# پروژه درس اسمبلی

## فاز اول

در این فاز سعی شده شما با یکی از جنبه های جالب کاربرد اسمبلی که دی اسمبل کردن برنامه ها و فهم ساز و کار آنها برای کارهایی مثل کرک کردن و ... است آشنا شوید.

در این فاز شما باید یک رمز دوبخشی را پیدا کنید. به شما دو فایل قابل اجرا در ترمینال لینوکس ($executable$) داده شده است که هر کدام از آنها یک بخش از رمز را دارند. شما باید آنها را دی اسمبل کنید (راهنمایی : از$objdump$استفاده کنید) سپس این دو کد را تحلیل کنید و ببینید برای نمایش دادن رمز به چه چیزی نیاز دارند (بهتر است قبل از شروع کردن تحلیل، برنامه ها را یک بار اجرا کنید و پیام های آنها را ببینید ). اگر با برنامه ها همراهی کنید و هر آنچه میخواهند در اختیار آنها قرار دهید، آنها هم برای شما هر دو رمز را چاپ میکنند.

پیشنهاد من به شما اینه که سعی نکنید رمزها را محاسبه کنید و برای همکاری کردن با برنامه ها تلاش کنید :)

راهنمایی: برای هردو برنامه نیاز است شما کد اسمبلی بنویسید و حتما باید به همراه فایل های خود گزارشی از روند پروژتون آپلود کنید.

اطلاعات مورد نیاز برای برنامه اول ($part1$):

$filename \quad db \quad "part1.txt"$

اطلاعات مورد نیاز برای برنامه دوم ($part2$):

تمام $chari$ ها از حروف کوچک انگلیسی تشکیل شده اند ($a, b, c, ..., z$).

$filename \quad db \quad "part2.txt"$

در هردو برنامه فرض کنید که $buff$ یک آرایه $db$ با طول مناسب است.

هرسوال یا ابهامی که راجب این فاز براتون ایجاد شد رو میتونید از من ($@alihoseini02$) بپرسید

## فاز دوم

در این فاز، شما باید یک پردازشگر تصویر پیاده‌سازی کنید. خروجی این پردازشگر به صورت یک ماتریس است که می‌توانید با استفاده از پایتون آن را نمایش دهید. تصویر ورودی شما به صورت یک ماتریسی از اعداد درمی‌آید که مقدار عددی هر پیکسل در جایگاه خود نمایش داده شده است و شما با اجرای عملیات‌ها و تبدیل‌های مختلف روی این ماتریس، پردازش‌های مدنظر را روی آن انجام می‌دهید. در ابتدای برنامه شما باید منویی از دستورات و عملیات های قابل اجرا

به کاربر نشان دهید و بنا به انتخاب کاربر، عملیات مورد نظر را اجرا کنید. این پردازش و تبدیل‌ها عبارتند از:

### باز کردن تصویر $(opening)$

در مرحله اول، از شما خواسته میشود تصویر را باز کنید. با انتخاب این گزینه توسط کاربر، آدرس تصویر مورد نظر را از ورودی بگیرید و آن را با استفاده از یک برنامه پایتون باز کنید. عکس باز شده را به ماتریس تبدیل کنید. سپس ماتریس را در یک فایل $text$ با همان نام و در همان محل ذخیره کنید (توجه کنید که ذخیره سازی شما در این مرحله دلخواه است و به صورتی که در قسمت های بعدی برای کار با آن راحت تر هستید این عملیات را انجام دهید) و در مرحله بعد، این فایل را با استفاده از زبان اسمبلی باز کنید و آن را بخوانید. از این به بعد، عملیات های گفته شده را روی این ماتریس و فقط با زبان اسمبلی انجام دهید(استفاده از پایتون در مراحل بعدی هیچ نمره ای ندارد).

