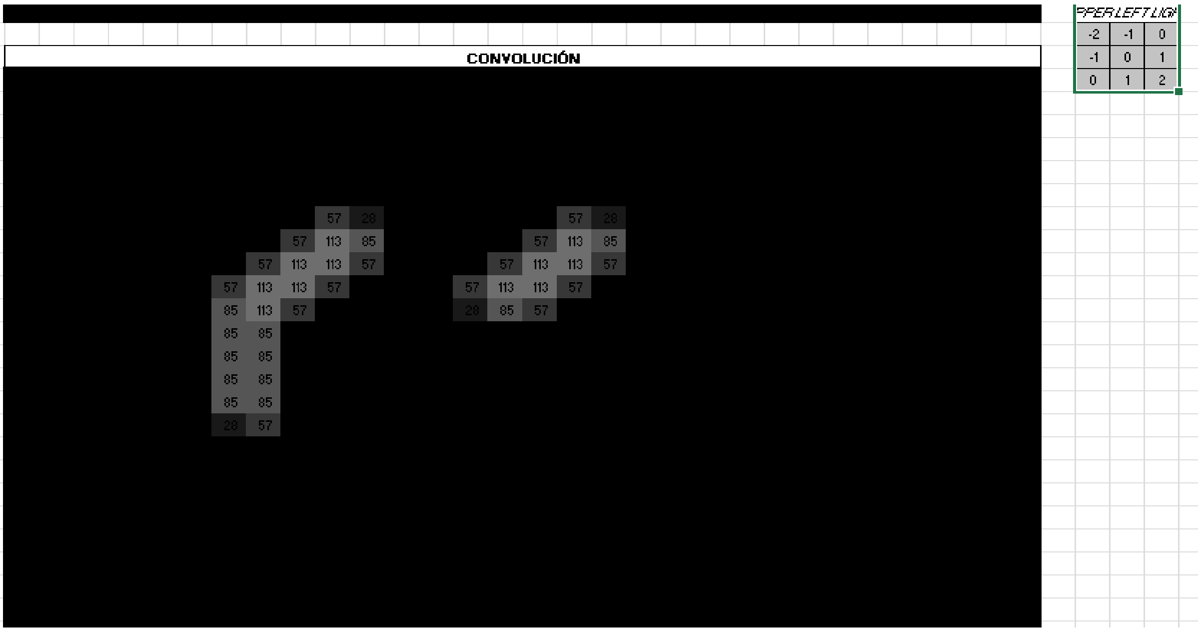
Como $\sum K \leq 0$, no aplica tu división por $\sum K$ bajo la condición "si $\sum K > 0$". Entonces:

$$\boxed{\text{resultado}(i,j) = \sum(\text{imagen} \cdot K)}$$

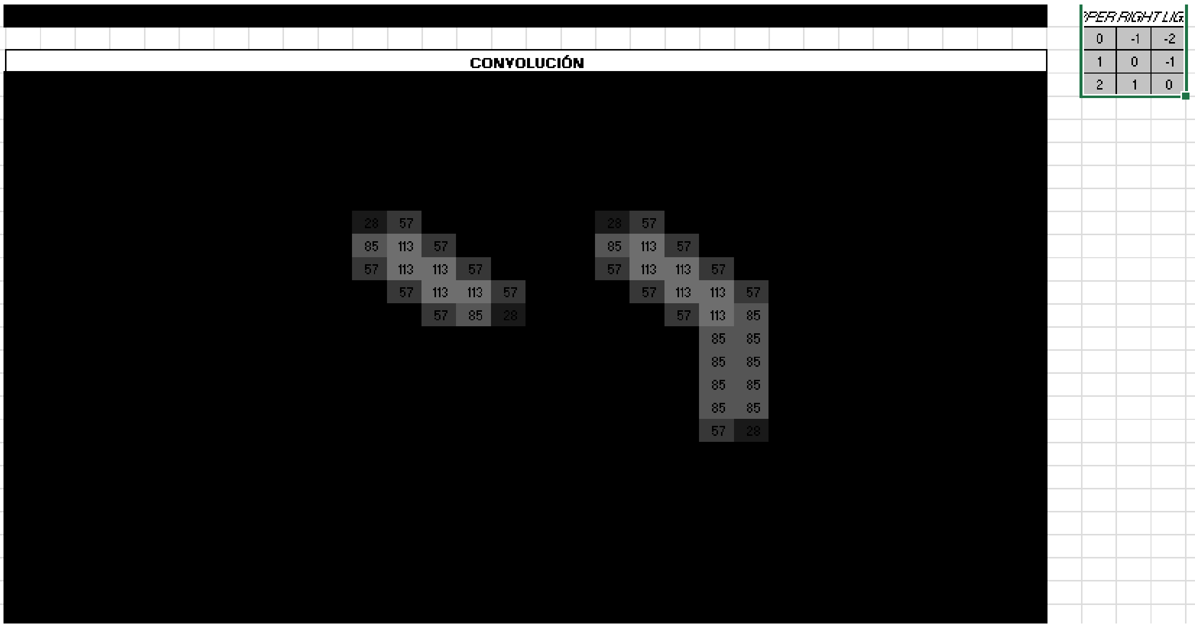
Desarrollado explícitamente (vecindario 3×3):

$$\boxed{\begin{aligned}\text{resultado}(i,j) = & -I_{i-1,j-1} + I_{i-1,j} - I_{i-1,j+1} \\ & + I_{i,j-1} - I_{i,j} + I_{i,j+1} \\ & - I_{i+1,j-1} + I_{i+1,j} - I_{i+1,j+1}\end{aligned}}$$

