

Image Processing – Phase 1

کورش حسن‌زاده – متین مژگانی – شروین خادمی کلانتری

انتخاب مسیر

مسیر پروژه به صورت مازوچار 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:

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

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

تعريف 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 با (Gaussian / Salt & Pepper) noise 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 segmentation تصویر را کنواخت اجرا کنیم؟
- چگونه adaptive thresholding (به خصوص) 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:

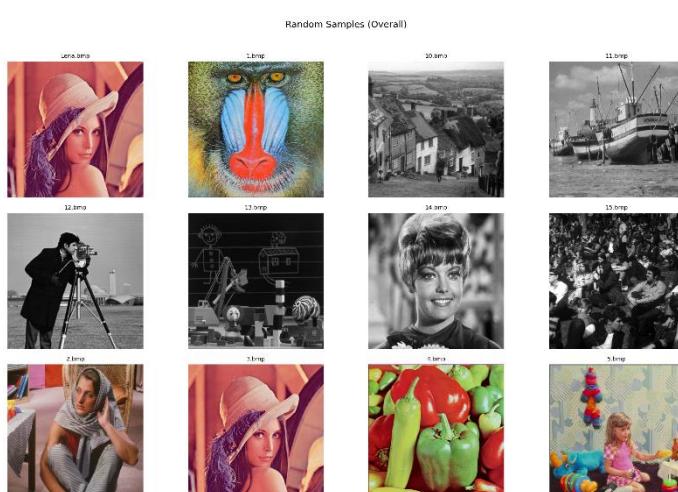
- برای Filtering و Thresholding از تصویر استاندارد Lenna استفاده شده.
- برای Denoising مجموعه‌ی از تصاویر (ترکیب RGB و grayscale) آمده شده.
- برای Edge Detection تصاویر کلاسیک مثل Cameraman و Goldhill استفاده شده.
- برای Histogram Analysis از تصاویر رنگی استفاده شده.
- برای Segmentation چند تصویر رنگی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

