"به نام خدا"

گزارش پروژه درس سیگنال ها و سیستم ها

دانشجو : محمد مهدی شرف بیانی – 401521372

نکته مهم :

بدلیل واضح بودن کد نوشته شده و کامنت گذاری های منظم ، از توضیح خطوط کد در این گزارش پرهیز شده است و فقط مفاهیم ، مشاهدات و نتایج آمده است.

سوال اول :

تحلیل و طراحی فیلترهای دیجیتال برای پردازش سیگنالهای صوتی با استفاده از تبدیل فوریه الگوریتم های فیلتر بالا و پایین گذر

این کد به بررسی و پردازش یک سیگنال صوتی می‌پردازد و از ابزارهای پردازش سیگنال دیجیتال برای تحلیل و تغییر آن استفاده می‌کند. در این گزارش، مفاهیم علمی و تئوری‌های مرتبط با فیلترهای دیجیتال، تبدیل فوریه، و تأثیر فیلترها بر سیگنال توضیح داده می‌شود.

---

1) تبدیل فوریه(Fourier Transform)

تبدیل فوریه یک ابزار ریاضی است که سیگنال را از حوزه زمان به حوزه فرکانس منتقل می‌کند. به زبان ساده، این تبدیل نشان می‌دهد که سیگنال ورودی از چه فرکانس‌هایی تشکیل شده است و شدت هر فرکانس چقدر است.

کاربرد در این کد:

- نمایش سیگنال در حوزه فرکانس :با استفاده از تبدیل فوریه سریع(FFT) ، مشخص می‌شود که سیگنال صوتی اصلی و سیگنال‌های فیلترشده چه فرکانس‌هایی را شامل می‌شوند.
- آنالیز فرکانسی :این تحلیل برای مشاهده تأثیر فیلترها بر محتوای فرکانسی سیگنال ضروری است.

---

2) فیلتر دیجیتال(Digital Filter)

فیلترهای دیجیتال ابزارهایی هستند که برای تغییر محتوای فرکانسی یک سیگنال به کار می‌روند. آن‌ها می‌توانند بخش‌هایی از فرکانس را حذف، تقویت یا تضعیف کنند.

انواع فیلترها در این کد:

- فیلتر پایین‌گذر:(Low-pass Filter)
این فیلتر فرکانس‌های بالاتر از یک مقدار مشخص (بُرش) را حذف می‌کند و فقط فرکانس‌های پایین‌تر از آن مقدار را عبور می‌دهد.
    - تأثیر :فرکانس‌های بالا که معمولاً مربوط به نویز یا جزئیات تیز هستند، حذف می‌شوند. نتیجه یک سیگنال نرم‌تر و عاری از فرکانس‌های بالا خواهد بود. برای مثال، در موسیقی، فیلتر پایین‌گذر می‌تواند جزئیات زیر و بم صدای آلات موسیقی را کاهش دهد.
- فیلتر بالاگذر:(High-pass Filter)
این فیلتر فرکانس‌های پایین‌تر از مقدار مشخص را حذف کرده و فقط فرکانس‌های بالا را عبور می‌دهد.
    - تأثیر :فرکانس‌های پایین که معمولاً شامل نویز پس‌زمینه یا اجزای کندتر سیگنال هستند، حذف می‌شوند. این کار باعث برجسته‌تر شدن جزئیات سریع‌تر و تیزتر سیگنال می‌شود.

- فیلتر Band_Pass (ترکیبی از دو فیلتر بالا)
این فیلتر ترکیبی از فیلترهای بالاگذر و پایین‌گذر است و یک باند خاص از فرکانس‌ها را عبور می‌دهد. در این کد، ابتدا فیلتر پایین‌گذر اعمال می‌شود و سپس فیلتر بالاگذر، که نتیجه آن عبور فقط فرکانس‌های میانی است.
    - تأثیر :سیگنال به یک محدوده خاص از فرکانس‌ها محدود می‌شود. این روش برای استخراج اطلاعات خاص از سیگنال کاربرد دارد.

