



پروژه نهایی درس طراحی سیستم های کامپیوتری دیجیتال

محمد حسین میرزائی - 99522158

متین محمودخانی - 99522095

نام استاد : دکتر بیت اللهی

این پروژه به طور کلی به پردازش تصویر می پردازد و از چند قسمت کلی تشکیل شده که در ادامه توضیح داده خواهد شد .

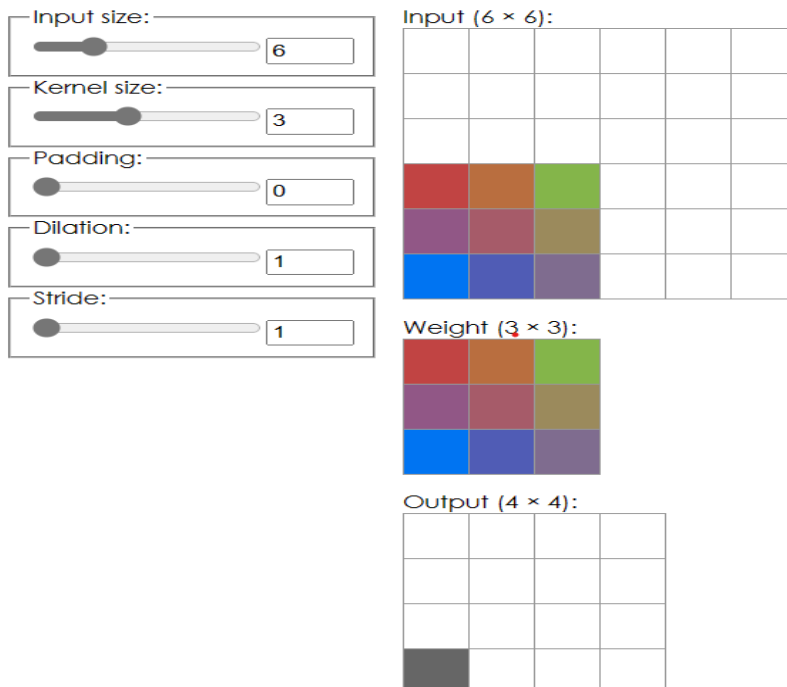
ابتدا تصویری را در نظر گرفته و آن را سیاه سفید می کنیم (مود grey scale) ، چون که تصویر های رنگی سه کاناله بوده و با تصویر سیاه سفید بررسی ما راحت تر خواهد شد .

سپس با استفاده از کتاب خانه های مورد نظر در زبانی برنامه نویسی نرم افزار نظیر پایتون ، عکس مورد بررسی را پیکسل بندی کرده و تبدیل به آرایه می کنیم ، حال آرایه بدست آمده را به عنوان داده به زبان برنامه نویسی سخت افزار برای پردازش می دهیم .

بخش سخت افزار نیز به سه بخش main , package , simulate تقسیم بندی شده است که به توضیح هر یک می پردازیم :

Package: ابتدا کتابخانه IEEE را اضافه می کنیم ، در این بخش اطلاعات کلی را ذخیره سازی و از آن ها استفاده خواهیم کرد ، به ترتیب بخش فیلتر ها که هر فیلتر از یک آرایه دو بعدی real تشکیل شده است (مثلا 3×3) ، تاییپی برای گرفتن عکس اصلی ما که با توجه به گفته سوال دو بعدی و از نوع integer و 128×128 می باشد ، و قسمت بعدی نیز عکس پردازش شده توسط فیلتر ها است که به صورت آرایه دو بعدی از نوع real می باشد و سائز آن به دلیل سائز عکس و سائز فیلتر 126×126 می باشد .

performing a correlation. however, most deep learning frameworks still can not



تصویر نظیر،
نحوه طی شده
فیلتر و دلیل
آن که عکس
پردازش شده
 126×128
می باشد را
نشان می دهد :

حال با توجه به تایپ های تعریف شده عضوی **constant** برای ورودی عکس که آرایه دو بعدی در نظر گرفته شد .

و در نهایت در بخش **package**، **function** برای گرفتن عدد **random** تعریف می شود ، در بخش **package body** نیز این **function** پیاده می شود که بدین صورت می باشد که با اضافه کردن کتاب خانه **IEEE** بخش **MATH_REAL** با فراخوانی **uniform**، **procedure** عددی انتخاب و با توجه به بازه مورد نظر عدد **random** انتخاب می شود(دقت شود که فیلتر ما شامل اعداد رندوم بین صفر و یک می باشد ، پس بازه مورد بررسی را بین صفر و یک در نظر می گیریم .)

Main: در این بخش به پیاده سازی کلی کد می پردازیم ، 1 ورودی **clock** و خروجی ما عکس پردازش شده می باشد و یک عدد **n** ، **generic** برای تعداد فیلتر ها و پیاده سازی کلی در نظر می گیریم .

حال در این بخش ورودی از تایپ **all_filters** که برای گرفتن تمام فیلتر ها می باشد ، یک **counter** از نوع **integer** و یک **result** برای تصویر های پردازشی در نظر می گیریم .

در بخش اول ، ابتدا به صورت **concurrent** و با استفاده از کلید واژه **generate** یک حلقه تو در تو ایجاد و تمام فیلتر ها را می گیریم ، در قسمت بعدی نیز با توجه به توضیحات مسئله، دو **process** پیاده می کنیم . **Process** ابتدایی به پیاده سازی ردیف به ردیف عکس پردازش شده می پردازد و به مقدار **counter** حساس می باشد و با تغییر **counter** ، ردیف های متفاوت را اجرا می کند و **convolution** این بخش نیز طبق گفته سوال به صورت درختی و با در نظر گرفتن هشت مقدار **temp** در هر بخش انجام شده است .

Process دوم نیز با توجه به **clock** عمل می کند و هر بار که ما **rising_clk** داریم مقدار **counter** را اضافه می کند ، دلیل استفاده از **clock** آن است که سطر به سطر طبق لبه بالا رونده **clock** پیش برویم و حجم عظیم داده به یک دفعه سرازیر نشود .

Simulate: در این بخش با مقدار دهی کلاک به صورت منظم و **port map** بخش **main** می توان داده های نهایی را شبیه سازی کرد(دقت شود به علت استفاده از تایپ **real** امکان سنتز کد وجود نداشته و صرفا شبیه ساز آن را در اختیار داریم .)

نمونه ای از خروجی شبیه سازی شده در صفحه بعد قابل رویت می باشد :

