

به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی برق



دکتر عمادالدین فاطمی زاده - پردازش تصاویر دیجیتال

نیم سال دوم ۱۴۰۱-۱۴۰۰

## تمرین عملی سری پنجم

**لطفاً به نکات زیر توجه بفرمایید: (رعایت نکردن این موارد باعث کاهش نمره می شود.)**

۱. نتایج و پاسخ های خود را در یک فایل با فرمت zip به نام HW5-Name-StudentNumber در سایت Quera قرار دهید. همچنین فایل پایتون یا متلب خود را به همان نام در قسمت مخصوص به خود آپلود کنید.
۲. کسب نمره کامل در هر سوال مستلزم تحویل کدها (۴۰ نمره) و توضیحات (۳۰ نمره) و نتایج (۳۰ نمره) می باشد.
۳. کدهای شما تماماً باید توسط خودتان نوشته شده باشند. هرگونه استفاده از کد دیگران، اعم از دوستان و اینترنت، به هر شکل ممکن، تقلب محسوب می شود و نمره تمام تمرینات جاری و تمام تمرینات قبلی صفر خواهد شد. با اجرای این کدها باید همان نتایجی که فرستاده اید قابل بازیابی باشند. برنامه شما باید به گونه ای باشد که بدون نیاز به هیچ تغییری قابل اجرا باشد، در غیر این صورت هیچ نمره ای تعلق نخواهد گرفت.
۴. برای تمام سؤالات، باید جزئیات روشی که استفاده کرده اید را توضیح دهید و نتایجی که گرفته اید را ارائه دهید. این توضیحات می تواند در یک فایل pdf و یا در یک فایل ipynb باشد. در توضیحات، باید اشاره کامل به کارهایی که انجام داده اید بنمایید به طوری که یک شخص آگاه از موارد درس بتواند به آسانی متوجه کاری که شما انجام داده اید شود.
۵. در طول ترم امکان ارسال با تاخیر پاسخ همه ی تمرین تا سقف شش روز و در مجموع بیست و یک روز وجود دارد. پس از گذشت این مدت، پاسخ های ارسال شده پذیرفته نخواهند بود. همچنین، به ازای هر روز تأخیر غیر مجاز بیست درصد از نمره تمرین به صورت روزانه کسر خواهد شد.
۶. اگر از Jupyter notebook استفاده می کنید، میتوانید خروجی ها را پاک کنید تا حجم فایل تحویلی زیاد نشود.
۷. مهلت تحویل: ۲۳ خرداد ساعت ۲۳:۵۹
۸. نام طراح هر سوال در زیر آن نوشته شده است و شما میتوانید سوالات خود را از طریق ایمیل یا تلگرام از طراح سوال بپرسید.