---

3) طراحی فیلتر(Filter Design)

فیلترهای دیجیتال در این کد با استفاده از روش Butterworth طراحی شده‌اند. این نوع فیلتر به دلیل پاسخ فرکانسی صاف و بدون نوسان در محدوده عبور، بسیار محبوب است. ویژگی‌های کلیدی طراحی فیلتر:

- مرتبه فیلتر :(Order) در این کد، مرتبه 4 انتخاب شده است که نشان‌دهنده شیب پاسخ فرکانسی در محدوده قطع است. مرتبه بالاتر باعث شیب تندتر می‌شود اما محاسبات بیشتری نیاز دارد.
- فرکانس قطع :(Cutoff Frequency) این مقدار تعیین می‌کند که کدام بخش از سیگنال حذف یا عبور داده شود.

---

4) تأثیر اعمال فیلترها بر سیگنال

- فیلتر پایین‌گذر:
با حذف فرکانس‌های بالا، سیگنال نرم‌تر و کم‌جزئیات‌تر می‌شود. این کار برای کاهش نویز یا حذف صداهای تیز و ناخواسته مفید است. مثلاً در یک سیگنال موسیقی، صدای بم بیشتر شنیده می‌شود و صدای زیر کم‌رنگ می‌شود.
- فیلتر بالاگذر:
با حذف فرکانس‌های پایین، سیگنال تیزتر و پرجزئیات‌تر می‌شود. این فیلتر برای برجسته‌سازی جزئیات سریع یا حذف نویز پس‌زمینه کاربرد دارد. مثلاً در یک مکالمه ضبط‌شده، نویز باس حذف شده و صدای صحبت واضح‌تر می‌شود.
- فیلتر محدوده مشخص (ترکیب دو فیلتر بالا) :
این فیلتر بخش‌های پایین و بالای فرکانس را حذف می‌کند و فقط محدوده میانی را عبور می‌دهد. نتیجه آن یک سیگنال محدود به فرکانس‌های خاص است که ممکن است برای استخراج اطلاعات خاص یا تمرکز بر یک محدوده خاص از سیگنال مفید باشد.

---

5) نمودار های رسم شده در کد

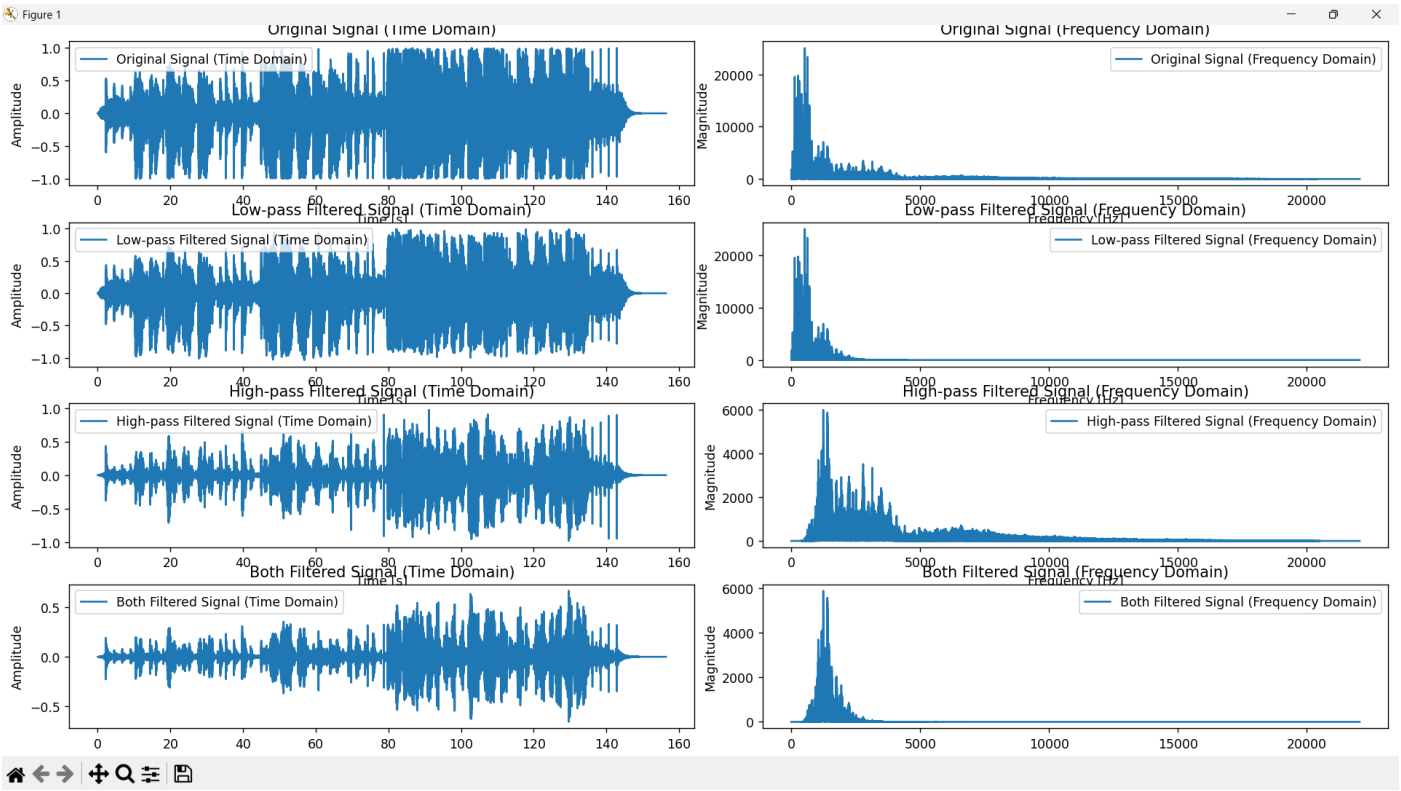
در این کد برای هر سیگنال که در یک سطر رسم شده است ، سمت چپ سطر ، نمودار سیگنال در حوزه زمان را می بینیم و در سمت راست سطر نیز نمودار همان سیگنال را در حوزه فرکانس می بینیم.

