



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM

FACULDADE DE TECNOLOGIA

ENGENHARIA DA COMPUTAÇÃO

PIPELINE DE PROCESSAMENTO DE IMAGENS COM
BAIXA LUMINOSIDADE E MÉTODOS CEGOS DE
AVALIAÇÃO

Lucas Ribeiro Almeida

Manaus - AM

2025

Lucas Ribeiro Almeida

PIPELINE DE PROCESSAMENTO DE IMAGENS COM
BAIXA LUMINOSIDADE E MÉTODOS CEGOS DE
AVALIAÇÃO

Monografia apresentada à Coordenação do Curso de Engenharia da Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Engenheiro da Computação.

Orientador(a)

Prof. Dr. Francisco de Assis Pereira Januário

Universidade Federal do Amazonas - UFAM

FACULDADE DE TECNOLOGIA

Manaus - AM

2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

A447p Almeida, Lucas Ribeiro
Pipeline de processamento de imagens com baixa luminosidade e métodos cegos de avaliação / Lucas Ribeiro Almeida. - 2025.
91 f. ; 31 cm.

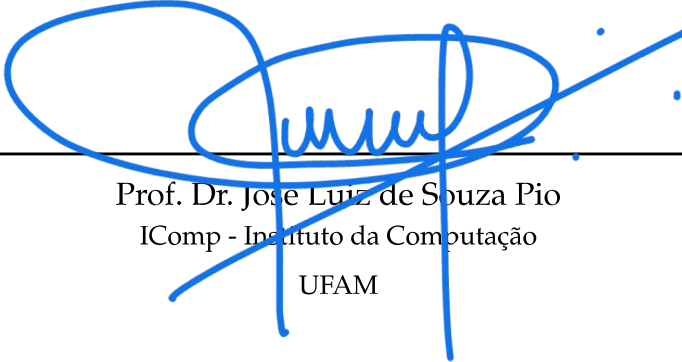
Orientador(a): Francisco de Assis Pereira Januário.
Trabalho de Conclusão de Curso (graduação) - Universidade Federal do Amazonas, Faculdade de Tecnologia, Curso de Engenharia da Computação, Manaus, 2025.

1. processamento de imagem. 2. imagem RAW. 3. baixa luminosidade. 4. avaliação de qualidade. 5. BRISQUE. I. Januário, Francisco de Assis Pereira. II. Universidade Federal do Amazonas. Faculdade de Tecnologia. Curso de Engenharia da Computação. III. Título

Monografia de Graduação sob o título *Pipeline de processamento de imagens com baixa luminosidade e métodos cegos de avaliação* apresentada por Lucas Ribeiro Almeida e aceita pela Faculdade de Tecnologia da Universidade Federal do Amazonas, sendo aprovada por todos os membros da banca examinadora abaixo especificada:

Francisco de Assis Pereira Januário

Prof. Dr. Francisco de Assis Pereira Januário
Orientador(a)
DETEC - Departamento de Eletrônica e Computação
UFAM



Prof. Dr. Jose Luiz de Souza Pio
IComp - Instituto da Computação
UFAM

Frederico da Silva Pinagé

Prof. Dr. Frederico Da Silva Pinagé
DETEC - Departamento de Eletrônica e Computação
UFAM

Manaus - AM, 17 de Dezembro de 2025.

AGRADECIMENTOS

Quero, antes de tudo, agradecer a Deus, que sempre foi a minha rocha e o meu alicerce. Foi Ele quem me sustentou nos momentos de incerteza, quem me deu forças quando o caminho parecia difícil e quem renovou a minha esperança para que eu jamais desistisse dos meus objetivos.

Agradeço profundamente à minha mãe, Elci Ribeiro Almeida, que deixou sozinha sua terra natal para construir uma nova vida em Manaus. Ela me criou com coragem, amor e sacrifício, abrindo mão de muitos sonhos para que eu pudesse ter acesso a oportunidades que ela nunca teve. Crescer ao seu lado, testemunhando sua determinação e resiliência, moldou quem eu sou e me inspirou a sempre buscar o meu melhor. Nada do que conquistei seria possível sem o seu exemplo e sem o seu amor incondicional.

Sou igualmente grato à minha esposa, Iara Caroline, que com sabedoria, paciência e carinho foi meu porto seguro ao longo desses anos. Seu apoio constante, suas palavras de incentivo e sua presença firme nos momentos mais desafiadores foram essenciais para que eu continuasse avançando. Ela esteve ao meu lado quando eu duvidei de mim mesmo e nunca permitiu que eu desistisse.

Assim como também expressei minha gratidão a todos os meus amigos e familiares que, de alguma forma, contribuíram para esta caminhada. Cada palavra de incentivo, cada gesto de apoio e cada demonstração de carinho tiveram um papel essencial ao longo do percurso.

A todos vocês, meu sincero muito obrigado.

*Para que todos vejam, e saibam, e considerem, e juntamente entendam que a mão do SENHOR
fez isso, e o Santo de Israel o criou.*

Isaías 41:20

PIPELINE DE PROCESSAMENTO DE IMAGENS COM BAIXA LUMINOSIDADE E MÉTODOS CEGOS DE AVALIAÇÃO

Autor: Lucas Ribeiro Almeida

Orientador: Prof. Dr. Francisco de Assis Pereira Januário

Resumo

Este trabalho propõe o desenvolvimento de uma pipeline de processamento de imagens voltada à melhoria de fotografias capturadas em ambientes de baixa luminosidade. A proposta fundamenta-se no uso de imagens RAW, que preservam informações completas do sensor da câmera, permitindo maior flexibilidade no pós-processamento. A pipeline desenvolvida utiliza um conjunto de etapas clássicas de Image Signal Processing (ISP), incluindo a leitura e linearização da imagem, demosaicing, correção de cor, balanço de branco, ajuste de exposição, correção de gama, remoção de névoa (dehaze), realce de contraste local (CLAHE), redução de ruído e ajuste de saturação. A avaliação da qualidade das imagens processadas foi realizada por meio de métricas objetivas sem referência, como BRISQUE e NIQE, além de observações visuais qualitativas. Os resultados indicam que a pipeline proposta é capaz de recuperar regiões escurecidas, melhorar a nitidez e tornar visíveis detalhes previamente ocultos. A comparação entre os cenários com e sem o uso da filtragem de ruído também forneceu insights relevantes sobre a influência desse bloco na percepção de qualidade da imagem. O sistema desenvolvido é modular, interpretável e aplicável em diferentes contextos de captura noturna.

