

## گزارشکار مربوط به استخراج ویژگی به کمک عملگر کانولوشن 2بعدی

ابتدا تابع  $\text{cinv2D}$  توضیح داده می‌شود:

تابع  $\text{cinv2D}$  دارای 4 ورودی است.  $x$  تصویر ورودی مان است.  $h$  بیانگر مجموعه فیلترهای ما است.  $\text{padding}$  که مقدار آن می‌تواند  $\text{same}$  یا  $\text{valid}$  باشد.  $\text{Stride}$  که یک عدد طبیعی است. حال به توضیح تابع می‌پردازیم:

ابتدا  $h$  را 90 درجه می‌چرخانیم. سپس برای انجام عملیات ریاضی بر روی تصویر، تصویر را به  $\text{double}$  تبدیل می‌کنیم. در مرحله بعد به ترتیب سائز فیلتر و سائز تصویر را بدست می‌آوریم.  $ch$  در سائز فیلتر، تعداد کانال‌های خروجی را به ما می‌دهد.  $Chi$  در سائز تصویر، تعداد کانال تصویر را به ما می‌دهد که اگر تصویر رنگی باشد مقدار آن 3 و اگر تصویر خاکستری باشد مقدار آن یک است. حال اگر مقدار  $\text{padding}$  ما  $\text{valid}$  باشد، طبق گفته صورت سوال ما عملیات  $\text{zeropadding}$  را قبل از کانولوشن انجام نخواهیم داد. سپس با استفاده از رابطه زیر ابتدا سائز خروجی را به دست می‌آوریم:

$$\text{sizey} = \text{floor}(([\text{Mx}, \text{Nx}] - [\text{Mh}, \text{Nh}]) / \text{stride} + 1);$$

حال اگر مقدار  $\text{padding}$  ما  $\text{same}$  باشد، طبق گفته صورت سوال ما عملیات  $\text{zeropadding}$  را قبل از کانولوشن انجام خواهیم داد. در این حالت سائز خروجی ابعاد یکسانی دارد که در کد مشخص شده است. حال باید تصویر را  $\text{padding}$  کنیم. ابتدا از رابطه زیر مقدار  $\text{padding}$  را بدست می‌آوریم و سپس عملیات  $\text{zeropadding}$  را بر روی تصویر  $x$  اعمال می‌کنیم.

پس از اجرای عملیات  $\text{zeropadding}$  بر روی تصویر  $x$ ، با استفاده از چند حلقه  $\text{for}$  که بر روی سطر و ستون‌های خروجی اعمال می‌شود، درایه‌های خروجی را بدست می‌آوریم. حال اگر مقدار  $\text{padding}$  ما  $\text{same}$  باشد، عملیات بر روی یک سمت تصویر شروع می‌شود و طبق  $\text{Stride}$  تعریف شده، پنجره هایمان را جلو می‌بریم و در  $h$  ضرب می‌کنیم و سپس بر روی تمامی پنجره‌ها مجموع گرفته می‌شود.

