

UNIVERSIDAD DE DARMSTADT
B. FRÖMMER, E. HERGENRÖTHER, B. MEYER



8 DE JUNIO de 2023

VISUAL INFORMÁTIC SoSe 2023

A

TAREA 5

5.1 Transformación lineal del valor de gris

Cargue la imagen "schrott.png". Usando la función `plotHistogram` que se encuentra en Moodle, puedes mirar el histograma de valores de gris de la imagen. ¿Qué cambia si estableces `acumulativo` a verdadero?

Ahora hay que realizar una transformación lineal del valor de gris en esta imagen para reducir artificialmente el contraste. Para ello, primero se necesitan los valores de los valores de gris máximo y mínimo. A continuación, escale la imagen de modo que tenga un rango de contraste limitado de 100 a 150 (es decir, que el valor mínimo de brillo sea 100 y el máximo 150).

En el siguiente paso, se trata de maximizar el rango de contraste reescalando la imagen de bajo contraste (es decir, cubriendo el rango de valores $[0 \dots 255]$) ¡y así superar incluso el de la imagen original! Compare los histogramas de la imagen resultante y de la imagen original, ¿qué se nota y cómo se explica esta diferencia?

5.2 Ecualización del histograma

De la lección (y de la tarea 5.1) conoce una forma de maximizar el rango de contraste de una imagen. Esto suele ser deseable porque facilita el reconocimiento de los detalles de la imagen. Sin embargo, el simple escalado de la imagen no puede evitar una distribución desigual de píxeles claros y oscuros. La imagen `schrott.png`, por ejemplo, contiene un número relativamente grande de píxeles en la gama media de grises, por lo que el contraste "local" de la imagen sólo cambia mínimamente al escalar, y los detalles de la imagen no se mejoran.

Un desarrollo posterior del escalado es la ecualización del histograma, que garantiza que todas las intensidades posibles de los píxeles aparecen aproximadamente con la misma frecuencia en la imagen. Implementa el siguiente algoritmo y Pruébalo en el archivo `scrap.png` (si lo deseas, puedes adaptar el método auxiliar `plotHistogram` para ello).

- Paso 1: Cálculo del histograma
- Paso 2: Cálculo del histograma acumulado
- Paso 3: Cálculo de la función de distribución acumulativa (FDA): Dividir el recuento de píxeles acumulados por la resolución de la imagen (es decir, el número máximo de píxeles).
- Paso 4: Multiplicar la función de distribución por la intensidad de píxel más alta posible de la imagen de destino (normalmente 255) y redondear hacia abajo al número entero más próximo.
- Paso 5: Aplicar el mapeado resultante a cada píxel de la imagen de entrada.

Un ejemplo del cálculo y funcionamiento del algoritmo puede verse en la Figura 1, para una imagen de entrada de 4 bits.

Si tu implementación funciona, echa un vistazo al histograma resultante. ¿Qué ocurre en el histograma acumulado? ¿Cómo se comporta el algoritmo con la imagen ajustada con contraste reducido de la tarea 5.1?

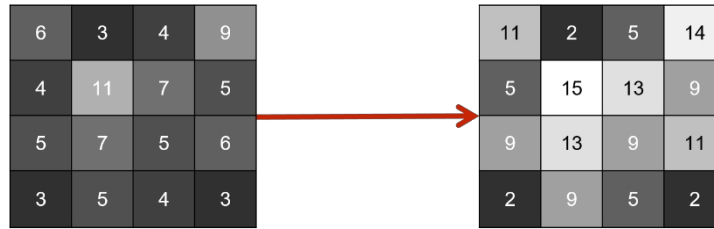


Figura 1: Ejemplo de conversión para una imagen de entrada con un rango dinámico de 4 bits

Pixel Intensität	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pixelanzahl	0	0	0	3	3	4	2	2	0	1	0	1	0	0	0	0
Kumulativ	0	0	0	3	6	10	12	14	14	15	15	16	16	16	16	16
Kumulative Verteilung	0	0	0	0,1 875	0,3 75	0,6 25	0,7 5	0,8 75	0,8 75	0,9 375	0,9 375	1	1	1	1	1
Abbildung	0	0	0	2	5	9	11	13	13	14	14	15	15	15	15	15

Figura 2: Ejemplo de cálculo para una imagen de entrada con un rango dinámico de 4 bits

5.3 Plegado de imágenes

Antes de ver el cálculo de la convolución en OpenCV, es útil echar un vistazo más de cerca al cálculo subyacente. La imagen "kante.png" debería servir como ejemplo, los valores de píxel para esto se pueden encontrar en la tabla de la derecha.