Palavras-chave: processamento de imagem, imagem RAW, baixa luminosidade, avaliação de qualidade, BRISQUE, NIQE.

PIPELINE DE PROCESSAMENTO DE IMAGENS COM BAIXA LUMINOSIDADE E MÉTODOS CEGOS DE AVALIAÇÃO

Autor: Lucas Ribeiro Almeida

Orientador: Prof. Dr. Francisco de Assis Pereira Januário

Abstract

This work presents the development of an image processing pipeline aimed at enhancing photographs captured in low-light environments. The proposal is based on the use of RAW images, which preserve the complete information from the camera sensor, allowing for greater flexibility in post-processing. The designed pipeline is composed of a set of classical Image Signal Processing (ISP) stages, including RAW image reading and linearization, demosaicing, color correction, white balance, exposure adjustment, gamma correction, haze removal, local contrast enhancement (CLAHE), noise reduction, and saturation adjustment. The quality of the processed images was evaluated using no-reference objective metrics, such as BRISQUE and NIQE, as well as qualitative visual inspection. The results show that the proposed pipeline is effective in recovering dark regions, improving sharpness, and revealing previously hidden details. Comparisons between scenarios with and without noise filtering provided additional insight into the impact of this step on perceived image quality. The system is modular, interpretable, and suitable for various low-light capture contexts.

Keywords: image processing, RAW images, low-light, quality assessment, BRISQUE, NIQE.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de blocos do sistema de processamento.	29
Figura 2 – Estrutura de Diretórios de projeto.	44
Figura 3 – Imagens processadas.	47
Figura 4 – Imagens processado.	48
Figura 5 – Imagens processadas.	50
Figura 6 – Imagens processadas.	52
Figura 7 – Remoção de névoa.	53
Figura 8 – CLAHE.	55
Figura 9 – Saturação.	56
Figura 10 – BRISQUE com bloco de ruído.	62
Figura 11 – NIQE com bloco de ruído.	63
Figura 12 – BRISQUE sem bloco de ruído.	63
Figura 13 – NIQE sem bloco de ruído.	64
Figura 14 – Imagens processadas.	65
Figura 15 – Imagens processadas.	66
Figura 16 – BRISQUE - Com e sem denoise.	66
Figura 17 – NIQE- Com e sem denoise.	67
Figura 18 – Quadra de basquete com detalhe ampliado.	69
Figura 19 – Cabine telefônica com detalhe ampliado.	70
Figura 20 – Quadra de basquete e Lago	78
Figura 21 – Rua e Piscina.	79
Figura 22 – Vaso e Cadeira.	79

Figura 23 – Placa e banco.	79
Figura 24 – Carro e Telefone.	80
Figura 25 – Quadra de basquete e Lago	90
Figura 26 – Placa e Piscina	90
Figura 27 – Rua e Vaso	91
Figura 28 – Cadeiras e Telefone	91
Figura 29 – Carro e Banco	91

LISTA DE TABELAS

Tabela 1 – Vantagens do formato RAW e sua base técnica.	23
Tabela 2 – Comparação de trabalhos e técnicas aplicadas em avaliação de imagens de baixa luminosidade	27
Tabela 3 – Bibliotecas utilizadas e suas finalidades.	46
Tabela 4 – Resultados BRISQUE e NIQE com e sem bloco de ruído.	62
Tabela 5 – Comparação BRISQUE e NIQE.	65
Tabela 6 – Comparação BRISQUE e NIQE - Variação Percentual	68

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Contextualização	14
1.2	Justificativa	15
1.3	Objetivo Geral	16
1.4	Objetivos Específicos	16
1.5	Organização do Trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Imagem Digital	18
2.2	Processamento de imagem	19
2.3	Aquisição de imagem e ISP	19
2.4	Princípios Fundamentais da Aquisição	20
2.5	Modalidades de Aquisição	20
2.6	Modalidades de Aquisição	21
2.7	Formato RAW: Aquisição, Relevância e Aplicações	22
2.8	Avaliação da qualidade de Imagem	23
3	TRABALHOS RELACIONADOS	25
4	METODOLOGIA	29
4.1	Pipeline de Processamento	30
4.1.1	Leitura da Imagem bruta - RAW	30
4.1.2	Correção de Nível de Preto	31
4.1.3	Demosaico	32
4.1.4	Balanço de Branco	34
4.1.5	Correção de Exposição	35

4.1.6	Correção de Cor – Matriz de Correção Cromática (CCM)	35
4.1.7	Redução de Ruído	36
4.1.8	Correção Gama	37
4.1.9	Remoção de Névoa	38
4.1.10	Equalização de Contraste Local – CLAHE	40
4.1.11	Ajuste de saturação	42
5	DESENVOLVIMENTO	43
5.1	Estrutura da Pipeline Implementada	43
5.1.1	Organização do código	44
5.2	Ambientes e Ferramentas Utilizadas	44
5.2.1	Ambiente de Desenvolvimento	45
5.3	Resultados Intermediários do Processamento	46
5.3.1	Leitura da Imagem RAW e Correção de Nível de Preto	46
5.3.2	Demosaico e Balanço de branco	48
5.3.3	Correção de exposição e Correção de Cor	49
5.3.4	Redução de ruído e Correção de Gama	50
5.3.5	Remoção de Névoa, CLAHE e Saturação	52
5.4	Considerações Técnicas	56
5.4.1	Formato RAW e limitações dos metadados	56
5.4.2	Ajustes manuais nos ganhos de exposição e balanço de branco	57
5.4.3	Correção Gama e percepção visual	57
5.4.4	CLAHE	57
5.4.5	Saturação e estética da imagem	57
5.4.6	Modularização e salvamento por etapas	58
5.5	Preparação para Avaliação da Qualidade	58
5.5.1	Métricas de Qualidade Utilizadas	58
5.5.2	Implementação da Avaliação	59
6	RESULTADOS E DISCUSSÕES	60
6.1	Métricas BRISQUE e NIQE	61