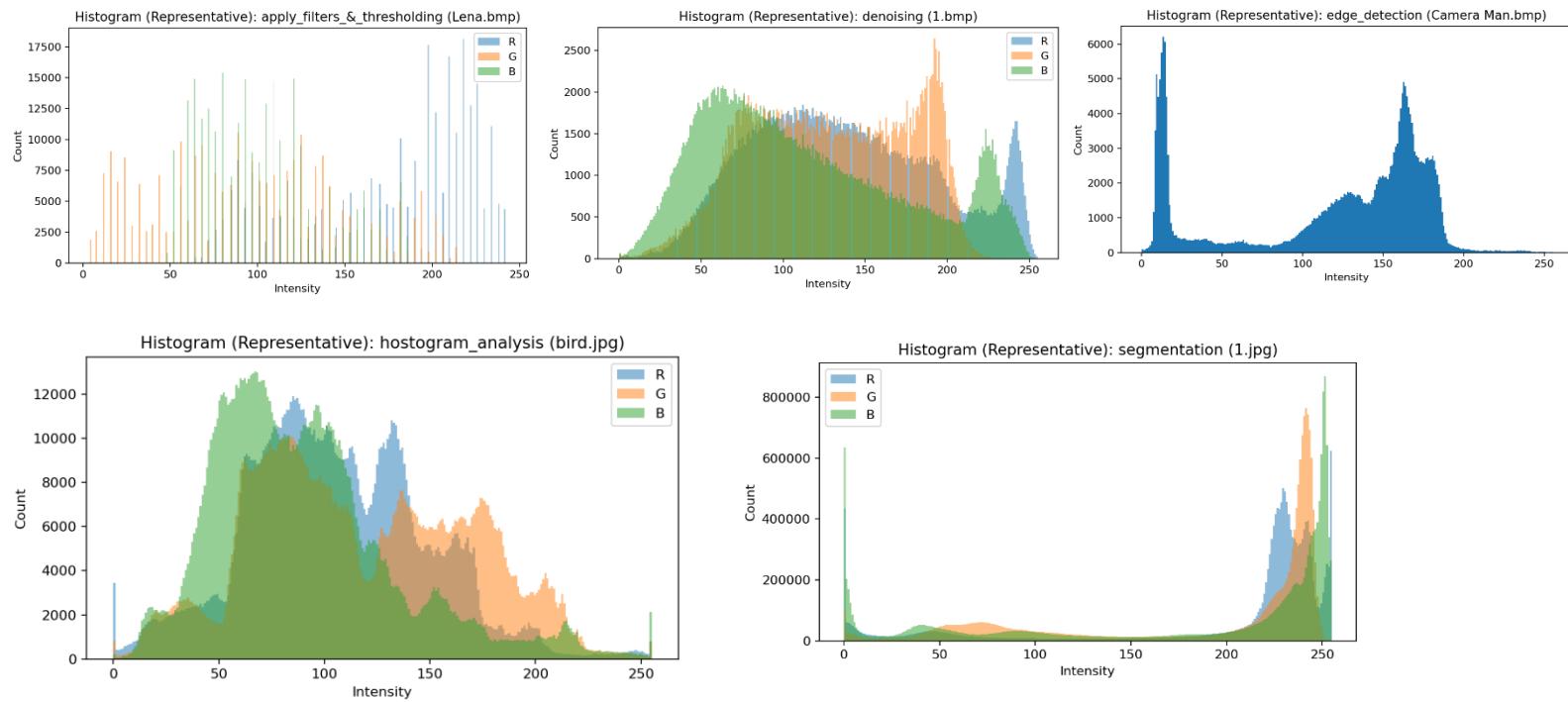
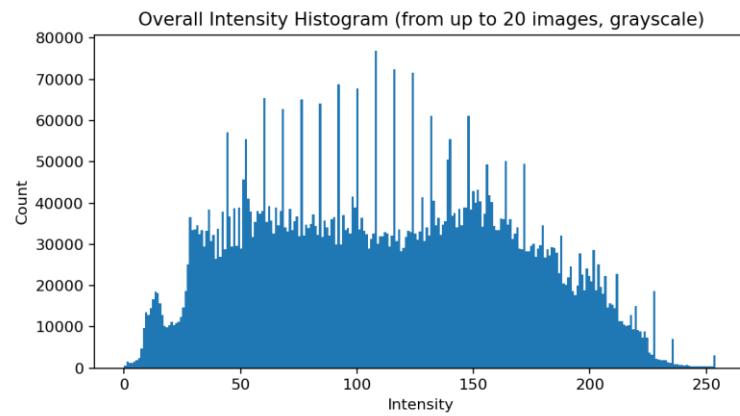
این انتخاب باعث می‌شود هر بخش روی داده‌ای اجرا شود که واقعاً برای آن task رایج و استاندارد است.

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 RGB-3 و channel grayscale-1 (مثالاً channel grayscale-1 همها)
- Dtype همه (uint8)
- آمار ابعاد (height/width)
- آمار intensity شامل min/max/mean/std
- تولید خروجی‌های تصویری:
 - از Sample Grid
 - نماینده از هر پوشه
- يك Histogram کلی از چند تصویر به صورت grayscale
- يك Histogram از 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 ($512 \times 512 \times 3$)

- denoising: 15 images ($512 \times 512 \times 1/3$)
- edge_detection: 2 images ($512 \times 512 \times 1$)
- histogram_analysis: 3 images (RGB, ابعاد متفاوت)
- segmentation: 3 images (RGB, خیلی بزرگ)

پیش‌پردازش‌ها (Preprocessing)

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

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

مدل پایه (Baseline)

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

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

یک مدل Convolutional Autoencoder با:

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

پلن آزمایش‌ها (Experiment Plan)

پلن آزمایش‌ها به تفکیک مازول‌ها:

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
- مقایسه 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 روی نمونه‌های noisy denoising