به نام خدا

نام: امیرمسعود

نام خانوادگی: شاکر

شماره دانشجویی: 97243081

تمرین چهارم

الف) تابع conv2D در ورودی خود img, filters, stride, padding را دریافت میکند و feature map را به عنوان خروجی میدهد.

متغیر f برابر سایز فیلتر(ها) است.

متغیر no\_filter برابر تعداد فیلترهاست.

به کمک شرط if و تابع آماده strcmp چک میشود که اگر مقدار padding برابر با same بود (یعنی نیاز به zero padding داشتیم)، سایز padding حساب شده و با استفاده از تابع آماده padarray، zero padding به عکس اعمال شود.

در خط های بعد heigt, weight عکس و همچنین تعداد چنل ها مقدار دهی میشوند.

سپس در قسمت اصلی تابع، با استفاد از چهار for تو در تو، feature map را محاسبه میکنیم.

به این صورت که یک حلقه روی تعداد فیلترها، یک حلقه روی تعداد چنل ها و دو حلقه تو در تو از 1 تا h-f+1 و از 1 تاw-f+1 میزنیم و در هر پیمایش به اندازه stride، مقدار متغیر حلقه را زیاد میکنیم.

سپس محاسبات لازم را با توجه به فرمول کانوولوشن و با استفاده از ضرب نقطه ای انجام داده و مقدار محاسبه شده را به double، cast میکنیم.

چون میتواند مقادیر غیر صحیح بگیرد.

سپس متغیر های x,y محاسبه میشوند که مختصات هر خانه از جدول خروجی را نشان میدهند.

در نهایت feature map را مقدار داده و با sum(conv, ‘all’) جمع میزنیم.

پس از پایان این چهار حلقه تو در تو، feature\_map را مجددا به unsigned int، cast میکنیم.

Conv2D:



اسکریپت main شامل تعدادی فیلتر است که قرار است روی عکس های ما اعمال شوند.

پس از clc, clear، ابتدا با دستور imread، عکس ها را میخوانیم.

در ادامه در خط 43، متغیر های filters2, filters3 حاصل concat 2 و 3 فیلتر هستند که برای عکس های دوم و سوم مورد استفاده قرار خواهند گرفت.

سپس تابع conv2D نوشته شده را برای هر کدام از عکس ها با توجه به ورودی های مد نظر که در صورت تمرین آمده بود، صدا میزنیم و خروجی ها را به دست می آوریم.

در نهایت برای هر کدام از عکس ها، با استفاده از دستور subplot، عکس اصلی و عکس (های) فیلتر شده را رسم میکنیم.

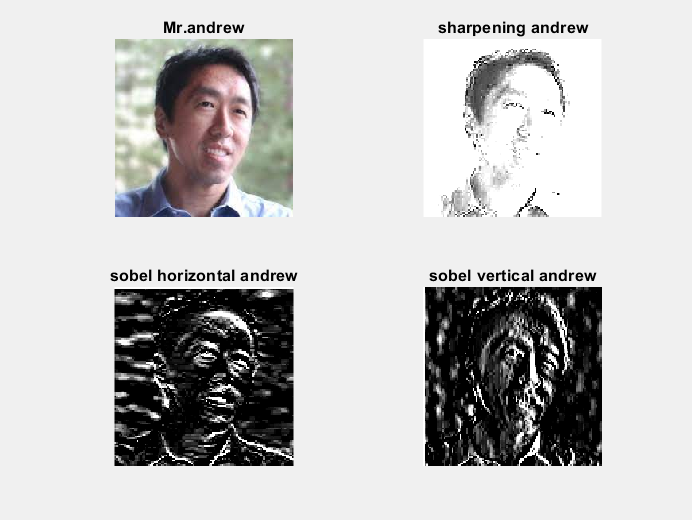
Main:

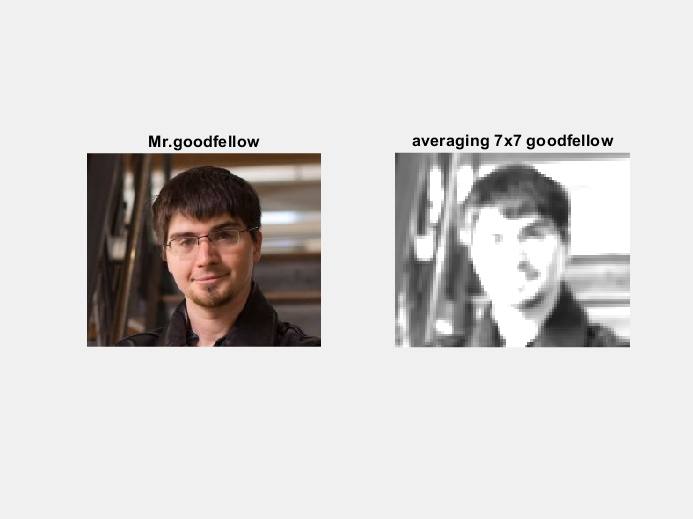


ب) چهار عکس به دست آمده به کمک دستور subplot:





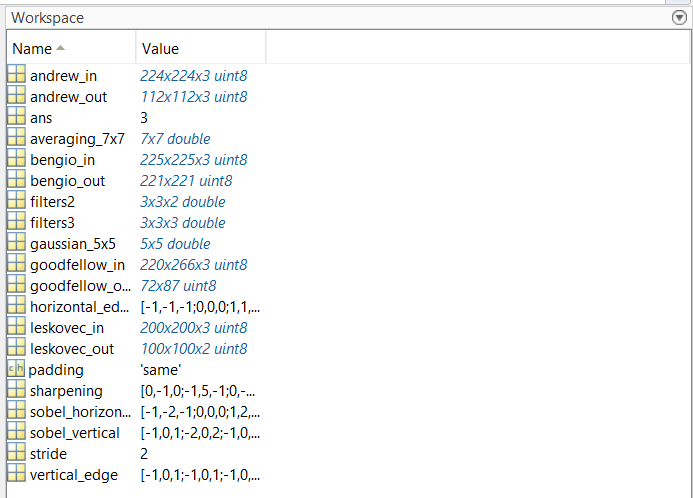




ج)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| output size | padding size | input size | img |  |
| 221\*221 | 0 | 225\*225\*3 | bengio.jpeg | 1 |
| 100\*100\*2 | 1 | 200\*200\*3 | leskovec.jpeg | 2 |
| 112\*112\*3 | 1 | 224\*224\*3 | andrew.jpeg | 3 |
| 72\*87 | 0 | 220\*266\*3 | goodfellow.jpeg | 4 |

جدول بالا از روی مشخصات متغیر های تعریف شده در کد به دست آمده است:



با توجه به اینکه در عکس های bengio, goodfellow مقدار padding برابر valid است، zero padding نداریم و بنابراین padding size برابر صفر است.

همچنین در عکس های leskovec, Andrew که zero padding داریم، سایز فیلتر(ها) برابر 3 است که با استفاده از فرمول p = (f-1)/2 به مقدار 1 برای padding size میرسیم.

محاسبه دستی سایز خروجی ها:

1. در عکس bengio مقادیر زیر را داریم:

m = 225, n = 225, p = 0, f = 5, s = 1, no\_fiter = 1

-> (((m + 2p – f) / s) + 1) \* (((n + 2p – f) / s) + 1) = (((225 + 0 – 5) / 1) + 1) \* (((225 + 0 – 5) / 1) + 1) = 221 \* 221

2. در عکس leskovec مقادیر زیر را داریم:

m = 200, n = 200, p = 1, f = 3, s = 2, no\_filter = 2

-> (((m + 2p – f) / s) + 1) \* (((n + 2p – f) / s) + 1) \* no\_filter = (((200+ 2 – 3) / 2) + 1) \* (((200 + 2 – 3) / 2) + 1) \* 2 = 100 \* 100 \* 2

3. در عکس Andrew مقادیر زیر داریم:

m = 224, n = 224, p = 1, f = 3, s = 2, no\_filter = 3

-> (((m + 2p – f) / s) + 1) \* (((n + 2p – f) / s) + 1) \* no\_filter = (((224+ 2 – 3) / 2) + 1) \* (((224 + 2 – 3) / 2) + 1) \* 3 = 112 \* 112 \* 3

4. در عکس goodfellow مقادیر زیر را داریم:

m = 220, n = 266, p = 0, f = 7, s = 3, no\_filter = 1

-> (((m + 2p – f) / s) + 1) \* (((n + 2p – f) / s) + 1) = (((220 + 0 – 7) / 3) + 1) \* (((266 + 0 – 7) / 3) + 1) = 72\*87