سیگنال ها و سیستم ها – تکلیف کامپیوتری ۳ (پیاده سازی متلب)

محمدرضا امینی - ۹۷۲۴۳۰۷۶

از کانولوشن دو بعدی برای پیدا کردن پترن‌ها در تصاویر استفاده می‌شود. گاهی این پترن‌ها از پیش تعریف شده هستند مانند فیلتر تشخیص لبه‌های افقی و گاهی پترن‌ها باید در یک فرایند آموزش داده شوند مانند فیلترهای آموزش داده شده در شبکه‌های عصبی کانولوشنی[1]. در شبکه‌های عصبی کانولوشنی به طور گسترده از مفهوم فیلتر در حوزه فضا استفاده می‌شود. اگر F فیلتر یا کرنل باشد و Y تصویر دوبعدی تک کاناله باشد، پیکسل سطر i و ستون j تصویر بعد از اعمال کانولوشن از رابطه زیر بدست می‌آید.

$$(Y * F)[i,j] = \sum_{n}\sum_{m} Y[i-n, j-m]F[n,m]$$

عبارت بالا بیان می‌کند که یک پیکسل از تصویر بعد از کانولوشن حاصل جمع حاصل ضرب درایه‌های متناظر یک فیلتر و یک پنجره از تصویر اصلی است. برای پیدا کردن پیکسل دیگر از تصویر بعد از اعمال فیلتر بایستی فیلتر را روی پنجره دیگری از تصویر اولیه قرار داد و محاسبات را انجام داد. بنابراین کل فرایند کانولوشن در تصویر را می‌توان قرار دادن یک فیلتر بر روی تصویر و سپس جابه‌جایی آن فیلتر و انجام محاسبات لازم دانست.

در این تمرین ما یک تابع متلب برای انجام کانولوشن ایجاد کردیم. این تابع conv۲D(img_path, stride, is_same, filter) است. ورودی‌های این تابع به ترتیب زیر است.

- img_path: مسیر تصویر است مانند input_images/lecun.jpg
- Stride: یک عدد که تعیین کننده تعداد پرش‌های درایه در مرحله بعد جابه‌جایی فیلتر بر روی تصویر است مانند ۲
- is_same: یک پارامتر است که می‌تواند true یا false باشد. اگر true باشد و stride=۱ سایز تصویر بعد از اعمال فیلتر و قبل از اعمال فیلتر یکی است.
- filter: ماتریس فیلتر اعمالی بر تصویر است.

بنابراین یک فراخوانی این تابع به صورت زیر است.

conv۲D('input_images/lecun.jpg', ۲, false, filter)

[1] Convolution neural networks (CNNs)

که قبل از آن بایستی ماتریس filter تعریف شده باشد.

این کد ابتدا تصویر را از دیسک می‌خواند و آن را به سیاه-سفید تبدیل می‌کند. در ادامه بر حسب این‌که روش same یا valid مدنظر است سایز تصویر خروجی محاسبه می‌شود و برای هر پیکسل این تصویر محاسبات بیان شده در بالا انجام می‌شود. در پایان نیز بر اساس ورودی‌های داده شده به این تابع، عنوان فیلتر وارد شده تعیین می‌شود و با فرمت عنوان شده در صورت سوال تصویر در پوشه output_images ذخیره می‌شود. برای مثال بالا نام تصویر خروجی در صورتی که ماتریس Gaussian به تابع داده شود به صورت زیر است.

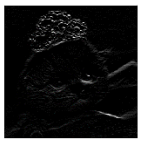
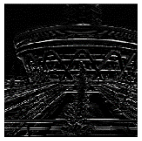
lecun_gaussian_valid_۲.jpg

برای پیاده‌سازی دستورات صورت سوال کد main.m توسعه داده شده است. این کد محتویات پوشه input_images را می‌خواند و برای تک تک تصاویر با فرت jpg در این پوشه، تصویر خروجی با ۴ سناریو فیلتر کردن خواسته شده را محاسبه می‌کند. این کد همچنین نتایج را در یک تصویر به صورت خلاصه به تصویر می‌کشد.

سناریوهای خواسته شده به شرح زیر هستند:

- Sharpness filter (valid/stride=۱)
- Horizontal Edge filter (same/stride=۲)
- Embossing filter (valid/stride=۳)
- Gaussian filter (same/stride=۱)

نتایج آزمایشات انجام شده در سناریوهای خواسته شده به همراه ابعاد بدست آمده در شکل زیر قابل مشاهده است.

kitty.jpg					
	Size: 415×415	Size: 413×413	Size: 208×208	Size: 138×138	Size: 415×415
lecun.jpg					
	Size: 377×377	Size: 375×375	Size: 189×189	Size: 125×125	Size: 377×377
mask.jpg					
	Size: 415×415	Size: 413×413	Size: 208×208	Size: 138×138	Size: 415×415
milad.jpg					
	Size: 415×415 Original	Size: 413×413 Sharpness	Size: 208×208 Horizontal Edge	Size: 138×138 Embossing	Size: 415×415 Gaussian