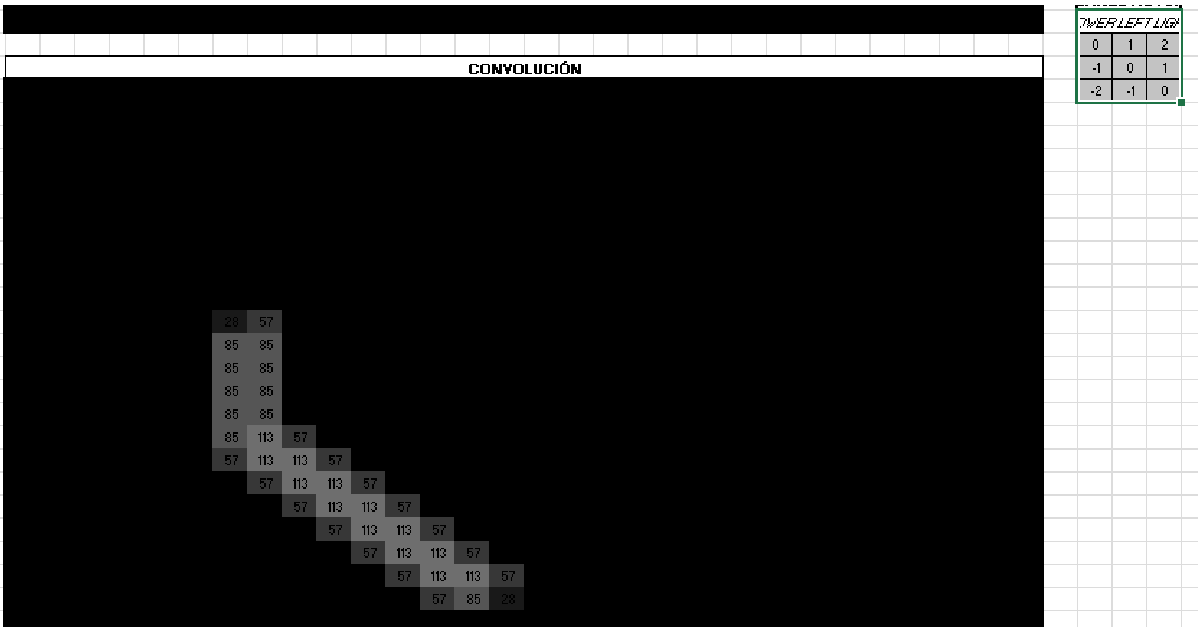
Expresión matemática kernel INCLINATION



Convolución kernel: UPPER LEFT LIGHT (propio)



Convolución kernel: UPPER RIGHT LIGHT (propio)



Convolución kernel: LOWER LEFT LIGHT (propio)

Convolución kernel: LOWER RIGHT LIGHT (propio)

Estos últimos 4, están basados en el kernel EMBOSS. Así como el emboss crea un efecto de iluminación desde la parte superior izquierda y va bajando la intensidad, cada uno de estos crea un efecto de iluminación similar, con la diferencia de que la iluminación solo se encuentra en el borde que choca con la “fuente de luz”, lo otro permanece totalmente sin luz.
El funcionamiento es similar, la línea viene desde el extremo donde están los pesos negativos y va iluminando hacia donde se encuentra la mayor intensidad.