6.1.1	Pontuações	61
6.2	Avaliação Visual da Recuperação de Cena	69
6.2.1	Recuperação de detalhes em regiões escuras	69
7	CONSIDERAÇÕES FINAIS	71
7.1	Limitações	72
7.2	Trabalhos Futuros	73
	Referências	75
APÊNDICE A	PRIMEIRO APÊNDICE	78
APÊNDICE A	SEGUNDO APÊNDICE	81
ANEXO A	PRIMEIRO ANEXO.	90

1

INTRODUÇÃO

1.1 Contextualização

A aquisição de imagens em condições de baixa luminosidade representa um desafio significativo tanto para sistemas fotográficos convencionais quanto para aplicações avançadas de visão computacional. Nesses cenários, as imagens resultantes frequentemente apresentam limitações críticas, incluindo baixo brilho, contraste reduzido, distorções cromáticas e níveis elevados de ruído, comprometendo tanto sua qualidade visual quanto sua utilidade para análise algorítmica (RASHEED; SHI; KHAN, 2023).

Convencionalmente, a pipeline de processamento de imagens digitais compreende etapas sequenciais como correção de nível preto, tratamento de pixels defeituosos, *demosaicing*, balanço de branco, ajuste cromático, redução de ruído, mapeamento de tons, aplicação de curva *gamma* e realce de nitidez (ZAMIR et al., 2019). Embora eficiente sob condições normais de iluminação, este fluxo demonstra limitações pronunciadas quando aplicado a ambientes de luminosidade extremamente reduzida, onde a baixa relação sinal-ruído (*SNR*) degrada severamente a qualidade da saída (XU et al., 2022)).

Em resposta a esses desafios, pesquisas recentes têm explorado abordagens inovadoras (ZAMIR et al., 2019) demonstraram que redes neurais profundas podem aprender mapeamentos diretos entre dados *RAW* capturados em baixa luz e imagens *sRGB* otimizadas perceptualmente, superando pipelines tradicionais na reconstrução de detalhes, contraste e fidelidade cromática. Paralelamente, (GUO et al., 2021) introduziram o conceito de *Zero-Reference Deep Curve Estimation (Zero-DCE)*, que formula o realce luminoso como um problema de estimativa de curvas específicas por imagem, elimi-

nando a necessidade de dados pareados para treinamento e preservando características naturais mesmo sob iluminação não uniforme.

Avanços significativos também emergiram na integração de informações locais e não-locais: (XU et al., 2022) propuseram um pipeline guiado por mapas de *SNR* que aplica operações de longo alcance (*transformers*) em regiões críticas de ruído e convoluções locais em áreas de melhor visibilidade, otimizando tanto métricas objetivas quanto avaliação subjetiva.

Contudo, a avaliação da qualidade perceptual permanece um desafio. Métricas tradicionais como *PSNR* e *SSIM* mostram baixa correlação com a percepção humana em cenários noturnos (RASHEED; SHI; KHAN, 2023), destacando a necessidade de abordagens especializadas como o BRISQUE.

1.2 Justificativa

A crescente demanda por sistemas de captura e análise de imagens robustos têm evidenciado a importância de soluções que atuem eficientemente em cenários de baixa luminosidade. Imagens capturadas à noite ou em ambientes mal iluminados frequentemente apresentam baixos níveis de detalhe, ruído acentuado, desequilíbrio de cores e exposição comprometida, o que dificulta tanto a interpretação visual quanto a aplicação de algoritmos posteriores de análise de imagem.

Nesse contexto, o uso do formato *RAW* se mostra vantajoso por manter os dados originais capturados pelo sensor, sem compressão ou pós-processamento embutido, permitindo que o desenvolvedor aplique algoritmos personalizados de aprimoramento com maior controle sobre os resultados. Diferentemente de abordagens baseadas em aprendizado de máquina, o presente trabalho propõe uma solução fundamentada em técnicas clássicas e bem estabelecidas de processamento de imagens, organizadas de forma modular em uma pipeline inspirada nas etapas convencionais de um *Image Signal Processor* (ISP). Entre as etapas consideradas estão a correção de exposição, remoção de ruído, balanço de branco, correção de cor e aplicação do algoritmo CLAHE (*Contrast Limited Adaptive Histogram Equalization*), com o objetivo de obter uma imagem final mais

nítida, fiel em cores e perceptualmente agradável.

Além do aprimoramento visual, o trabalho também se dedica à avaliação objetiva da qualidade da imagem resultante. Para isso, serão utilizadas métricas como o NIQE (*Naturalness Image Quality Evaluator*) e o BRISQUE (*Blind/Referenceless Image Spatial Quality Evaluator*), que possibilitam mensurar a eficácia das transformações aplicadas de forma técnica e reproduzível e sem uma imagem referência. Assim, esta pesquisa contribui com uma abordagem prática, transparente e reprodutível para o problema do processamento de imagens noturnas, sem depender de bases de dados extensas ou treinamento de modelos, o que reforça seu valor tanto acadêmico quanto aplicado, especialmente em contextos onde a simplicidade, a explicação e a modularidade são essenciais.