حال اگر مقدار  $\text{padding}$  ما  $\text{same}$  باشد، ابتدا با استفاده از ابعاد تصویر ما یک ناحیه برای  $m$  و  $n$  تشکیل می‌دهیم. که  $m$  محور عمودی و  $n$  محور افقی بر روی تصویر در نظر گرفته می‌شود. ابتدا نقطه ای را به عنوان نقطه مبدا در نظر می‌گیریم. در این رابطه  $k1$  و  $k2$  مختصات مرکز پنجره ما هستند. حال مانند حالت قبلی، پنجره هایمان را جلو می‌بریم و در  $h$  ضرب می‌کنیم و سپس بر روی تمامی پنجره‌ها مجموع گرفته می‌شود.

```

function [y,p] = conv2D(x,h,padding,stride)

h = rot90(h,2);
x = double(x);
[Mh, Nh ,ch] = size(h);
[Mx, Nx ,chi] = size(x);
if strcmp(padding,'valid')
    sizey = floor(([Mx, Nx]-[Mh, Nh])/stride+1);
    y = zeros(sizey(1),sizey(2), ch);
    p=0;
elseif strcmp(padding,'same')
    sizey = [Mx, Nx];
    y = zeros(sizey(1),sizey(2), ch);
    p = ([Mx, Nx]*(stride-1)+[Mh, Nh]-stride)/2;
    p=ceil(p);
    x = [zeros(Mx+2*p(1),p(1),chi),[zeros(p(2),Nx,chi)...
        ;x;zeros(p(2),Nx,chi)],zeros(Mx+2*p(1),p(1),chi)];
end

for k1 = 1:sizey(1)
    for k2 = 1:sizey(2)
        for k3 = 1:ch
            x_temp = x((k1-1)*stride+(1:Mh),(k2-1)*stride+(1:Nh),:);
            h_temp = repmat(h(:, :, k3), 1, 1, chi);
            y(k1,k2,k3) = sum(x_temp.*h_temp,'all');
        end
    end
end
end

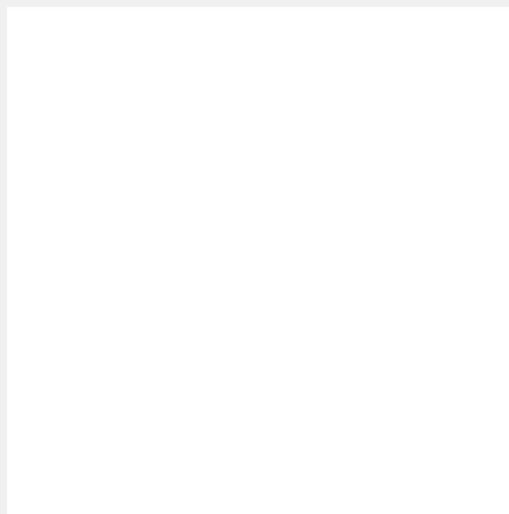
```

حال با توجه به فایل main که به ما داده شده است و در آن نوع فیلتر و مقدار داده شده است. تابع خود را آزمایش می کنیم.

ورودی‌ها:

	<i>img</i>	<i>stride</i>	<i>padding</i>	<i>filters</i>
1	bengio.jpeg	1	valid	[guassian_5x5]
2	leskovec.jpeg	2	same	[horizontal_edge, vertical_edge]
3	andrew.jpeg	2	same	[sharpening, sobel_vertical, sobel_horizontal]
4	goodfellow.jpeg	3	valid	[averaging_7x7]

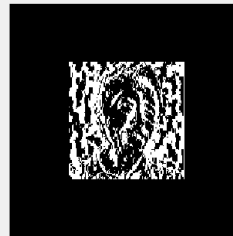
خروجی ردیف اول:



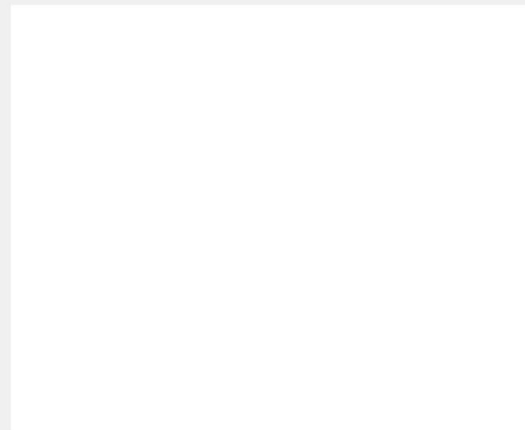
خروجی ردیف دوم:



خروجی ردیف سوم:



خروجی ردیف چهارم:



	img	Input size	Padding size	Output size
1	bengio.jpeg	225*225*3	0	221*221
2	leskovec.jpeg	200*200*3	101*101	200*200*3
3	andrew.jpeg	224*224*3	113*113	224*224*3
4	goodfellow.jpeg	220*226*3	0	72*78