### تغییر شکل $(Reshaping)$

در این مرحله، شکل ماتریس تصویر تغییر می‌کند بدون اینکه مقادیر پیکسل‌ها تغییر کنند. این عملیات معمولاً برای آماده‌سازی داده‌ها برای مدل‌های یادگیری ماشین یا پردازش‌های خاص استفاده می‌شود. به عنوان مثال، وقتی داده‌ی تصویری ما، ۵ بعد دارد و ما تنها دو بعد از آن را نیاز داریم، تنها دو بعد اول آن را به صورت دست‌نخورده نگه داشته و سه بعد دیگر را حذف می‌کنیم و یک آرایه‌ی دوبعدی خروجی می‌دهیم.

ورودی: تعداد ابعاد جدید
خروجی: ماتریس تصویر با ابعاد جدید
در صورتی که اجرای عملیات ممکن نبود، خطای مناسب را چاپ کنید.
مثال: تغییر شکل ماتریس سه بعدی به یک ماتریس دو بعدی یا تک بعدی

### تغییر اندازه $(Resizing)$

در این مرحله، اندازه تصویر با توجه به ابعاد مشخص شده تغییر می‌کند. این عملیات می‌تواند باعث کاهش یا افزایش تعداد پیکسل‌ها شود و تصمیم اینکه پیکسل‌ها چگونه تغییر پیدا کنند که تصویر مقصد شکل کلی خود را حفظ کند و تغییر اندازه به بهترین شکل انجام شود، به عهده‌ی شما خواهد بود.

ورودی: ماتریس تصویر با اندازه اولیه و سایز مقصد.
خروجی: ماتریس تصویر با اندازه جدید.
مثال: تغییر اندازه تصویر از ۱۰۰ در ۱۰۰ به ۵۰ در ۵۰. یکی از روش های پیشنهادی برای این قسمت استفاده از الگوریتم $nearestneighborinterpolation$ است.

### تبدیل به مقیاس خاکستری $(Grayscaling)$

در این مرحله، تصویر رنگی به تصویر مقیاس خاکستری تبدیل می‌شود، که تمامی مقادیر پیکسل‌ها را نسبت به یکدیگر و رنگ‌های سفید و سیاه، اسکیل می‌کند. شیوه ای که برای این مرحله عموما استفاده میشود، بر اساس $luminanceperception$ چشم انسان و با فرمول زیر است:

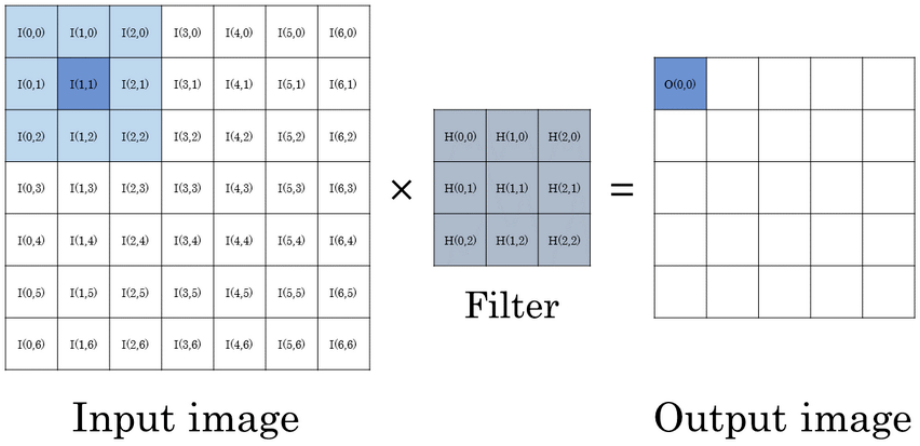
$$Gray = 0.299 \times Red + 0.587 \times Green + 0.114 \times Blue$$

ورودی: ماتریس تصویر رنگی.
خروجی: ماتریس تصویر در مقیاس خاکستری.
مثال: تبدیل یک داده رنگی به تصویر خاکستری برای یادگیری بهتر.

### فیلترهای کانولوشن $(ConvolutionFilters)$

در این مرحله، فیلترهای کانولوشن برای استخراج ویژگی‌های خاص از تصویر اعمال می‌شوند.