1.3 Objetivo Geral

Desenvolver uma pipeline tradicional de processamento de imagens *RAW* capturadas em ambientes de baixa luminosidade, utilizando técnicas tradicionais de PDI, com o intuito de aprimorar aspectos visuais como exposição, ruído e cores, e avaliar a qualidade das imagens resultantes por meio de métricas objetivas.

1.4 Objetivos Específicos

- Realizar um levantamento técnico sobre os fundamentos e princípios do processamento de imagens em formato *RAW*.
- Projetar e estruturar uma pipeline de Processamento Digital de Imagem, com blocos bem definidos que permitam intervenções e testes isolados.
- Implementar algoritmos para as etapas da pipeline: correção de exposição, remoção de ruído, balanço de branco, correção de cor, controle de gama, *CLAHE* e *Dehaze*.

- Analisar a qualidade final das imagens, com ênfase na preservação de detalhes finos, texturas e nitidez.
- Comparar quantitativamente os métodos implementados, apresentando análise comparativa entre técnicas e configurações.

1.5 Organização do Trabalho

Este trabalho está organizado em seis capítulos, estruturados de forma a conduzir o leitor desde os fundamentos teóricos até a análise dos resultados obtidos.

O **Capítulo 1** apresenta o tema, os objetivos gerais e específicos, a contextualização do problema, bem como a motivação e a justificativa que sustentam a realização desta pesquisa.

O **Capítulo 2** reúne a fundamentação teórica necessária para o desenvolvimento do estudo, abrangendo conceitos relacionados à formação da imagem digital, características dos sensores, especificidades do formato *RAW*, etapas clássicas do fluxo ISP e métricas de avaliação de qualidade de imagem.

No **Capítulo 3**, descreve-se a metodologia adotada, detalhando a abordagem empregada na concepção da pipeline, as ferramentas computacionais utilizadas e o delineamento experimental que orienta o processo de desenvolvimento.

O **Capítulo 4** é dedicado à apresentação do desenvolvimento da solução proposta, contemplando a implementação dos módulos que compõem a pipeline e a descrição das técnicas aplicadas em cada etapa.

O **Capítulo 5** expõe os resultados obtidos e discute seu desempenho a partir da análise qualitativa e quantitativa das imagens processadas, incluindo a comparação entre diferentes configurações e métricas utilizadas.

Por fim, o **Capítulo 6** sintetiza as conclusões alcançadas, discute as limitações identificadas ao longo da pesquisa e apresenta sugestões para a continuidade e aprofundamento deste trabalho em investigações futuras.

2

FUNDAMENTAÇÃO TEÓRICA

2.1 Imagem Digital

Uma imagem digital constitui uma representação discreta bidimensional de uma cena visual, discretizada em elementos pictóricos, denominados pixels, onde cada pixel codifica a intensidade luminosa e para imagens coloridas no espaço RGB (*red, blue e green*), os valores de intensidade nos canais espectrais vermelho, verde e azul. Formalmente, do ponto de vista computacional, uma imagem é representada por uma matriz bidimensional de valores escalares discretos. Cada elemento (i, j) desta matriz contém o nível de cinza (em imagens monocromáticas) ou um vetor de intensidades nos canais RGB em imagens coloridas.(GONZALEZ; WOODS, 2018). Cada pixel pode armazenar um ou mais valores, dependendo do modelo de cor. Em imagens coloridas no padrão RGB, como dito anteriormente, cada pixel é representado por três componentes de intensidade, comumente quantificados em 8 bits por canal, o que permite $2^8=256$ níveis diferentes por canal. Computacionalmente, essa estrutura é modelada como uma matriz matemática que pode ser representada por $I_{m \times n}$, onde cada elemento $I(i, j)$ contém:

- Valor escalar, em níveis de cinza: $I(i, j) [0, L_{max}]$, onde 0 representa a cor preta e 255 representa a cor branca para imagens em 8 bits
- Vetor RGB: $I(i, j) = [R, G, B] [0...255]$, espaço de cores padrão.

2.2 Processamento de imagem

Um marco significativo na aplicação prática do Processamento Digital de Imagens (PDI) remonta aos anos 1920, quando sistemas eletrônicos foram implementados para codificar e transmitir fotografias por meio de cabos submarinos que cruzavam o Oceano Atlântico. Inicialmente, essas transmissões utilizavam imagens em escala de cinza, limitadas a apenas cinco níveis de intensidade luminosa. Contudo, a tecnologia avançou rapidamente e, ainda na mesma década, a capacidade de reprodução tonal foi ampliada para até quinze tons de cinza. (GONZALEZ; WOODS, 2018)

Embora os sistemas contemporâneos de aquisição digital de imagens possam apresentar completa dissociação dos métodos analógicos, seus fundamentos físico-ópticos remontam ao princípio da câmara obscura, primeiramente descrito cientificamente por Ibn al-Haytham em seu tratado Livro da Óptica, por volta de 1021 (AL-HAYTHAM, 1021). Nesse sistema pioneiro, a formação imagética ocorre mediante uma sequência de fenômenos ópticos: (1) a criação de um orifício estenopeico (pinhole) numa superfície opaca, (2) a passagem unidirecional de raios luminosos, (3) a projeção de uma imagem invertida na parede oposta, decorrente do princípio da propagação retilínea da luz, e (4) sua captura em uma superfície receptora (HANNAN, 2016).

Contudo, essa tecnologia tem experimentado evolução acelerada nas últimas décadas. Atualmente, a aquisição e o processamento de imagens alcançou níveis de acessibilidade universal, sendo realizado inclusive por dispositivos móveis. Essa onipresença tecnológica torna suas operações imperceptíveis aos usuários, dada a praticidade e transparência com que tais procedimentos são executados na contemporaneidade.

