

## Image Processing – Phase 1

کورس حسن‌زاده – متین مژگانی – شروین خادمی کلانتری

### انتخاب مسیر

مسیر پروژه به صورت ماژولار انتخاب شده؛ یعنی به جای تمرکز روی یک مسئله‌ی واحد، چند محور اصلی Image Processing از سطح پایه تا پیشرفته پوشش داده شده است. این مسیر شامل:

1. **Spatial & Frequency Filtering**
2. **Denoising**
3. **Edge Detection**
4. **Segmentation**
5. **Adaptive Thresholding**
6. **Histogram Analysis + HOG**
7. **Deep Learning Integration (Autoencoder on MNIST)**

هدف از این مسیر، ساخت یک دید عملی end-to-end از مراحل رایج پردازش تصویر و مقایسه‌ی خروجی‌ها در سناریوهای مختلف است.

### تعریف مسئله

مسئله‌ی اصلی پروژه به شکل یک «مسئله‌ی واحد» تعریف نشده، بلکه شامل چند مسئله‌ی استاندارد در پردازش تصویر است:

- چگونه با filtering حوزه مکان و فرکانس کیفیت تصویر را بهبود دهیم یا ویژگی استخراج کنیم؟
- چگونه (Gaussian / Salt & Pepper) noise را اضافه و سپس با فیلترهای مناسب حذف کنیم؟
- چگونه با الگوریتم‌های مختلف edge detection مرزها و ساختارها را آشکار کنیم؟
- چگونه با روش‌هایی مثل K-Means تصویر را segmentation کنیم؟
- چگونه thresholding (به خصوص adaptive) را برای تصاویر با روشنایی غیر یکنواخت اجرا کنیم؟
- چگونه با Histogram / HOG features تحلیل و feature extraction انجام دهیم؟
- چگونه در نهایت با یک Denoising Autoencoder نویز را با مدل یادگیری ماشین کاهش دهیم؟

## معرفی دیتاست

در این پروژه دیتاست ثابت نداریم و بسته به هر مسئله، داده‌ی مناسب همان بخش انتخاب شده است:

- برای Filtering و Thresholding از تصویر استاندارد Lenna استفاده شده.
- برای Denoising مجموعه‌ای از تصاویر (ترکیب grayscale و RGB) آماده شده.
- برای Edge Detection تصاویر کلاسیک مثل Cameraman و Goldhill استفاده شده.
- برای Histogram Analysis از تصاویر رنگی استفاده شده.
- برای Segmentation چند تصویر رنگی با ابعاد متفاوت استفاده شده.
- برای بخش Deep Learning از دیتاست استاندارد MNIST استفاده می‌شود.

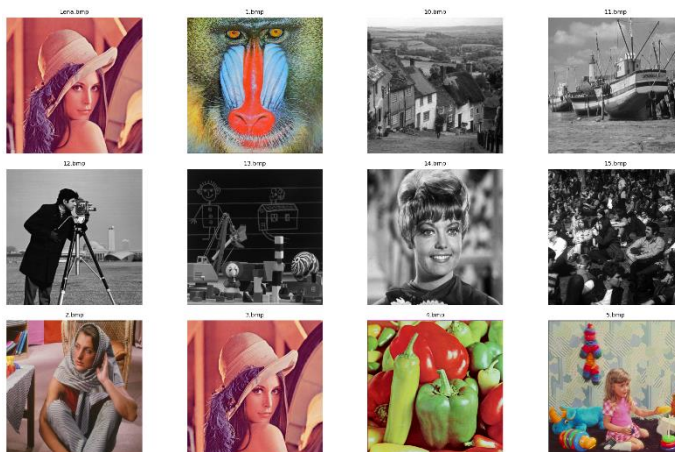
این انتخاب باعث می‌شود هر بخش روی داده‌ای اجرا شود که واقعاً برای آن task رایج و استاندارد است.