ارسلان فیروزی: @Arsalanfiroozi - Arsalan.firoozi@gmail.com

سید سعید رضوی: @RazooIs - Saeedrazavi890@gmail.com

امیرحسین جوادی: @Amirhosein\_javadi - Javadiamirhosein.2000@gmail.com

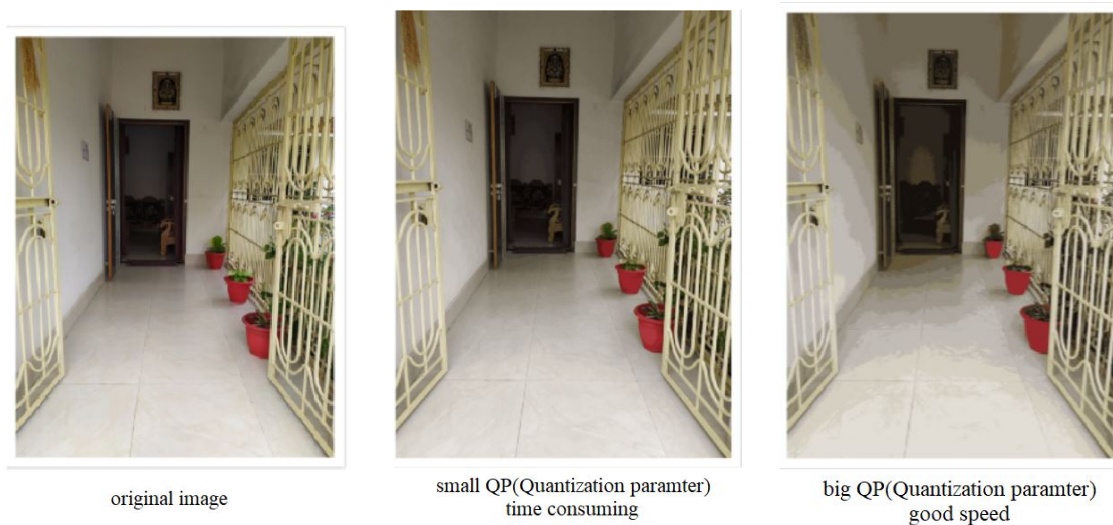
## ۱ Image Compression

طراح: سید سعید رضوی

در این سوال قصد کاهش حجم عکس (compression) و بخش بندی (clustering) آن را به طور همزمان داریم. ابتدا تصویر q1.jpg را بخوانید و هیستوگرام مربوط به هر کانال را رسم و به ترتیب با نام‌های hist\_b.jpg، hist\_g.jpg، hist\_r.jpg ذخیره نمایید. سپس با توجه به داده‌های هر هیستوگرام، مقادیر بین 0 تا 255 را با یک مقدار مشخص کوانتیزیشن (این مقدار در حقیقت همان طول تقسیم بندی می‌باشد)، تقسیم بندی (quantize) کنید. برای مثال اگر مقدار کوانتیزیشن برابر 32 باشد، 8 بازه داریم و تمام مقادیری که در یک بازه مشخص هستند دقیقاً به وسط آن بازه ارجاع داده می‌شوند. این کار را برای تمام پیکسل‌ها و بر روی تمامی کانال‌ها انجام دهید و تصویر بخش بندی شده را با نام clustered.jpg ذخیره نمایید. هیستوگرام تصویر جدید را برای کانال‌های مختلف با نام‌های hist\_b\_clus.jpg، hist\_g\_clus.jpg، hist\_r\_clus.jpg ذخیره کنید. همچنین حجم اشغال شده تصویر جدید را با تصویر q1.jpg مقایسه کنید و علش را به صورت دقیق در گزارش ذکر کنید.

(دقت شود اگر مقدار کوانتیزیشن رو از حدی بیشتر بگیرید تصویری نهایی با تصویر اصلی بسیار متفاوت می‌شود و اگر این مقدار را از حدی کمتر بگیرید نتیجه شما همچنان پر حجم و پیاده سازی آن زمان‌بر می‌شود. مقدار بهینه این پارامتر را در گزارش خود ذکر کنید.)

نمونه از این بخش بندی در شکل زیر قابل مشاهده است:

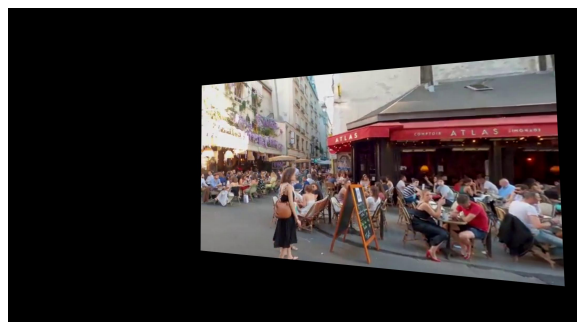


شکل ۱

## ۲ Dynamic Programming

طراح: سید سعید رضوی

یکی از کاربردهای laplacian pyramid (هرم لاپلاسیان) در ساخت یک عکس بزرگتر با استفاده از چند عکس می‌باشد. در واقع اگر چند عکس که دارای نواحی مشترک می‌باشند داشته باشیم، می‌توانیم با استفاده از هرم لاپلاسیان در ناحیه مشترک، این عکس‌ها را به هم بچسبانیم. حال دو عکس زیر را در نظر بگیرید:



شکل ۲

اگر بدون کمک از هرم لاپلاسنین بخواهیم این دو تصویر را روی هم بیندازیم داریم:



شکل ۳

همانطور که مشاهده می‌کنید تصویر حاصل کیفیت جالبی ندارد. به همین دلیل با استفاده از هرم لاپلاسنین می‌توانید ترکیب دو تصویر را کمی طبیعی‌تر کنید. احتمالاً اگر این کار را انجام دهید کیفیت تصویر نهایی نسبت به تصویر بالا بهتر شده اما باز هم مشخص است که تصویر نهایی شما ترکیب دو تصویر دیگر است. یکی از بهترین روش‌ها برای طبیعی‌تر کردن تصویر، پیدا کردن بهترین مرز جدایی این دو تصویر است. اسم این روش *minimum error boundary cut* می‌باشد که با استفاده از برنامه نویسی پویا (*dynamic programming*) بهترین مرز بین دو تصویر با ناحیه مشترک را پیدا میکند (برای مطالعه بیشتر در مورد این روش می‌توانید این لینک این لینک و اسلایدهای 29 تا 36 آن را مطالعه نمایید). در نهایت سعی کنید با پیدا کردن بهترین مرز جدایی و در آخر استفاده از هرم لاپلاسنین تصویری بزرگتر از دو تصویر اولیه بسازید. همچنین در صورت پیدا کردن مرز بهینه در ناحیه اشتراک آن را در تصویری با نام *boarder.jpg* ذخیره نمایید. استفاده از هر روش دیگری که باعث رسیدن به نتیجه بهتر شود بلامانع است. نمره این سوال تا حد خوبی به نتیجه نهایی شما بستگی دارد. تصویر نهایی را با نام *res\_q2.jpg* ذخیره نمایید.

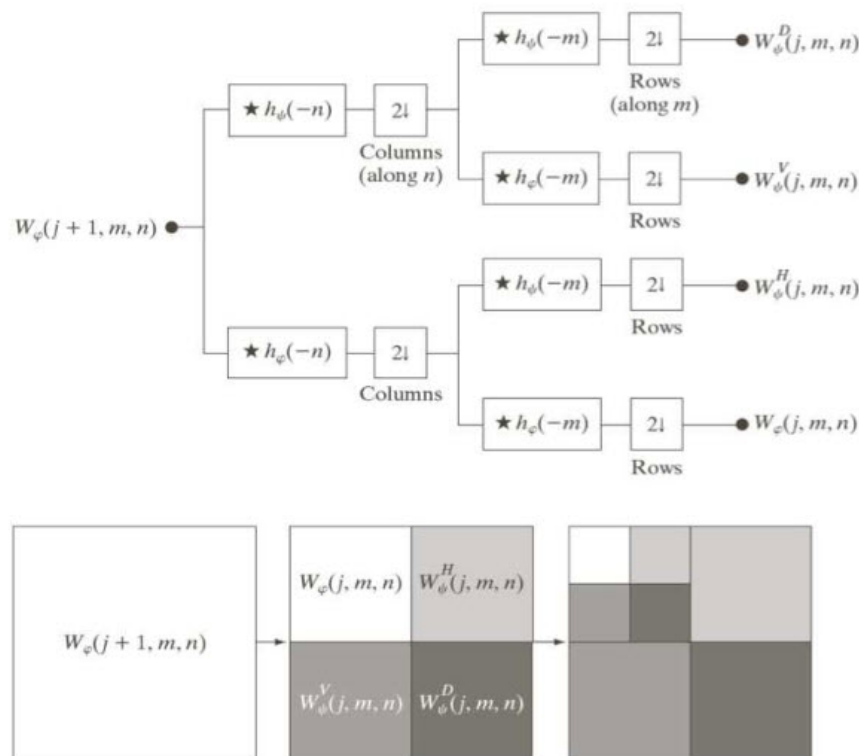
### ۳ Wavelet

طراح: ارسلان فیروزی

در این سوال قصد داریم نویز موجود در تصویر *Q3.jpg* را به وسیله تبدیل *Wavelet* کاهش دهیم.

۱. از تصویر داده شده در 2 سطح تبدیل *Wavelet* بر مبنای فیلتر *Haar* بگیرید و مطابق شکل زیر ضرایب بدست آمده را

نمایش دهید:



شکل ۴: One level of the wavelet decomposition

۲. با استفاده از روش VisoShrink به یک مقدار آستانه مناسب برسید و ضرایب را از طریق هر دو روش Soft و Hard اصلاح کنید. سپس تصاویر را بازگردانی کنید و از مناسب بودن آستانه مطمئن شوید.

۳. با استفاده از روش BayesShrink به یک مقدار آستانه مناسب برسید و ضرایب را از طریق هر دو روش Soft و Hard اصلاح کنید. سپس تصاویر را بازگردانی کنید و از مناسب بودن آستانه مطمئن شوید.