2.3 Aquisição de imagem e ISP

A aquisição de imagens constitui a etapa primordial em qualquer sistema de processamento digital de imagens, consistindo na conversão de uma cena do mundo real em uma representação numérica discreta. Este processo fundamenta-se na captura da radiação eletromagnética (luz visível, infravermelho, etc.) refletida ou emitida pelos objetos, mediante sensores optoeletrônicos. Tais sensores convertem o sinal luminoso em elé-

trico, o qual é subsequentemente quantizado, armazenado e processado. (GONZALEZ; WOODS, 2018)

2.4 Princípios Fundamentais da Aquisição

Os sistemas contemporâneos empregam predominantemente sensores baseados em tecnologia CCD (*Charge-Coupled Device*) ou CMOS (*Complementary Metal-Oxide-Semiconductor*). Ambos operam mediante uma matriz de fotodiodos (fotossítios), onde cada elemento gera uma carga elétrica proporcional à intensidade luminosa incidente (HOLST; LOMHEIM, 2011). Contudo, diferenciam-se significativamente em arquitetura e desempenho:

- **Sensores CCD:** Caracterizam-se por elevada sensibilidade fotométrica e baixo ruído de leitura (*read noise*), resultando em excelente relação sinal-ruído (SNR). Esta qualidade os torna preferenciais em aplicações científicas, astronômicas e fotografia profissional de alta gama (NORTON, 2001).
- **Sensores CMOS:** Oferecem vantagens em termos de custo reduzido, menor consumo energético e integração de circuitos periféricos. Predominam em dispositivos como smartphones e câmeras DSLR (NAYAR; BRANZOI, 2003).

A radiação captada pelo sensor é tipicamente submetida a um filtro de Bayer – um mosaico que permite que cada pixel registre apenas uma das cores primárias (RGB). Consequentemente, exige-se interpolação cromática (*demosaicing*) para reconstruir os valores RGB faltantes (RAMANATH et al., 2005), processo que pode introduzir artefatos

2.5 Modalidades de Aquisição

Existe diversos tipos de aquisição de imagens como podemos ver a seguir:

- **Imagem Monocromática:** Captura apenas a intensidade luminosa, gerando uma matriz em escala de cinza.

- **Imagem Colorida RGB:** Requer separação espectral via filtro de Bayer, prismas dicróicos ou sensores em camadas (GONZALEZ; WOODS, 2018).
- **Imagem Multiespectral/Hiperspectral:** Captura múltiplas bandas espectrais estreitas (LANDGREBE, 2003).
- **Imagem Térmica (Infravermelho Termal):** Detecta radiação emitida por calor intrínseco (ROGALSKI, 2012).
- **Formato RAW:** Armazena dados mínimos processados do sensor, preservando máxima informação original (FRASER, 2004).

2.6 Modalidades de Aquisição

A imagem é armazenada em formatos padronizados, classificáveis conforme a estratégia de compressão (SKODRAS; CHRISTOPOULOS; EBRAHIMI, 2001):

- **Formatos com Compressão com Perdas:**
 - JPEG: utiliza Transformada Discreta do Cosseno (DCT) e quantização (WALLACE, 1991).
 - WebP: oferece melhor compressão que JPEG em alguns casos e suporta transparência (NEDERMEIJER et al., 2013).
 - HEIF/HEIC: container eficiente baseado em compressão HEVC; armazena mais dados com menos espaço, amplamente utilizado em dispositivos móveis (SULLIVAN et al., 2012).
- **Formatos sem Compressão ou com Compressão sem Perdas:**
 - RAW: preserva os dados brutos do sensor (FRASER, 2004).
 - TIFF: suporta alta profundidade de cor e compressão sem perdas (MURRAY; VANRYPER, 1996).
 - PNG: utiliza compressão DEFLATE e suporta transparência (BOUTELL, 1997).

2.7 Formato RAW: Aquisição, Relevância e Aplicações

O formato RAW representa uma das formas mais puras de aquisição de imagem digital. Ele corresponde à leitura direta dos dados capturados pelo sensor de imagem antes que qualquer tipo de compressão, correção ou processamento seja aplicado. Diferentemente de formatos processados (e.g., JPEG), o RAW mantém intactos os valores quantizados provenientes do conversor analógico-digital (ADC), dispensando operações como interpolação cromática, balanceamento de branco ou compressão com perdas. Como resultado, as imagens RAW preservam a maior quantidade possível de informações físicas da cena registrada. (FRASER, 2004; GONZALEZ; WOODS, 2018).

Durante a captura, sensores CCD ou CMOS convertem fótons incidentes em carga elétrica, posteriormente amplificada e quantizada pelo ADC. Em sensores com filtro de Bayer, cada fotossítio registra apenas uma componente espectral (R, G ou B), gerando uma matriz de valores monocromáticos (*RAW mosaic*) (RAMANATH et al., 2005).

Conforme ilustra Farrell et al. (FARRELL et al., 2011, p. 72):

"O arquivo RAW encapsula três elementos essenciais: (1) a matriz de valores digitais diretos do ADC; (2) metadados operacionais (ISO, tempo de exposição, curva de resposta tonal nativa); (3) parâmetros de correção óptica/radiométrica não aplicados, mas armazenados como referência."

Essa estrutura resulta em arquivos volumosos (12–16 bits/canal vs. 8 bits do JPEG), porém com faixa dinâmica superior e preservação integral do sinal eletrônico (HOLST; LOMHEIM, 2011).

A primazia do RAW reside em sua capacidade de preservar a informação física da cena, viabilizando ajustes não destrutivos e análises quantitativas. Sua aplicação é crítica em cenários como:

1. Cenas de alta faixa dinâmica (HDR): Recuperação de detalhes em regiões subexpostas/superexpostas via *tone mapping* (REINHARD et al., 2010);
2. Condições de iluminação adversas: Controle preciso de ruído térmico e de disparo (*shot noise*) em baixa luminosidade (FARRELL et al., 2011);

3. Análises forenses/científicas: Validação de integridade de imagem e extração de dados radiométricos para calibração (KRIZ; HUBBARD; CHEN, 2016);
4. Reconstrução fotométrica: Modelagem de propriedades de materiais e fontes luminosas (DEBEVEC; MALIK, 1997).

