

INTRODUCTION TO COMPUTER VISION

PROJECT DOCUMENTATION



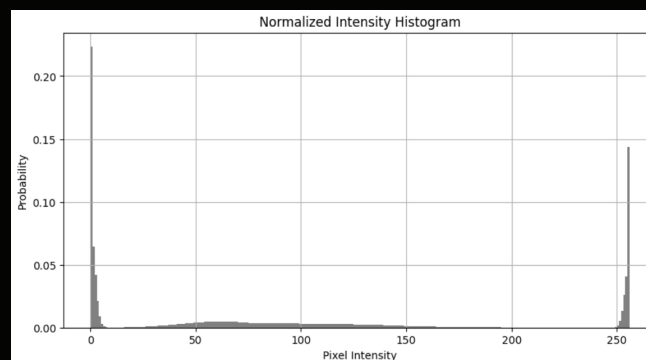
Ferdowsi University of Mashhad
Department of Computer Engineering

SPRING 2025

نام و نام خانوادگی	شماره دانشجویی
امیرحسین افشار	۴۰۱۲۲۶۲۱۹۶

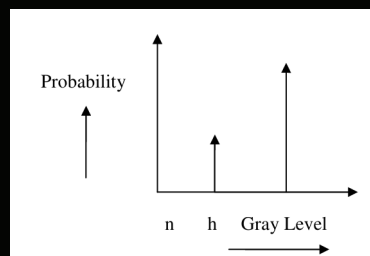
۱) سوال اول: معمای داوینچی

در ابتدا برای حل پازل، عملیات denoising را برای تصویر *processed_img_part_1.jpg* انجام شد. بدین منظور، از هیستوگرام تصویر بهره گرفته شده است ^[۱]؛ هیستوگرام تصویر با سطوح خاکستری، بر اساس نرمال کردن میزان پیکسل های با شدت بین ۰ تا ۲۵۵ ساخته می شود تا شکل *function probability distribution* بدست آید. پلات زیر، بیانگر هیستوگرام برای تصویر نویزی شماره ۱ است:



شکل ۱: پلات هیستوگرام تصویر نویزی شماره ۱

این شکل، بیان می کند که تعداد پیکسل ها با مقدار نزدیک به صفر و مقدار نزدیک به ۲۵۵ بسیار زیاد است. همچنین هیستوگرام احتمال قابل انتظار برای تصاویر نویزی با نوع نویز نمک فلفل به شکل زیر است:



شکل ۲: هیستوگرام قابل انتظار برای تصاویر نویزی نمک و فلفل

^[۱] Asoke Nath: Image Denoising Algorithms: A Comparative Study of Different Filtration Approaches Used in Image Restoration

با مقایسه شکل ۱ و ۲ می توان نتیجه گرفت که نویز تصویر شماره ۱ از نوع نمک و فلفل می باشد. بنابراین، برای denoise کردن تصویر، بهترین گزینه، استفاده از فیلتر median می باشد. شایان ذکر است که در عملیات denoise کردن تصویر برای اطمینان بیشتر از نوع نویز، از انواع فیلترها استفاده شد که به شرح زیر هستند:

Category	Filter Name	Function
Smoothing Filters	Box Filter	denoise_box_filter
	Gaussian Filter	denoise_gaussian_filter
Statistical Filters	Median Filter	denoise_median_filter
	Max Filter	denoise_max_filter
	Min Filter	denoise_min_filter
Advanced Filters	Bilateral Filter	denoise_bilateral_filter
	Non-Local Means	denoise_nl_means
	Wavelet Filter	denoise_wavelet_filter
	Total Variation Filter	denoise_total_variation

همانطور که از شکل ۱ انتظار می رفت، بهترین عملکرد خروجی با استفاده از فیلتر میانه یا همان median بدست می آید. در نهایت، برای نهایی کردن بهترین خروجی با استفاده از فیلتر میانه، دو روش پیگیری شد:

۱. استفاده از چند مرحله فیلتر median با ابعاد یکسان

۲. استفاده از یک فیلتر median با کرنل بزرگتر

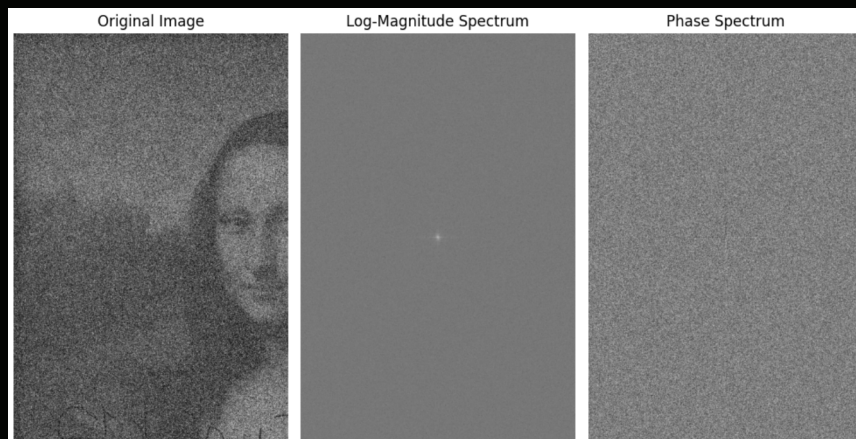
روش اول، به طور کلی برای حفظ جزئیات تصویر مناسب تر است؛ زیرا هرگونه کرنل بزرگ با ریسک از دست دادن جزئیات همراه است. از طرفی، ممکن است که همگرا شدن تصویر پس از اعمال چند بار فیلتر میانه، به کندی پیش رود و حتی برخی نویزها در تصویر باقی بمانند که این مشکل، در روش دوم وجود ندارد.

در نهایت، با توجه به مسئله که تنها نیاز است یک متن از تصویر استخراج شود (و حفظ سایر جزئیات دارای اهمیت کمتری است)، از روش دوم استفاده شد.



شکل ۳: شکل نهایی تصویر denoise شده

همچنین شایان ذکر است که برای درک بهتر نویز، از حوزه ی فرکانس نیز کمک گرفته شد^[1] که بتوان نوع نویز را حدس زد و در نهایت با استفاده از فیلتر های notch و یا band-pass و یا band-reject از نویز تصویر کاست. شکل سوم بیانگر این موضوع است.



شکل ۴: تصویر شماره ۱ در حوزه فرکانس

با توجه به magnitude تصویر در حوزه فرکانس، نمی توان نویز خاصی را قائل شد که نهایتاً، همان فیلتر میانه که از هیستوگرام تصویر درک شده بود، استفاده شد.

^[1] Digital Image Processing By Gonzalez 2nd Edition 2002, chapter 4.2

در ابتدا، به کمک کتابخانه *cv2* تابع زیر را برای دریافت فیچرها تکمیل کردیم.

```
1 def extract_features(image_path)
2   # input is an image and the ouput is an array of features.
3
```

Listing 1: extract feature function



شکل ۶: new caption



شکل ۵: caption another