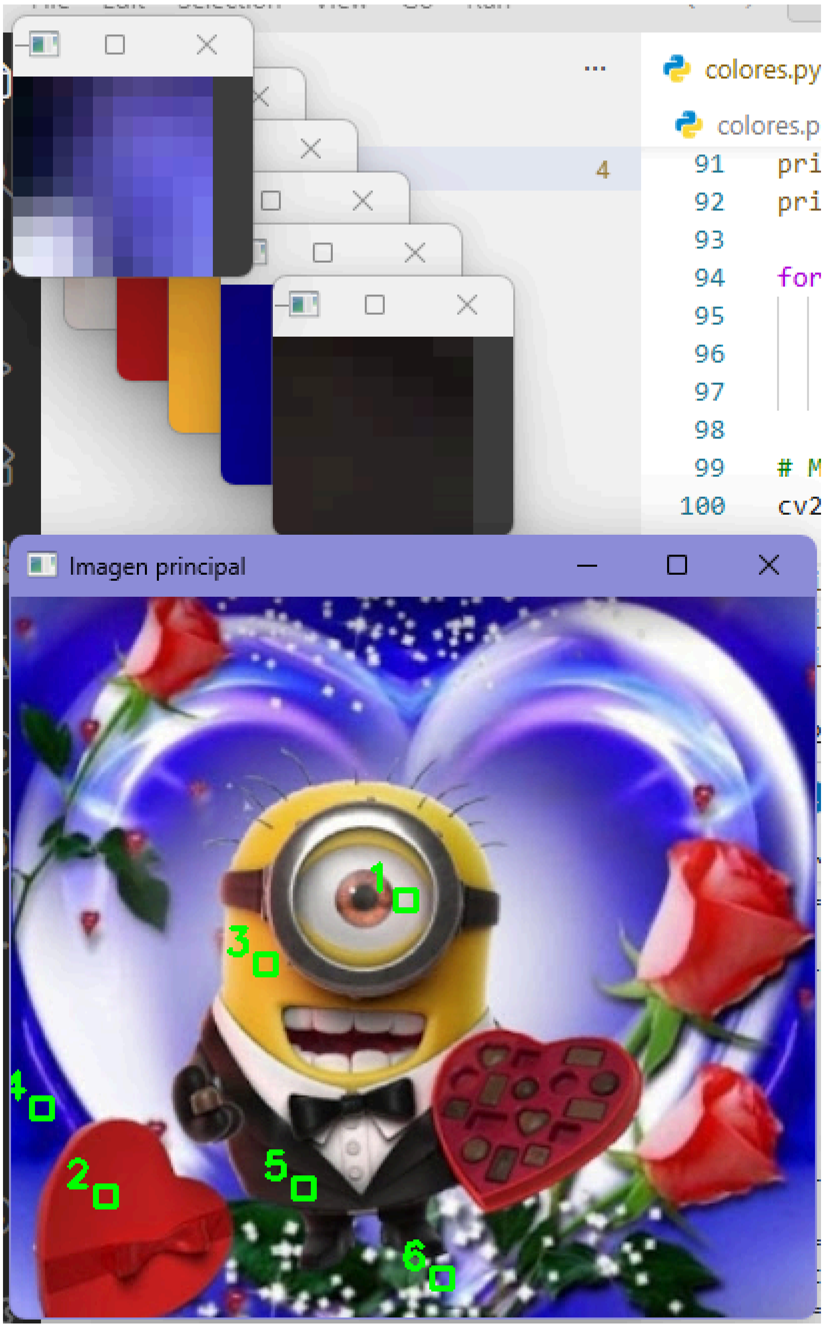
$$\sum K = 0 \rightarrow \text{no conservan brillo}$$

Por eso el centro de la imagen, a pesar de ser blanca, termina negra pues no se conserva ese brillo donde no hay cambios.

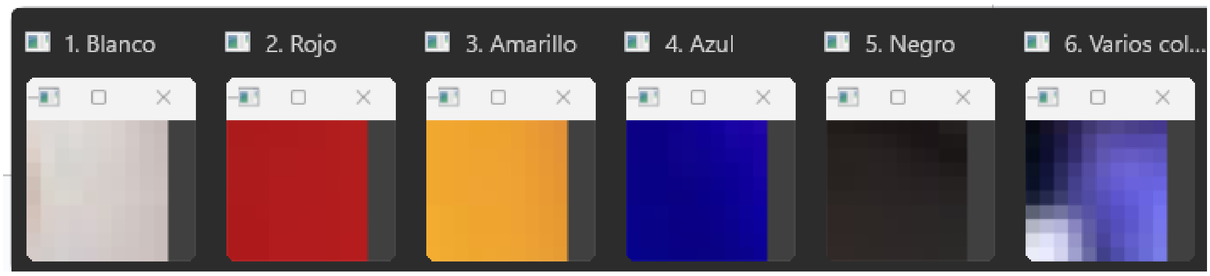
El padding se rellenó con los mismos ceros.

Colores

Al ver la tabla de resultados, es posible observar que el color con promedios más altos es el blanco, mientras que el promedio más bajo lo tiene el color negro, y ambos son los colores con los valores más consistentes en sus tres canales. En cuanto a las desviaciones estándar, la región de interés con más colores tiene la mayor desviación estándar en todos sus canales, que es de esperarse pues tiene la variedad más alta de valores, por otro lado, el negro y el azul tienen la menor desviación estándar pues tienen tonos muy uniformes en sus regiones de interés.



Ubicación de las regiones de interés y sus respectivas ventanas



Colores de regiones de interés

Seguimiento de figura en video

Al encender la cámara se puede ver que identifica varios elementos del fondo por sus respectivos colores, su precisión dependerá de la especificidad con la que se definan las variables de los colores, sin embargo, los colores de las figuras en la hoja de papel los identifica correctamente, el único problema siendo la iluminación del ambiente, pero aún así responde de manera adecuada a los colores que se muestran y cumple su objetivo.

Colores detectados en la hoja