A manipulação de RAW oferece vantagens decisivas sobre formatos processados:

Parâmetro	Vantagem RAW	Base Técnica
Balanco de branco	Ajuste não destrutivo pós-captura via ganhos de canal	Valores RGB nativos não balanceados (FRASER, 2004)
Profundidade de bits	Edições tonais sem <i>banding</i> (12–14 bits vs. 8 bits)	Quantização fina de níveis de cinza (GONZALEZ; WOODS, 2018)
Faixa dinâmica	Recuperação de ~ 1 –2 EV adicionais em sombras/altas-luzes	Curva tonal linear vs. curva S aplicada em JPEG (REINHARD et al., 2010)
Redução de ruído	Aplicação de algoritmos avançados (e.g., wavelet, ML) com parâmetros customizados	Preservação de estatísticas de ruído originais (FARRELL et al., 2011)

Tabela 1 – Vantagens do formato RAW e sua base técnica.

2.8 Avaliação da qualidade de Imagem

A Avaliação da Qualidade de Imagem (Image Quality Assessment - IQA) constitui-se como um processo sistemático de quantificação da fidelidade perceptual ou estética de imagens processadas. Esta quantificação fundamenta-se na comparação com padrões ideais (imagens de referência) ou na aplicação de modelos computacionais que emulam a percepção visual humana, particularmente em cenários onde uma referência explícita não está disponível. A IQA desempenha um papel crítico na validação de algoritmos destinados ao aprimoramento, compressão, transmissão ou restauração de conteúdo visual (WANG; BOVIK, 2006).

A avaliação da qualidade de imagens estrutura-se em dois paradigmas metodológicos distintos:

- **Avaliação Subjetiva:** A avaliação é realizada por meio da análise visual direta, onde observadores humanos atribuem juízos qualitativos (como notas ou preferências) com base em atributos perceptuais, como nitidez, fidelidade cromática e apelo estético. Embora esse método se aproxima da percepção sensorial real, sua confiabilidade é comprometida pela variabilidade entre os observadores e suas preferências. (WANG; BOVIK, 2006).
- **Avaliação Objetiva:** Conduzida por algoritmos computacionais que quantificam degradações mediante modelos matemáticos (e.g., análise espectral, estatísticas de textura ou redes neurais). Oferece vantagens metodológicas decisivas: reprodutibilidade analítica, eficiência computacional e aplicabilidade em fluxos automatizados. Contudo, enfrenta o desafio fundamental de correlacionar-se com a experiência humana, pois métricas convencionais (e.g., PSNR) frequentemente negligenciam nuances perceptivas como sensibilidade contextual a artefatos ou expectativas estéticas (SHEIKH; BOVIK, 2006).

3

TRABALHOS RELACIONADOS

A melhoria de imagens capturadas em ambientes com baixa luminosidade tem despertado crescente interesse na literatura científica, impulsionada, em grande parte, pelos avanços em técnicas de aprendizado profundo. Esses estudos buscam enfrentar desafios recorrentes em imagens noturnas ou subexpostas, como alto nível de ruído, baixa razão sinal-ruído (*SNR*), perda de cor e contraste (LORENTZ et al., 2021). As abordagens propostas na literatura variam desde pipelines determinísticos tradicionais até modelos supervisionados com elevada complexidade computacional (ZHANG et al., 2021). Esta seção apresenta os principais trabalhos contemporâneos sobre o tema, com ênfase em suas propostas metodológicas, objetivos, técnicas empregadas e resultados obtidos.

(CHEN et al., 2018) propõem uma abordagem pioneira baseada em *deep learning* para gerar imagens de alta qualidade em condições de iluminação extremamente baixa. Seu método processa dados *RAW* do sensor diretamente por meio de uma rede neural convolucional (CNN) de ponta a ponta, realizando *demosaicing*, redução de ruído e ajuste tonal de forma integrada. Para viabilizar o treinamento, os autores introduziram o *dataset See-in-the-Dark* (SID), composto por pares alinhados de imagens *RAW* captadas com curta exposição/alta ISO (condição "escura") e longa exposição/baixo ISO (referência). A solução supera métodos tradicionais em preservação de detalhes, fidelidade cromática e redução de ruído, estabelecendo um novo paradigma para o processamento de imagens em baixa luz.

Em continuidade, (ZAMIR et al., 2019) expandiram essa abordagem no trabalho *Learning Digital Camera Pipeline for Extreme Low-Light Imaging*, no qual apresentam uma

nova arquitetura neural residual capaz de simular toda a pipeline de uma câmera digital, desde a leitura do sinal *RAW* até a produção da imagem RGB final. O modelo é treinado de forma *end-to-end*, incorporando dentro da própria rede etapas como interpolação cromática, correção de cores e compressão tonal. A rede é otimizada com múltiplas funções de perda, incluindo *perceptual loss*, *L1 loss* e *SSIM loss*, e foi projetada para operar em condições de iluminação extremamente adversas, onde sensores convencionais falham em produzir imagens úteis. Os autores demonstraram que o método supera abordagens anteriores nos principais *benchmarks* do *dataset SID*, tanto em métricas quantitativas quanto qualitativas. No entanto, o modelo mantém as limitações típicas de redes profundas: forte dependência de dados supervisionados, baixa interpretabilidade e elevado custo computacional na inferência.

