تشخیص لبه با OpenCV

1. مقدمه :تشخیص لبه چیست؟

تشخیص لبه یکی از تکنیک‌های کلیدی در پردازش تصویر است که به دنبال شناسایی مرزها و تغییرات شدید روشنایی در تصویر می‌گردد .این تکنیک برای استخراج ویژگی‌های مهم تصویر به کار می‌رود.

چرا تشخیص لبه مهم است؟

- تمرکز روی جزئیات کلیدی :اطلاعات مهم مثل خطوط و اشکال را مشخص می‌کند.
- کاهش حجم داده‌ها :فقط بخش‌های مهم تصویر را نگه می‌دارد.
- کاربردها :شناسایی اشیا، پردازش چهره، ردیابی حرکت و موارد دیگر.

2. روش‌های اصلی تشخیص لبه

الف) مشتقات مرتبه اول: Sobel

این روش‌ها گرادیان تصویر را پیدا می‌کنند؛ یعنی میزان تغییر روشنایی را در جهت‌های افقی و عمودی می‌سنجند.

گرادیان افقی :(X) تغییر روشنایی از چپ به راست.

گرادیان عمودی :(Y) تغییر روشنایی از بالا به پایین.

ب) مشتقات مرتبه دوم: لاپلاسین

لاپلاسین تغییرات روشنایی را می‌سنجد. برخلافSobel ، لاپلاسین جهت تغییرات را نادیده می‌گیرد و به همه جهات حساس است. به طور کلی لاپلاسین به شکل دیگری به تغییرات تصویر نگاه می‌کند و علاوه بر بررسی میزان تغییرات (مثل گرادیان)، سرعت تغییرات را هم می سنجد. بنابراین لاپلاسین برای پیدا کردن **لبه‌هایی که خیلی واضح هستن** کاربردی است اما به نویز تصویر حساس است.

ج) روشCanny

یک الگوریتم پیشرفته‌تر که شامل این مراحل است:

1. کاهش نویز :حذف نویز با فیلتر Gaussian
2. محاسبه گرادیان: یافتن شدت و جهت تغییرات روشنایی.

3. حذف نوامی غیربیشینه: لبه‌های ضعیف‌تر حذف می‌شوند.

4. آستانه‌گذاری :فقط لبه‌های قوی و بین دو حد مجاز نگه داشته می‌شوند.

3. مقایسه روش‌ها

روش	مزایا	معایب
Sobel	سریع و ساده	دقت پایین در تصاویر نویزی
Laplacian	حساس به تغییرات شدید روشنایی	حساسیت بالا به نویز
Canny	دقیق و تمیز، کاهش نویز	تنظیم پارامترهای زیاد، پیچیدگی بالا

4. پیاده‌سازی درOpenCV

الف) بارگذاری تصویر و پیش‌پردازش

تصاویر معمولاً دارای نویز هستند که می‌تواند باعث شناسایی اشتباه لبه‌ها شود .برای همین ابتدا تصویر را بلور میکنیم:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# تصویر خواندن
image = cv2.imread('path_to_image.jpg', cv2.IMREAD_GRAYSCALE)

# فیلتر با نویز کاهش Gaussian
blurred_image = cv2.GaussianBlur(image, (5, 5), 0)
```

ب) محاسبه لبه‌ها باSobel

```
# عمودی و افقی گرادیان محاسبه
sobel_x = cv2.Sobel(blurred_image, cv2.CV_64F, 1, 0, ksize=3)
sobel_y = cv2.Sobel(blurred_image, cv2.CV_64F, 0, 1, ksize=3)
```

ج)محاسبه لبه‌ها با لاپلاسین

```
لاپلاسین اعمال #
laplacian = cv2.Laplacian(blurred_image, cv2.CV_64F, ksize=3)
```

د) محاسبه لبه‌ها با روشCanny

```
Canny روش اعمال #
edges = cv2.Canny(blurred_image, threshold1=50, threshold2=150)
```

5. نکات کلیدی

- ✓ پیش‌پردازش تصویر :حتماً تصویر را بلور کنید تا نویز کاهش یابد.
- ✓ انتخاب روش مناسب :بسته به نیاز پروژه، روش مناسب را انتخاب کنید.
- ✓ تنظیم پارامترها :برای روش‌هایی مثلCanny ، پارامترهای آستانه باید با آزمون و خطا تعیین شوند.

6. نتیجه‌گیری

تشخیص لبه به ما کمک می‌کند تا بخش‌های کلیدی و مهم تصویر را شناسایی کنیم .روش‌های مختلفی مثلSobel ، Laplacian و Canny وجود دارند که بسته به نیاز پروژه می‌توان از آن‌ها استفاده کرد.