Define tú mismo una matriz de convolución razonable de 3x3 para un desenfoque gaussiano (ver clase) y dobla a mano (en papel) el kante.png original con la matriz.

Para ello, piense cómo quiere tratar los valores límite que faltan. ¿Qué alternativas existen?

Evalúa la imagen resultante, ¿qué ha ocurrido exactamente? ¿Qué puedes decir sobre el contraste local de la imagen?

(es decir, la diferencia de brillo de los píxeles vecinos)?

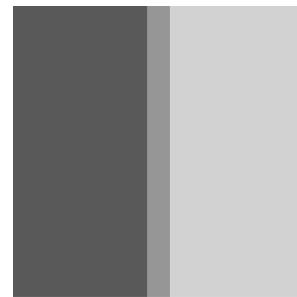
continuación, queremos probar la convolución integrada de OpenCV:

C++:

```
cv::filter2D (InputArray src,
              OutputArray dst,
              int ddepth,
              InputArray kernel,
              Punto ancla = Punto(-1,-1), doble
              delta = 0,
              int borderType = BORDER_DEFAULT
              )
```

Python:

```
cv2.filter2D( src,
              dprofundidad
              , núcleo[,
              dst[,
              ancla[,
              delta[,
              borderType]]]] ) -> dst
```



90	90	90	150	210	210	210
90	90	90	150	210	210	210
90	90	90	150	210	210	210
90	90	90	150	210	210	210

Figura 3: Edge.png A

Para ello, proceda del siguiente modo:

- Cargue la imagen "kante.png" y realice una convolución con la matriz de convolución F1. (Establezca ddepth en -1, entonces la profundidad de imagen de la imagen resultante corresponde a la imagen de entrada).
- Los valores resultantes son cercanos a cero (porque los pesos sumados de la matriz de convolución dan como resultado cero, no uno). Por lo tanto, añadir el valor 128 a la convolución, esto se hace a través del parámetro delta.
- Aplique la matriz de convolución F2 al resultado (de nuevo con delta=128).
- Aplicar la matriz de convolución F3 (delta=128) a una copia de la imagen original.

$$F_1 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0,333 & 0,333 & 0,333 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0,333 & -0,666 & 0,333 \\ 0 & 0 & 0 \end{bmatrix}$$

Preguntas:

- 1) ¿Qué aporta la comparación de los dos últimos resultados generados?
- 2) Las dos matrices de convolución F1 y F2 diferencian la imagen (es decir, cada una de ellas forma la derivada) en la dirección de las filas. ¿Qué hace la matriz de convolución F3 con respecto al término "derivada"?

- Doble una copia de la imagen original con la matriz de convolución F4.
- Doble otra copia de la imagen original independientemente con la matriz de convolución F5.
- Si ahora restáramos la imagen creada con F5 de la imagen creada con F4, esto correspondería a una convolución de la imagen original con la matriz de convolución F6, que tiene un valor de La resta de estas dos matrices de convolución es: $F_6 = F_4 - F_5$. Realiza la convolución con F6 (delta=128).

Preguntas:

- 3) ¿Qué se obtiene al comparar píxel a píxel el resultado de la convolución con F6 con el resultado de la convolución con F3? Razona las diferencias.
- 4) ¿Qué hacen los dos pliegues F4 y F5?
- 5) ¿Qué se obtiene si se gira 90 grados la matriz de convolución F3 y se añade a la propia F3? ¿Cómo se llama este operador y qué hace con respecto al término "derivada"?