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 تا w-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:

```
function [feature_map] = conv2D(img,filters, stride, padding)
f = size(filters, 1);
% size of filters
no_filter = size(filters, 3);
% number of filters
if strcmp(padding, 'same')
    % check if our padding has value 'same'
    p = ((f-1)/2);
    img = padarray(img, [p p]);
    % do zero padding before convulution
    % using padarray built in function
end
h = size(img, 1);
% height of input image
w = size(img, 2);
% weight of input image
ch = size(img, 3);
% number of channels of the image
% if image is rgb, it equals 3
% and if image is gray, it equals 1
feature_map = zeros(floor((h-f)/stride)+1, floor((w-f)/stride)+1);
for num = 2 : no_filter
    feature_map(:, :, num) = 0;
end
% fill our output with zeros
% we're gonna calculate the real values and add them to these zeros
% during the convultion
for i = 1 : no_filter
    % iterate on each filter
    for j = 1 : ch
        % iterate on each channel
        channel_i = img(:, :, j);
        % channel i of image
        for row = 1 : stride : h-f+1
            for col = 1 : stride : w-f+1
                local = channel_i(row:row+f-1, col:col+f-1);
                conv = double(local) .* filters(:, :, i);
                % calculate convulution according to its formula
                % cast it to double cause it can get non integer values
                x = ceil(row/stride);
                y = ceil(col/stride);
                % calculate the coordinates of our output image
                feature_map(x, y, i) = feature_map(x, y, i) + sum(conv, 'all');
                % assign each part of our output image
            end
        end
    end
end
feature_map = uint8(feature_map);
% cast our output back to unsigned int
end
```

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:

```
clc;
clear;

%%%%%%%%%%% READ THE INPUT IMAGES %%%%%%%%%%%
bengio_in = imread('images/bengio.jpeg');
% 1. bengio image
leskovec_in = imread('images/leskovec.jpeg');
% 2. leskovec image
andrew_in = imread('images/andrew.jpeg');
% 3. andrew image
goodfellow_in = imread('images/goodfellow.jpeg');
% 4. goodfellow image

%%%%%%%%%%% DEFINE FILTERS %%%%%%%%%%%
horizontal_edge = [[-1,-1,-1];
                   [0,0,0];
                   [1,1,1]];

vertical_edge = [[-1,0,1];
                 [-1,0,1];
                 [-1,0,1]];

sharpening = [[0,-1,0];
              [-1,5,-1];
              [0,-1,0]];

sobel_horizontal = [[-1,-2,-1];
                    [0,0,0];
                    [1,2,1]];
sobel_vertical = [[-1,0,1];
                  [-2,0,2];
                  [-1,0,1]];

gaussian_5x5 = (1/273)*[[1,4,7,4,1];
                        [4,16,26,16,4];
                        [7,26,41,26,7];
                        [4,16,26,16,4];
                        [1,4,7,4,1]];

averaging_7x7 = (1/49)*ones(7,7);

% concatening filters to build the `filters` tensor
filters2 = cat(3, horizontal_edge, vertical_edge);
filters3 = cat(3, sharpening, sobel_horizontal, sobel_vertical);
%%%%%%%%%%% PARAMETERS SETTING %%%%%%%%%%%
stride = 2;
padding = 'same';
%%%%%%%%%%% YOUR CODE HERE %%%%%%%%%%%
bengio_out = conv2D(bengio_in, gaussian_5x5, 1, 'valid');
leskovec_out = conv2D(leskovec_in, filters2, 2, 'same');
andrew_out = conv2D(andrew_in, filters3, 2, 'same');
goodfellow_out = conv2D(goodfellow_in, averaging_7x7, 3, 'valid');
% 1. bengio
figure(1);
subplot(1, 2, 1), imshow(bengio_in);
title('Mr.bengio');
subplot(1, 2, 2), imshow(bengio_out);
title('gaussian 5x5 bengio');
% 2. leskovec
figure(2);
subplot(2, 2, 1), imshow(leskovec_in);
title('Mr.leskovec');
subplot(2, 2, 2), imshow(leskovec_out(:, :, 1));
```

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

Mr.leskovec

horizontal edge leskovec

vertical edge leskovec

Mr.andrew

sharpening andrew

sobel horizontal andrew

sobel vertical andrew

Mr.goodfellow

averaging 7x7 goodfellow

ج)

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

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

Workspace

Name	Value
andrew_in	224x224x3 uint8
andrew_out	112x112x3 uint8
ans	3
averaging_7x7	7x7 double
bengio_in	225x225x3 uint8
bengio_out	221x221 uint8
filters2	3x3x2 double
filters3	3x3x3 double
gaussian_5x5	5x5 double
goodfellow_in	220x266x3 uint8
goodfellow_o...	72x87 uint8
horizontal_ed...	[-1,-1,-1;0,0,0;1,1,...
leskovec_in	200x200x3 uint8
leskovec_out	100x100x2 uint8
padding	'same'
sharpening	[0,-1,0;-1,5,-1;0,-...
sobel_horizon...	[-1,-2,-1;0,0,0;1,2,...
sobel_vertical	[-1,0,1;-2,0,2;-1,0,...
stride	2
vertical_edge	[-1,0,1;-1,0,1;-1,0,...

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