6) تحلیل نمودار خروجی برای فایل aud1.mp3 :

کات آف برای فیلتر پایین گذر روی 2000 هرتز گذاشته شده است.

کات اف برای فیلتر بالا گذر روی 1000 هرتز قرار داده شده است.

پس فیلتر Band_Pass فقط محدوده 1000 تا 2000 هرتز را نگه میدارد و این در نمودار مشهود است.



سطر اول مربوط به سیگنال صوت است و سطر دوم سیگنال فیلتر شده با فیلتر پایین گذر می باشد که طبق فرکانس رسم شده از ان می بینیم که فرکانس بالای 2000 هرتز وجود ندارد و سطر سوم برای سیگنال تولید شده توسط فیلتر بالاگذر است و طبق رسم حوزه فرکانس آن مشاهده میشود که فرکانس زیر 1000 هرتز موجود نیست

و سطر 4 ام مربوط به band_pass فیلتر می باشد که طبق رسم حوزه فرکانس آن ، فقط فرکانس بین 1000 هرتز و 2000 هرتز موجود است.

همچنین نتایج این فیلتر ها در فایل های با نام های مشخص در پوشه q1 ذخیره شده اند.

سوال دوم :

کانولوشن سیگنالهای تصادفی و تحلیل آن در دامنه فرکانس

طبق این خاصیت تبدیل فوریه که در تصویر پایین آمده است اگر دو سیگنال در حوزه زمان کانوالو شوند پس در حوزه فرکانس در هم ضرب می شوند.

$$h(t) = f(t) * g(t) \quad \Longrightarrow \quad H(\omega) = F(\omega) \cdot G(\omega)$$

Here:

- $h(t)$ is the convolution of $f(t)$ and $g(t)$.
- $H(\omega)$, $F(\omega)$, and $G(\omega)$ are the Fourier Transforms of $h(t)$, $f(t)$, and $g(t)$, respectively.