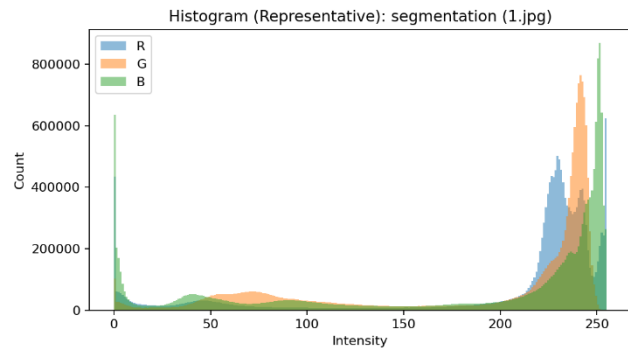
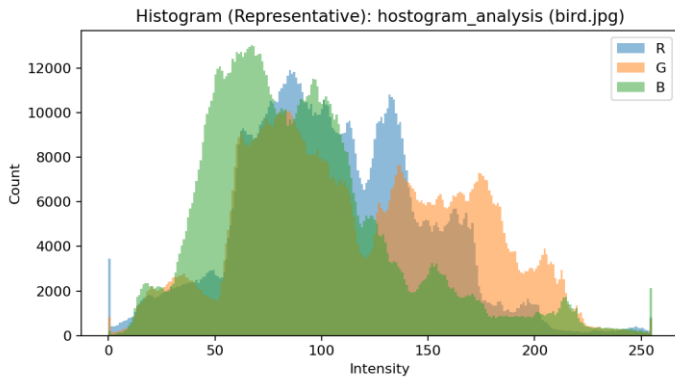
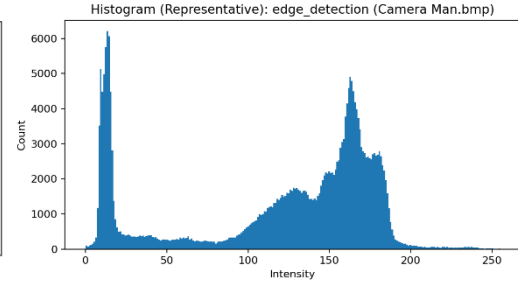
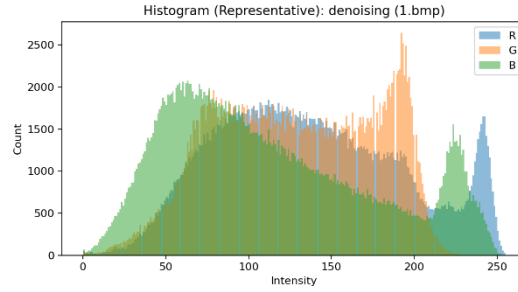
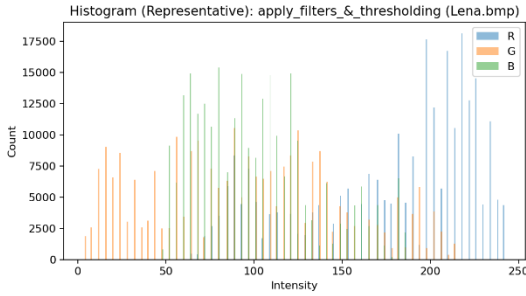
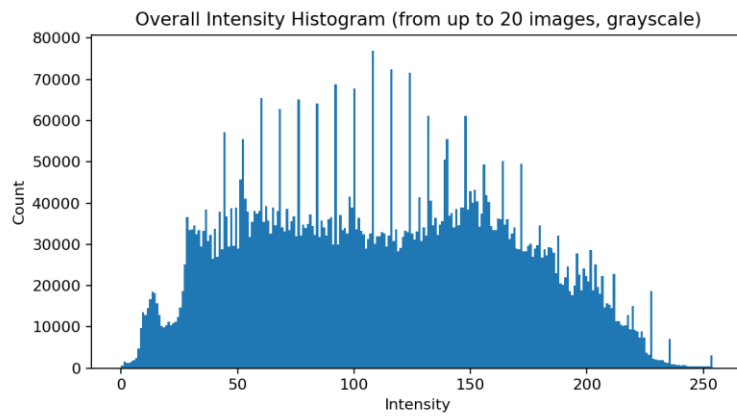
## تحلیل داده‌ها (EDA)

برای بررسی کیفیت و ساختار داده‌ها، یک اسکریپت **EDA** نوشته شده که تمام تصاویر مسیر data/ را می‌خواند و موارد زیر را گزارش می‌دهد:

