

(۱۵) درجه حبت انتی :

$M_H =$

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

$M_{\sqrt{2}}$

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

درجه حبت یودی :

M_H

$$\begin{bmatrix} 3 \\ 1 \\ 0 \\ 3 \\ 1 \\ 0 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

$=$

$$\begin{aligned} 6+1+ -6+ -1 &= 0 \\ 3+2+ -3+ -2 &= 0 \\ 3+1+ -3+ -1 &= 0 \\ -6-1+6+1 &= 0 \\ -3-2+3+2 &= 0 \\ -3-1+3+1 &= 0 \\ 6+1-6-1 &= 0 \\ 3+2-3-2 &= 0 \\ 3+1-3-1 &= 0 \end{aligned}$$

(b)

۰ ۳×۳
ماتریس
نه چون همگرا و درجه
انتی لم یارب جواب منفی است

در جهت عمودی

$$M_v = \begin{bmatrix} 3 \\ 1 \\ 6 \\ 3 \\ 1 \\ 0 \\ 3 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2+1+1 \\ -6-3-3 \\ 6-2+3-1+3-1 \\ 1+2+1 \\ -3-6-3 \\ 3-1+6-2+3-1 \\ \text{بالین برت} \end{bmatrix} = \begin{bmatrix} 4 \\ -9 \\ 8 \\ 4 \\ -9 \\ 8 \\ 4 \\ -9 \\ -8 \end{bmatrix}$$

\Rightarrow ماتریس حاصل $= \begin{bmatrix} 4 & -9 & 8 \\ 4 & -9 & 8 \\ 4 & -9 & 8 \end{bmatrix}$

در صورتی که در جهت عمودی برابر 0 و 9- هستند.

(c) به طور کلی در تک ها میانی ماشین که می خواهیم در یک ناحیه مشخص از تصویر چیزی را بیابیم (مانند

اشیا یا لبه) الگوریتم محلی است در یک ناحیه چگونگی عدد از آن چیزی مورد نظر را بازگرداند (مثلاً در

تشخیص لبه. در یک ناحیه 3×3 لبه ها عمدتاً با $\text{gradient intensity}$ ها عمدتاً پیدا شود. در object detection چندین مشکل در یک شیء کشیده شود. $\text{non maximal suppression}$ در یک ناحیه میانی

همه را میبندد. بهترین را انتخاب می کنند. مانند الگوریتم canny که در سوال سوم بیاد دارید

شده است. الگوریتم در هر راستا در ناحیه ها 3×3 پیکسلی. بیشترین مقدار intensity را انتخاب می کنند و بقیه را 0 می کنند.

$$\begin{bmatrix} -1 & 0 & 1 \\ -3 & 0 & 3 \\ -1 & 0 & 1 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -3 & -1 \\ 0 & 0 & 0 \\ 1 & 3 & 1 \end{bmatrix} \leq \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 1 \end{bmatrix}$$

(d)

در صورت جدای نپذیرفتن این فیلتر، عمل فیلتر کردن یک تصویر را هم بصورت مادی در هر پیکسل زمانی $O(KN^2)$ انجام می شود. در مرتبه اول، $O(2n^3)$ انجام داد. که بسیار بهتر و دلیل آن این است که ما توانیم عمل کانولوشن یک پیکسل $K \times K$ در یک پیکسل $N \times N$ را که به رنج است در مرتبه اول زمانی $O(K^2(N-K+1)^2)$ انجام می شود. تبدیل به در عمل کانولوشن در مرتبه اول زمانی $O(KN(N-K+1))$ کرد.

(e) خیر. فیلترهای Sobel در شکل اصلی دارند.

(1) لبه ها را فقط در جهت قائم عمود در انی پیدا می کنند و لبه ها در جهت های دیگر تشخیص می دهند از روی این به جهت مستقیم

(2) این فیلترها فقط بزرگ صاف و grayscale عمل می کنند و برای تصاویر رنگی هم باید بتواند تبدیل به grayscale شود که ممکن است در این صورت اطلاعات از دست برود.

(f) خیر. گوی می توان این کار را انجام داد ولی ریزش ها بهتر مانند فیلتر کردن خطوط. تصویر ... وجود دارد.

$$h(x) = u(x) * f(x) = \sum_{k=-\infty}^{\infty} u(k) f(x-k) \quad (g)$$

$$\Rightarrow h'(x) = \sum_{k=-\infty}^{\infty} u(k) f'(x-k) = \boxed{u(x) * f'(x)}$$

$$g(x) = G_0(x) * u(x)$$

(h)

$$g'(x) = G_0'(x) * u(x)$$

صورت (g) :

$$g''(x) = G_0''(x) * u(x) = x^2 e^{-\frac{x^2}{2}} * u(x)$$

بزرگ صاف صاف (h)