در این سوال برای تولید سیگنال های تصادفی از این تابع استفاده کردیم :

```python
def generate_random_signals(length=1000, distribution='normal'):
        """
        Generate random signals with different distributions

        Args:
            length (int): Length of signals
            distribution (str): Type of distribution ('normal', 'uniform', 'exponential')

        Returns:
            tuple: Two random signals
        """
        if distribution == 'normal':
            signal1 = np.random.normal(0, 1, length)
            signal2 = np.random.normal(0, 1, length)
            # signal2 = np.zeros(length)
        elif distribution == 'uniform':
            signal1 = np.random.uniform(-1, 1, length)
            signal2 = np.random.uniform(-1, 1, length)
        elif distribution == 'exponential':
            signal1 = np.random.exponential(1, length)
            signal2 = np.random.exponential(1,)

        return signal1, signal2
```

تولید سیگنال‌ها

دو نوع سیگنال تصادفی تولید شده‌اند:

1. **سیگنال اول و دوم :**سیگنال‌هایی با توزیع نرمال با میانگین صفر و واریانس واحد.
2. این سیگنال‌ها به‌صورت تصادفی تغییرات سریع دارند و رفتار آن‌ها در حوزه فرکانس، طیف گسترده‌ای از فرکانس‌ها را نشان می‌دهد.

کانولوشن در حوزه زمان

کانولوشن سیگنال‌ها در حوزه زمان ترکیبی از دو سیگنال اولیه است که اثر یکی بر دیگری را نشان می‌دهد. سیگنال حاصل (سیگنال کانولوشن) معمولاً طول بیشتری نسبت به سیگنال‌های اولیه دارد.

- **مشاهده :**سیگنال کانولوشن به دلیل ترکیب دو سیگنال تصادفی، رفتار پیچیده‌تری در حوزه زمان نشان می‌دهد.

مشاهده

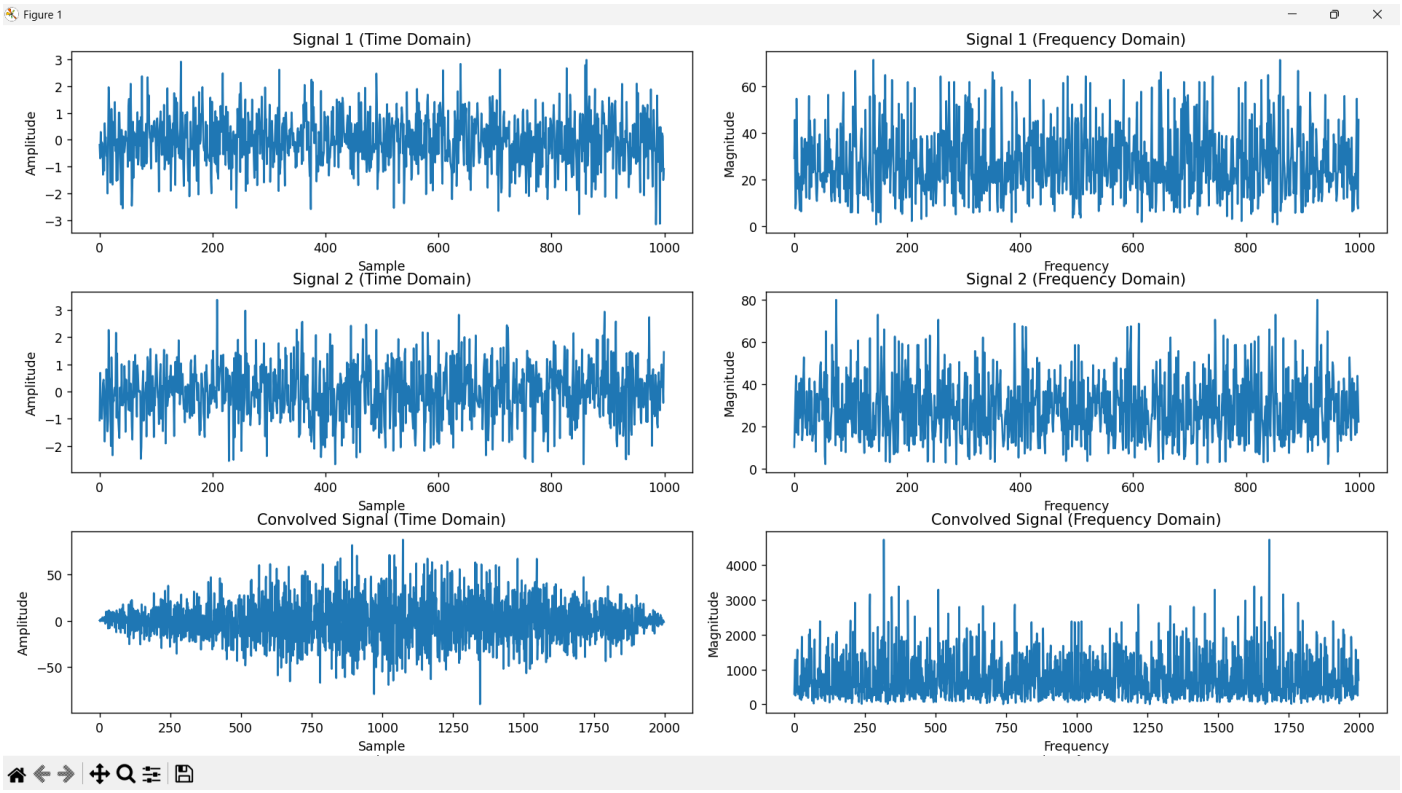
بر اساس خاصیت کانولوشن و تاثیر آن بر تبدیل فوریه سیگنال میتوان گفت :