- تعداد کل تصاویر
- تعداد channel ها (مثلاً 1-channel grayscale و 3-channel RGB)
- Dtype همه (uint8)
- آمار ابعاد (height/width)
- آمار intensity شامل min/max/mean/std
- تولید خروجی‌های تصویری:
  - Sample Grid از تصاویر هر پوشه
  - Histogram نماینده از هر پوشه
  - یک Histogram کلی از چند تصویر به صورت grayscale

Random Samples (Overall)





## خلاصه خروجی EDA

- Total images: **24**
- Unique channels: **[1, 3]**
- dtype: **uint8**
- Height: min=512, max=4834, mean≈943.9
- Width: min=512, max=3359, mean≈855.0
- Pixel mean: min≈57.95, max≈189.12, mean≈118.99
- Pixel std: min≈40.41, max≈80.28, mean≈55.21

## Per-folder Summary

- apply\_filters\_&\_thresholding: 1 image (512×512×3)

- denoising: 15 images ( $512 \times 512 \times 1/3$ )
- edge\_detection: 2 images ( $512 \times 512 \times 1$ )
- histogram\_analysis: 3 images (ابعاد متفاوت, RGB)
- segmentation: 3 images (RGB, بزرگ با ابعاد خیلی بزرگ)

## پیش پردازش‌ها (Preprocessing)

در این پروژه، بخش بزرگی از pipeline خود عملیات‌ها نقش Preprocessing دارند (مثل filtering, denoising, thresholding). بنابراین preprocessing جداگانه‌ی سنگین نیاز نبوده. تنها موارد پایه‌ای که در اکثر بخش‌ها implicitly انجام می‌شود:

- خواندن تصویر و استانداردسازی (format (grayscale/RGB)
- در برخی بخش‌ها تبدیل به grayscale برای histogram یا edge detection
- نرمال‌سازی/scale کردن داده‌ها در بخش Autoencoder به  $[0,1]$

## مدل پایه (Baseline)

برای فاز اول، مدل پایه در هر تسک با روش‌های کلاسیک تعریف می‌شود:

- Filtering Baseline: Average / Gaussian / Median به عنوان Sharpening/Sobel + baseline
- smoothing برای enhancement/gradient
- Denoising Baseline: Median filter به عنوان baseline برای Salt & Pepper، و مقایسه با Bilateral filter
- Edge Detection Baseline: پیاده‌سازی و مقایسه Prewitt/Kirsch/Marr-Hildreth/Canny
- Segmentation Baseline: K-Means clustering روی فضای رنگ RGB
- Thresholding Baseline: Global thresholding در مقابل Adaptive Thresholding
- Histogram Analysis Baseline: تحلیل histogram و استخراج HOG features
- Deep Learning Baseline: یک Denoising Autoencoder روی MNIST با معماری زیر:

یک مدل Convolutional Autoencoder با:

- Encoder: Conv2D(16, stride=2) سپس Conv2D(8, stride=2)
- Decoder: Conv2DTranspose و در انتها Conv2D(1, sigmoid)
- هدف: بازسازی تصویر ورودی و denoising

# پلن آزمایش‌ها (Experiment Plan)

پلن آزمایش‌ها به تفکیک ماژول‌ها:

## 1) Spatial & Frequency Filtering (Lenna)

- اجرای Spatial filters: Average, Gaussian, Median, Sharpening, Sobel
- اجرای Frequency filters: Low Pass, High Pass

## 2) Denoising

- اضافه کردن Gaussian noise و Salt & Pepper noise
- حذف نویز با:
  - Median filter
  - Bilateral filter
  - (artifact, edge preservation)

## 3) Edge Detection

- اجرای Prewitt, Kirsch, Marr-Hildreth, Canny
- مقایسه‌ی حساسیت به noise، حفظ لبه‌ها، false edges

## 4) Segmentation (K-Means)

- اجرای K-Means با k های مختلف
- تحلیل تاثیر k روی:
  - over-segmentation / under-segmentation
  - وضوح مرزبندی

## 6) Histogram Analysis + HOG

- رسم histogram و تفسیر توزیع intensity
- استخراج HOG features

## 7) Autoencoder on MNIST

- آموزش مدل روی MNIST clean/noisy
- تست denoising روی نمونه‌های noisy