Por outro caminho, (GUO et al., 2021) apresentaram o *Zero-DCE – Zero-reference Deep Curve Estimation*, um modelo leve e não supervisionado que realiza o aprimoramento de imagens escuras sem a necessidade de imagens de referência. O modelo interpreta o realce como um problema de ajuste de curvas de iluminação aplicadas diretamente à imagem de entrada, guiado por critérios perceptuais extraídos da própria imagem. O treinamento é realizado com múltiplas funções de perda que avaliam atributos como exposição média, suavidade local, não saturação e preservação de detalhes. O *Zero-DCE* demonstrou excelente desempenho em métricas sem referência, como BRISQUE e NIQE, sendo especialmente eficaz em aplicações em tempo real e dispositivos com recursos limitados. Entretanto, sua aplicação é restrita a imagens já convertidas (por exemplo, JPEG), não operando diretamente sobre arquivos *RAW*, além de oferecer menor controle e fidelidade visual em comparação com modelos supervisionados.

(XU et al., 2022), por sua vez, propuseram o modelo híbrido *SNR-Aware Low-Light Image Enhancement*, que combina convoluções profundas com mecanismos de atenção baseados em *transformers*. A principal inovação do trabalho reside na introdução de um mapa de SNR gerado dinamicamente, o qual permite à rede focar seletivamente em regiões mais degradadas da imagem, melhorando localmente áreas com maior ruído sem afetar regiões já bem expostas. A arquitetura apresentou desempenho superior aos modelos *SID*, *Zero-DCE* e *EnlightenGAN* em diversos *benchmarks* públicos, destacando-

se também pela capacidade de generalização em sensores distintos dos utilizados no treinamento. Ainda assim, o modelo apresenta elevada complexidade estrutural e demanda computacional significativa, o que limita sua aplicação em contextos que exigem leveza, interpretabilidade e eficiência operacional.

Complementando essas abordagens, (RASHEED; SHI; KHAN, 2023) realizaram uma revisão sistemática experimental, avaliando mais de 20 métodos de realce de imagens noturnas, incluindo soluções clássicas e baseadas em aprendizado. Os autores aplicaram métricas objetivas com e sem referência (PSNR, SSIM, BRISQUE, NIQE) em múltiplos conjuntos de dados, concluindo que, embora os métodos supervisionados superem tecnicamente os tradicionais em cenários controlados, nenhuma abordagem se mostrou universalmente superior. O estudo destaca que técnicas baseadas em aprendizado profundo tendem a apresentar artefatos, perda de detalhes e instabilidade quando aplicadas a cenários significativamente diferentes daqueles observados nos dados de treinamento. Ademais, ressalta-se a importância das abordagens clássicas por sua interpretabilidade, reprodutibilidade e maior adequação a contextos embarcados ou com restrições operacionais.

Diante desse panorama, a proposta deste trabalho adota uma abordagem tradicional e modular, fundamentada em uma pipeline explícita de processamento de imagens RAW. Tal abordagem permite a análise detalhada de cada etapa do processo, bem como ajustes específicos, comparações sistemáticas e avaliação transparente dos resultados com base em métricas objetivas, como NIQE e BRISQUE. Essa estratégia busca um equilíbrio entre qualidade perceptual, eficiência prática e aplicabilidade, sendo especialmente adequada para aplicações em que o uso de redes neurais profundas não é viável ou desejável.

Trabalho	NIQE	BRISQUE	PSNR	MOS-NIQE	SSIM
Chen et al. (2018)			X		X
Zamir et al. (2020)			X		
Guo et al. (2021)	X	X		X	
XU et al., 2022			X		
Rasheed et al. (2023)	X	X			

Tabela 2 – Comparação de trabalhos e técnicas aplicadas em avaliação de imagens de baixa luminosidade

O trilema "Desempenho vs. Eficiência vs. Generalização", sintetizado por (RASHEED; SHI; KHAN, 2023), permanece como o principal desafio do campo. Este quadro comparativo fundamenta a decisão metodológica do presente trabalho de adotar uma pipeline modular e interpretável, que busca um ponto de equilíbrio prático entre esses fatores, priorizando o controle sobre cada etapa do processamento e a validação por meio de métricas NIQE e BRISQUE.

A partir da análise dos trabalhos relacionados, verifica-se que parte das abordagens destinadas à melhoria de imagens em condições de baixa luminosidade concentra-se no emprego de técnicas baseadas em aprendizado de máquina ou redes neurais profundas. Tais métodos, embora capazes de produzir resultados expressivos em termos de realce visual e recuperação de detalhes, geralmente requerem extensos conjuntos de dados para treinamento, apresentam elevado custo computacional e oferecem menor interpretabilidade quanto aos processos internos envolvidos.

Em contraste, o presente estudo adota uma abordagem fundamentada em uma pipeline tradicional de Processamento de Sinal de Imagem (ISP), operando diretamente sobre imagens no formato RAW. Essa escolha possibilita maior controle sobre cada etapa do processamento, bem como a preservação das informações originais capturadas pelo sensor. Ademais, este trabalho enfatiza também a avaliação objetiva da qualidade das imagens geradas, empregando métricas sem referência, como BRISQUE e NIQE, o que permite uma análise quantitativa e comparativa dos efeitos produzidos por cada bloco da pipeline.

Dessa forma, o método proposto configura-se como uma alternativa interpretável e reprodutível às abordagens baseadas em aprendizado profundo, contribuindo para a compreensão do impacto individual das técnicas clássicas de processamento de imagem na melhoria de imagens em cenários de baixa luminosidade.

4

METODOLOGIA

Este trabalho foi desenvolvido por meio da implementação de um pipeline tradicional de processamento de imagem digital, aplicada a arquivos *RAW* capturados em ambientes de baixa luminosidade. O pipeline é composta por 11 blocos modulares como podemos ver na Figura 1, organizados de acordo com as etapas típicas de um *Image Signal Processor* (ISP), permitindo o aprimoramento gradual da imagem original. A linguagem de programação utilizada foi o *Python*, por sua ampla gama de bibliotecas dedicadas ao processamento de imagens, como *OpenCV*, *RawPy*, *NumPy* e *Scikit-Image*. A entrada do sistema é composta por arquivos no formato *RAW*, especificamente do tipo *ARW* (utilizado por câmeras da marca Sony), que são carregados por meio da biblioteca *RawPy*, a qual permite acesso aos dados brutos do sensor da câmera.