1. طیف فرکانسی سیگنال کانولوشن پهن‌تر از سیگنال‌های اولیه است.
2. در فرکانس‌های خاص، دامنه طیف کانولوشن نسبت به سیگنال‌های اولیه تقویت شده است.

نتایج و تحلیل

1. **اثر کانولوشن :**کانولوشن باعث ترکیب اطلاعات زمانی دو سیگنال شده و رفتار پیچیده‌تری در حوزه زمان ایجاد می‌کند.
2. **اثر در حوزه فرکانس :**دامنه فرکانسی سیگنال کانولوشن ترکیبی از طیف فرکانسی سیگنال‌های اولیه است و در برخی فرکانس‌ها **تقویت** یا **تضعیف** مشاهده می‌شود.
3. **رفتار تصادفی :**به دلیل ماهیت تصادفی سیگنال‌ها، طیف فرکانسی گسترده‌ای تولید شده که نشان‌دهنده توزیع انرژی در بازه وسیعی از فرکانس‌ها است.

نمونه ای از تحلیل نموداری این بخش از پروژه :



سوال سوم :

شناسایی لبه ها در تصاویر با استفاده از کانولوشن

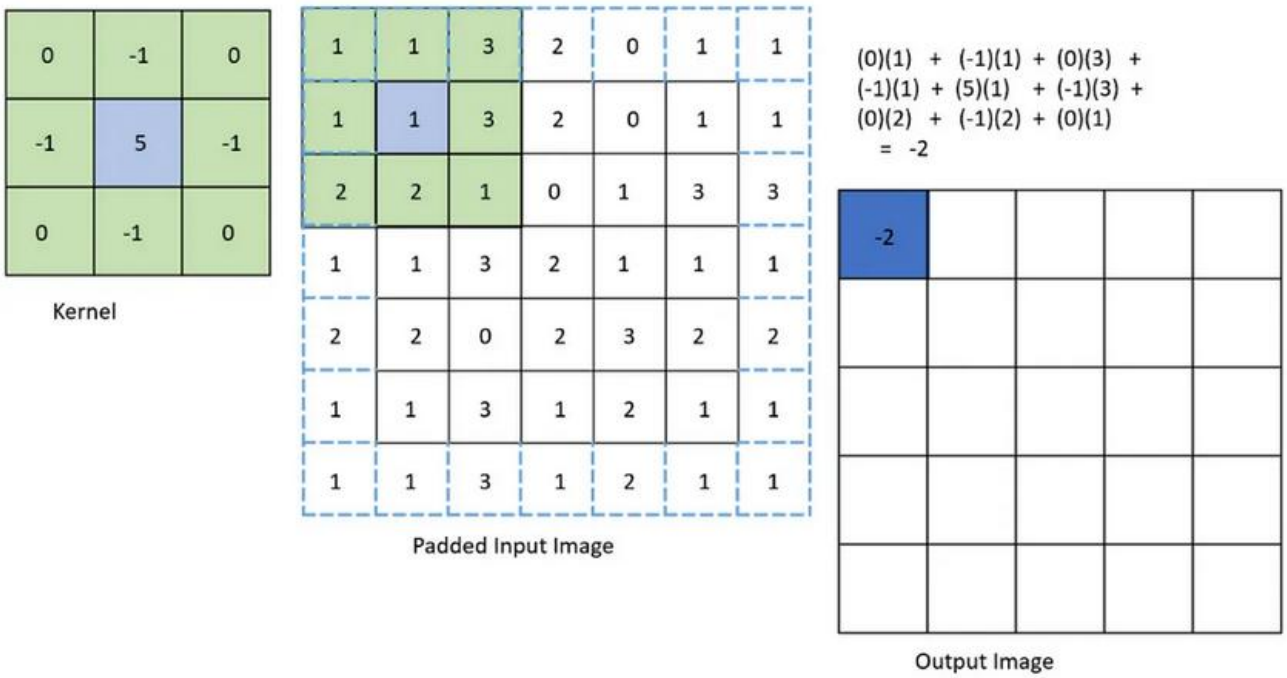
در ابتدا مفاهیم تیوری مسیله را شرح می دهیم :

