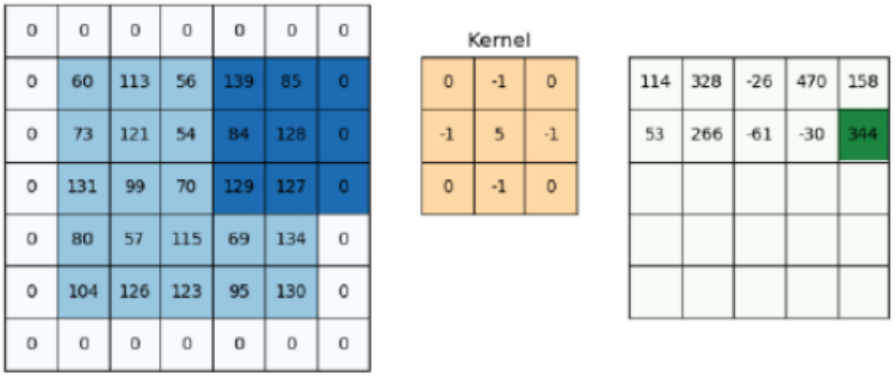
سیگنال‌ها و سیستم‌ها - دکتر سلیمی‌بدر

امیرحسین منصوری - ۹۹۲۴۳۰۶۹ - تمرین کامپیوتری سری ۳

پیاده‌سازی

برای پیاده‌سازی تابع `conv2d`، ابتدا به کمک کتابخانه Pillow عکس را باز کرده و آن را سیاه و سفید می‌کنیم تا تنها با یک کانال رنگی کار کنیم. سپس اطلاعات عکس را به یک آرایه numpy تبدیل می‌کنیم. همچنین با استفاده از طول و عرض فیلتر و عکس و همچنین پارامتر `is_same`، مقدار padding size را محاسبه می‌کنیم و با استفاده از تابع `np.pad`، به مقدار لازم مقدار صفر را به دور تا دور آرایه تصویر اضافه می‌کنیم (اگر مقدار `is_same` برابر False باشد، مقدار padding size را صفر قرار می‌دهیم تا مقداری به دور آرایه اضافه نشود). همچنین پس از محاسبه اندازه عکس خروجی، آرایه‌ای با همین اندازه و با مقدار اولیه صفر می‌سازیم تا در ادامه آن را پر کنیم.

حال برای محاسبه هر عضو از آرایه خروجی، عمل کانوولوشن را انجام می‌دهیم. به این صورت که قسمتی از آرایه ورودی که قرار است در محاسبه کانوولوشن استفاده شود (قسمت مشخص شده با آبی پررنگ) را در یک متغیر جدید ذخیره می‌کنیم، و سپس هر عضو فیلتر داده شده را عضو به عضو در این قسمت از آرایه ضرب می‌کنیم تا به آرایه‌ای جدید برسیم. این کار به سادگی با عملگر ضرب آرایه numpy قابل انجام است. جمع آرایه به دست آمده، حاصل کانوولوشن است.



همچنین دقت داریم که ممکن است حاصل کانوولوشن از صفر کمتر، یا از ۲۵۵ بیشتر شود. در صورتی که حاصل از صفر کمتر باشد، مقدار صفر و در صورتی که حاصل از ۲۵۵ بیشتر شود، مقدار ۲۵۵ را به عنوان حاصل در نظر می‌گیریم (به عبارتی، حاصل را بین صفر و ۲۵۵ clamp می‌کنیم).

در قدم آخر، آرایه به دست آمده را به صورت یک شی `PIL.Image` درآورده و آن را در فایل ذخیره می‌کنیم تا فایل عکس مورد نظر ساخته شود.

سپس همه فایل‌های jpg که در فولدر `input_images` شده قرار دارند و همچنین به ازای هر عکس، همه فیلترهای مطلوب سوال را به همراه نام فایل جدید به تابع می‌دهیم تا عکس‌های جدید ساخته شوند. خروجی در فولدر `output_images` قرار خواهد گرفت.