

## گزارش کار تمرین چهارم کامپیوتری سیگنال

محمدرضا محمدزاده اصل

98243097

برای پیاده سازی تابع خواسته شده دو تابع دیگر نیز پیاده سازی کردم:

**addPadding:** این تابع یک آرایه دو بعدی می گیرد و با توجه به مقدار  $f$  برای آن **padding** ایجاد می کند. مقدار  $h_o$  و  $w_o$  طول عرض آرایه دو بعدی پس از پدینگ هستند که برای راحتی به تابع پاس داده می شوند.

**singleConv:** با توجه به اینکه عکس ورودی ممکن بود RGB باشد و سه کانال داشته باشد و همچنین چون تعداد فیلتر ها می توانست بیشتر از 1 باشد برای کاهش پیچیدگی اعمال کردن یک فیلتر روی یک آرایه دو بعدی را با این تابع پیاده سازی کردم. آن را به اندازه تعداد کانال های عکس ورودی \* تعداد فیلتر های خواسته شده صدا می کنم.

در پیاده سازی تابع **conv2D** از دو تابع ذکر شده استفاده کردم. معنای متغیر های به کار رفته در کد به این صورت است:

**h:** ارتفاع عکس ورودی

**w:** پهنای عکس ورودی

**f:** طول و عرض فیلتر

**h<sub>o</sub>:** ارتفاع عکس نهایی

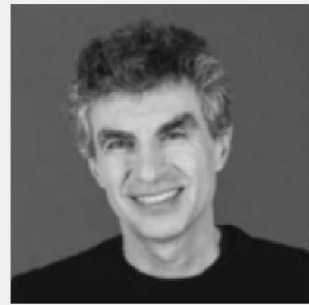
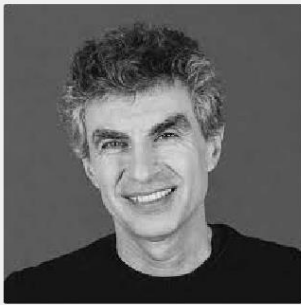
**w<sub>o</sub>:** پهنای عکس نهایی

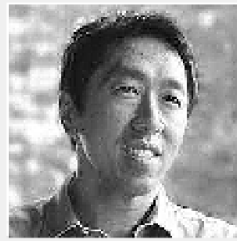
**fimg:** کانال  $z$  ام عکس پس از اعمال پدینگ (در صورت نیاز)

همچنین در نهایت برای چاپ هر فیلتر های مربوط به هر عکس از تابع **showAns** استفاده می کنم که **subplot** با 2 سطر و (1+تعداد فیلتر) ستون دارد.

سطر اول مربوط به عکس رنگی و سطر دوم مربوط به عکس سیاه سفید است.

چپ ترین عکس خود عکس بدون فیلتر است. عکس های بعدی به ترتیب چپ به راست فیلتر های اعمال شده اند.





	img	Input size	Padding size	Output size
1	bengio.jpeg	225*225	0	221*221
2	leskovec.jpeg	200*200	1	100*100
3	andrew.jpeg	224*224	1	112*112
4	goodfellow.jpeg	h:220, w: 226	0	h: 72, w: 87

مقدار padding از رابطه زیر به دست می آید:

$$p = \frac{f - 1}{2}$$

البته در قسمت های 1 و 4 چون padding در حالت valid بود مقدار آن ها صفر شد مگر نه مقدار آن ها به ترتیب برابر 2 و 3 می شد.

فرمول سائز خروجی نیز به صورت مقابل است:

$$\left( \frac{(m+2p-f)}{stride} + 1 \right) \times \left( \frac{(n+2p-f)}{stride} + 1 \right)$$