کانولوشن در فضای دو بعدی :

$$y(n,m) = \sum_k \sum_l x(k,l)k(n-k,m-l))$$

k, l represents the row, length indices of the kernel respectively. *x(n,m)* is the input and *y(n,m)* is the output images. *n,m* is the row, column indices of the input and output images.

در این رابطه منظور از کرنل همان فیلتر می باشد که می خواهیم در تصویر اعمال کنیم و به صورت شماتیک اینگونه خواهد بود :



Convolution computation illustrated.

به طور کلی ماتریس کرنل (فیلتر) روی ماتریس تصویر پیمایش می کند و در هر مرحله با توجه به محاسبات بالا که به طور واضح انجام شده است ، مقدار یکی از درایه های ماتریس تصویر خروجی را محاسبه می کند.

روال کدنویسی :

فایل اول : استفاده از فیلتر های آماده

طبق تصویر بالا ابتدا باید عکس را با ابزار های مربوطه به فرم ماتریسی تبدیل کنیم(imread) که در کد توضیح داده شده است و سپس با فراخوانی تابع مربوطه (filter2D) فیلتر را روی عکس اعمال کنیم.

نکته : در کد نوشته شده ، متغیری تحت عنوان filter موجود است که با نام گذاری آن با یکی از کلید های که در دیکشنری اول کد وجود دارد می توان آن فیلتر خاص را روی عکس ورودی اعمال کرد.

توضیحات هر کدام از فیلتر های موجود در دیکشنری :

1. **Sobel X** و **Y:** برای تشخیص لبه‌های افقی و عمودی استفاده می‌شوند.
2. **Prewitt X** و **Y:** مشابه Sobel هستند ولی با وزن‌دهی ساده‌تر.
3. **Laplacian:** تشخیص لبه‌ها با استفاده از تغییرات دوم شدت نور.
4. **LoG:** ترکیب فیلتر گوسی و لاپلاسی برای کاهش نویز و شناسایی لبه‌ها.
5. **Box Blur** و **Gaussian Blur:** برای محوشدگی تصاویر.
6. **Sharpen** و **Edge Enhancement:** برای برجسته کردن جزئیات.
7. **Emboss:** ایجاد اثر برجستگی سه‌بعدی در تصویر.
8. **Outline:** استخراج مرزهای اجسام.

فایل دوم : پردازش تصویرو تولید فیلتر سفارشی برای شناسایی لبه ها و ویژگی های تصویر

در این فایل که تمرکز اصلی پروژه بر این موضوع می باشد با استفاده از تابع create_feature_based_filter که ورودی آن یک تصویر می باشد ، ویژگی های تصویر اعم از لبه و گوشه را شناسایی می کنیم و سپس فیلتری متناسب با آن تصویر میسازیم و در نهایت طبق روال قبلی می توانیم از این فیلتر استفاده کنیم.