۴. با توجه به هیستوگرام ضرایب، در صورتی که سطح آستانه بزرگ در نظر گرفته شود چه تاثیری بر نتایج می‌گذارد؟ در صورتیکه سطح آستانه کوچک در نظر گرفته شود، چطور؟ نتایج با تحلیل خود را در گزارش بیاورید.

## ۴ Shannon Source Coding Theorem

طراح: امیرحسین جوادی

در این قسمت می‌خواهیم با کدینگ منبع (Source Coding) آشنا شویم. برای مدل کردن منبع اطلاعات، باید به هر سمبل از منبع یک احتمال نسبت بدهیم. در نتیجه می‌توانیم منبع اطلاعات را به این صورت نمایش دهیم.

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_M \\ p_1 & p_2 & \dots & p_M \end{pmatrix}$$

به عنوان مثال، اگر شما تصاویر سیاه و سفید تولید می‌کنید هر سمبل خروجی این منبع، عددی بین 0 تا 255 است (در نتیجه  $M = 256$ ). مسأله‌ی کدینگ منبع آنست که می‌خواهیم یک نگاشت مانند  $C$  از الفبای خروجی منبع به الفبای دوتایی  $\mathcal{D} = \{0, 1\}$  پیدا کنیم به گونه‌ای که طول متوسط هر کلمه‌کد کمینه شود. به تعبیر دیگر اگر طول کلمه‌کد متناظر با سمبل  $x \in \mathcal{X}$  را با  $l(x)$  نشان دهیم، هدف آن است که  $\mathbb{E}[l(X)]$  کمینه شود.

منبع اطلاعات زیر را در نظر بگیرید:

$$X = \begin{pmatrix} I_x = 0 & I_x = 50 & I_x = 100 & I_x = 150 & I_x = 200 & I_x = 250 & \text{otherwise} \\ \frac{1}{2} & \frac{1}{32} & \frac{1}{8} & \frac{1}{16} & \frac{1}{32} & \frac{1}{4} & 0 \end{pmatrix}$$

۱. کلمه کدها را برای منبع  $X$  بسازید. کدینگ شما باید قابل بازیابی باشد و همچنین بهینه باشد.
۲. تابعی به نام InformationSource بسازید که تصویری (آرایه دو بعدی) به اندازه  $28 \times 28$  به صورت iid تولید کند که توزیع احتمالی مانند توزیع احتمال منبع  $X$  باشد.
۳. تابعی با عنوان SourceEncoder بنویسید که در ورودی تصویر ساخته شده در قسمت قبل را در ورودی دریافت کند و در خروجی رشته از صفر و یک‌ها بدهد که کدشده‌ی تصویر ورودی است.
۴. تابعی با عنوان SourceDecoder بنویسید که در ورودی، دنباله‌ای از صفر و یک‌ها دریافت کند و در خروجی، تصویر متناظر را بدهد.
۵. درستی توابع SourceEncoder و SourceDecoder را بررسی کنید. 10 تصویر تصادفی تولید کنید و مراحل کدگذاری و کدگشایی را روی این تصاویر اعمال کنید. آیا تصاویر ورودی و خروجی برابر شد؟
۶.  $n = 1000$  تصویر تصادفی تولید کنید. برای هر تصویر طول رشته کدگذاری شده را بیابید. برداری به طول  $n$  به نام  $L\_avg$  بسازید و خانه‌ی  $i$  ام آن را میانگین طول رشته کد  $i$  عکس اول قرار دهید. نمودار تغییرات این بردار را رسم کنید. انتظار دارید میانگین خواسته شده به چه عددی میل کند؟

## ۵ Golomb Coding

طراح: ارسلان فیروزی

در این سوال از شما انتظار می‌رود که روش کدینگ Golomb را برای یک تصویر Grayscale دلخواه پیاده سازی کنید. از طریق تابع انکودینگ ابتدا تمام مقادیر پیکسل‌ها را با اعداد رندوم  $m$  انکود کنید. سپس از طریق تابع دیکود و همان مقادیر رندوم  $m$ ، مقادیر کد شده را دیکود کنید. با توجه به Lostless بودن فشرده سازی، انتظار داریم تصویر بدست آمده مطابق تصویر اولیه باشد. از طریق کامنت خطوط کد را توضیح دهید.

## ۶ تئوری

مسائل روبرو از فصل هشتم کتاب درسی (ویرایش سوم) را حل کنید: 1، 8، 10 و 18