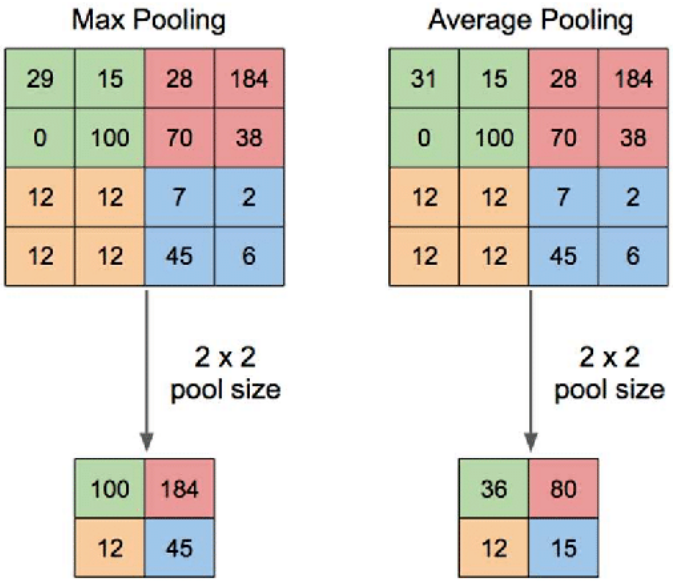
این عملیات معمولاً در پردازش تصویر و شبکه‌های عصبی کانولوشن $(CNN)$ استفاده می‌شود. این فیلترها شامل فیلترهای لبه‌یاب یا $EdgeDetection$، فیلترهای محوکننده یا $GaussianBlur$، فیلترهای تیزکننده یا $Sharpening$ و فیلترهای برجسته‌سازی یا $Emboss$ هستند. به دلخواه دو تا از این فیلترها را با استفاده از روش دلخواهتان، پیاده‌سازی کنید. نمونه‌ای کلی از این فیلترها به این صورت است:



ورودی: ماتریس تصویر و فیلتر کانولوشن مدنظر (کرنل).
خروجی: ماتریس تصویر فیلتر شده.
مثال: اعمال فیلتر لبه‌یاب برای تشخیص لبه‌های تصویر.
امتیازی: در این بخش میتوانید $zeropadding$ و $stride$ با اندازه های متفاوت را نیز به گزینه های قابل اعمال اضافه کنید.

## ادغام $(Pooling)$

در این مرحله، عملیات ادغام برای کاهش ابعاد تصویر و تمرکز بر ویژگی‌های مهم انجام می‌شود. این عملیات معمولاً بعد از فیلترهای کانولوشن در شبکه‌های عصبی استفاده می‌شود و مانند آن‌ها، در ماتریس به صورت ناحیه‌ای اعمال می‌شود و انواع مختلفی دارد. به عنوان مثال، ادغام حداکثر یا $MaxPooling$، تصویر را به ناحیه‌های کوچک تقسیم کرده و سپس در هر ناحیه، مقدار بیشینه را انتخاب می‌کند و در ماتریس نهایی ثبت می‌کند.
به طور مشابه ادغام میانگین یا $AveragePooling$ نیز در هر ناحیه، میانگین را محاسبه کرده و آن را در ماتریس نهایی جایگزین درایه مربوط به این ناحیه می‌کند:




ورودی: ماتریس تصویر و ادغام مورد نظر و اندازه آن

خروجی: ماتریس تصویر ادغام شده.
مثال: اعمال ادغام حداکثر $(MaxPooling)$ برای کاهش اندازه تصویر از ۴ در ۴ به ۲ در ۲.

## افزودن $noise$

در این مرحله با استفاده از روش $Salt-and-Pepper$ به تصویر داده شده $noise$ اضافه کنید.

امتیازی: روش $GaussianNoise$ را پیاده سازی کنید.
ورودی: ماتریس تصویر
خروجی: ماتریس تصویر دارای $noise$

## نمایش خروجی

در مرحله آخر، ماتریس تصویر نهایی را در یک فایل ذخیره کنید. سپس با استفاده از پایتون، فایل نهایی را باز کنید و تصویر بدست آمده را نشان دهید.