ضمیمه :

گزارش تابع `create_feature_based_filter`

تابع `create_feature_based_filter` یک فیلتر سفارشی ایجاد می‌کند که بر اساس ویژگی‌های تصویر ورودی مانند لبه‌ها، گوشه‌ها و گرادیان شدت طراحی شده است. این فیلتر می‌تواند برای تحلیل و پردازش تصویر به کار رود. در ادامه، جزئیات عملکرد و مفاهیم ریاضی استفاده‌شده توضیح داده می‌شود.

ورودی‌ها

1. `image` : تصویر ورودی که می‌تواند رنگی یا خاکستری باشد.
2. `kernel_size` : اندازه هسته (فیلتر) که باید عددی فرد باشد (پیش‌فرض: ۳).

خروجی

- یک فیلتر سفارشی با اندازه مشخص‌شده که بر اساس ویژگی‌های استخراج‌شده از تصویر ساخته شده است.

مراحل پردازش و مفاهیم ریاضی

1. تبدیل تصویر رنگی به خاکستری (در صورت نیاز)

- اگر تصویر ورودی رنگی باشد (۳ کانال)، با استفاده از معادله زیر به خاکستری تبدیل می‌شود:

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

- این تبدیل شدت روشنایی پیکسل‌ها را محاسبه می‌کند.

---

2. استخراج لبه‌ها با استفاده از الگوریتم Canny

- الگوریتم Canny برای تشخیص لبه‌ها استفاده می‌شود. این الگوریتم شامل مراحل زیر است:
- محاسبه گرادیان‌های شدت تصویر (∇_x ، ∇_y):

$$\frac{\partial I}{\partial x} = \nabla_x \quad , \frac{\partial I}{\partial y} = \nabla_y$$

- محاسبه اندازه گرادیان:

$$\sqrt{\nabla_x^2 + \nabla_y^2} = |\nabla|$$

- اعمال آستانه‌گذاری برای تشخیص لبه‌ها.

3. تشخیص گوشه‌ها با استفاده از الگوریتم Harris

- گوشه‌ها با استفاده از ماتریس گرادیان‌ها و معادله زیر محاسبه می‌شوند:

$$\begin{bmatrix} {}_{y}I_{x}I\sum & {}_{x}^{2}I\sum \\ {}_{y}^{2}I\sum & {}_{y}I_{x}I\sum \end{bmatrix} = M$$

- مقدار پاسخ Harris برای هر پیکسل:

$${}^{2}(\text{trace}(M)) \cdot k - \det(M) = R$$

که در آن k یک پارامتر ثابت است.

---

4. محاسبه گرادیان شدت با فیلترهای Sobel

- فیلترهای Sobel برای محاسبه گرادیان شدت تصویر استفاده می‌شوند:

$$\frac{I\partial}{y\partial} = {}_{y}G \quad , \frac{I\partial}{x\partial} = {}_{x}G$$

- اندازه گرادیان:

$$\sqrt{{}_{y}^{2}G_{x}^{2} + G} = |I\nabla|$$

---

5. ترکیب ویژگی‌ها

- ویژگی‌های استخراج‌شده (لبه‌ها، گوشه‌ها، گرادیان شدت) با یکدیگر ترکیب می‌شوند:

$$|I\nabla| + \text{Corners} + \text{Edges} = \text{Combined Features}$$

---

6. ساخت فیلتر سفارشی

- ویژگی‌های ترکیب‌شده به اندازه هسته ($kernel$) تغییر اندازه داده می‌شوند.
- مقدار هر عنصر هسته با استفاده از مقادیر میانگین ردیف‌ها و ستون‌های ویژگی‌ها پر می‌شود.
- فیلتر نهایی نرمال‌سازی می‌شود:

$$\frac{\text{Kernel} - \mu}{\max(|\text{Kernel}|)} = \text{Kernel Normalized}$$

که در آن μ میانگین مقادیر هسته است.

و در نهایت فیلتر ساخته شده با عکس کانوالو میشود.