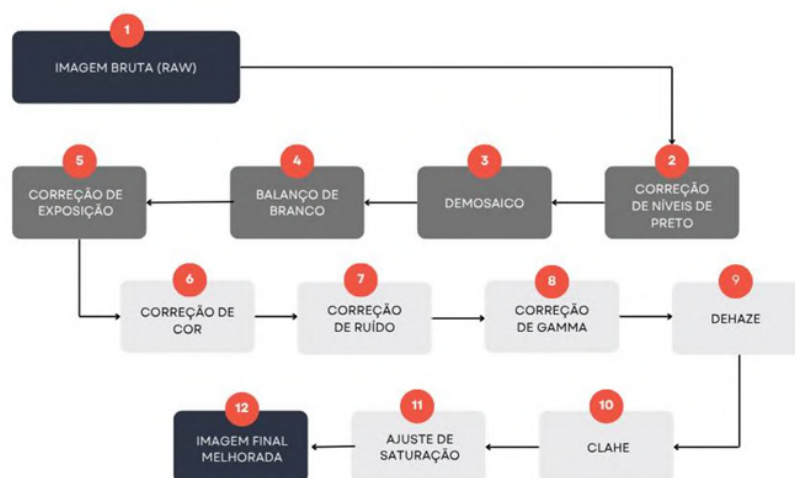


Figura 1 – Diagrama de blocos do sistema de processamento.

Fonte: Autoria Própria (2025).

No diagrama da figura 1, é possível dividir o pipeline em etapas distintas. Os blocos representados em tom de cinza escuro correspondem às etapas iniciais de leitura e processamento básico da imagem, responsáveis por correções fundamentais e estruturação dos dados brutos. Já os blocos em cinza claro indicam as etapas voltadas à melhoria da exposição e aos ajustes específicos para imagens capturadas em condições de baixa luminosidade. Conseqüentemente, a estrutura metodológica deste trabalho organiza-se em dois eixos complementares: a primeira fase dedica-se à descrição integral da pipeline de processamento, detalhando sequencialmente cada estágio transformacional; enquanto a segunda fase concentra-se na sistematização do protocolo avaliativo, estabelecendo critérios quantitativos e qualitativos para mensuração dos resultados obtidos.

4.1 Pipeline de Processamento

4.1.1 Leitura da Imagem bruta - RAW

A primeira etapa da pipeline consiste na leitura da imagem bruta no formato RAW, que corresponde à representação digital primária dos dados capturados pelo sensor da câmera, antes da aplicação de qualquer processamento interno do Image Signal Processor (ISP). Este formato preserva a informação radiométrica original dos fotodiodos, mantendo intactas características como a profundidade de bits nativa (geralmente 12 a 14 bits por canal) e a organização espacial do filtro de Bayer, sem a aplicação de etapas como interpolação cromática (demosaicing), balanceamento de branco ou compressão com perdas (FRASER, 2004; GONZALEZ; WOODS, 2018).

Para este trabalho, foram utilizados arquivos no formato .ARW (*Sony Alpha RAW*), padrão proprietário da Sony que encapsula os dados brutos do sensor juntamente com metadados técnicos (*EXIF*) essenciais para a interpretação correta dos valores de pixel, como níveis de preto (*black level*), ganhos dos canais colorimétricos (*white balance gains*) e curva de resposta tonal (SONY, 2023).

A escolha deste formato deve-se à sua capacidade de preservar a máxima faixa dinâmica e à ampla disponibilidade de conjuntos de dados públicos que o utilizam,

como o *See-in-the-Dark (SID) Dataset* (CHEN et al., 2018).

4.1.2 Correção de Nível de Preto

Após a leitura dos dados brutos, a primeira operação de calibração radiométrica aplicada é a Correção de Nível de Preto (*Black Level Correction*). Esta etapa fundamental remove o *offset* eletrônico intrínseco ao sensor, conhecido como *black level*, que corresponde ao valor mínimo de intensidade registrado pelo fotodiodo na ausência total de fótons incidentes. Sensores CCD e CMOS, por características de seu design e ruído de leitura (*read noise*), não geram um sinal nulo em condições de escuridão, criando um viés positivo que, se não corrigido, compromete a fidelidade radiométrica, distorce o balanço de branco e reduz a faixa dinâmica útil da imagem (HOLST; LOMHEIM, 2011; GONZALEZ; WOODS, 2018).

A forma mais direta de correção é a subtração simples deste *offset* (Δ) de cada pixel da matriz *Bayer* bruta:

$$I_{\text{corr}(x,y)=\max(I_{\text{raw}(x,y)}-\Delta,0)} \quad (4.1)$$

Essa operação elimina o viés de fundo e garante que áreas sem iluminação apareçam corretamente como preto absoluto. No entanto, embora essa subtração corrija o nível mínimo, ela não compensa a compressão da faixa dinâmica útil causada pela presença do *offset*. Para superar esta limitação, especialmente crítica em cenários de baixa luminosidade onde a informação da cena se concentra predominantemente nas regiões mais escuras do histograma, este trabalho adota uma correção de nível de preto com normalização. Esta técnica, amplamente recomendada em pipelines profissionais (FRASER, 2004), não apenas remove o *offset*, mas também reescala linearmente os valores para ocupar toda a extensão da faixa dinâmica codificável. A operação é definida pela seguinte transformação:

$$I_{\text{norm}}(x,y) = \frac{(X - \Delta)}{V_{\text{max}} - \Delta} \times V_{\text{max}} \quad (4.2)$$

Onde:

- x representa o valor original do pixel;
- Δ é o valor do *black level* (*offset* eletrônico);
- V_{\max} é o valor máximo da faixa de codificação do sensor (por exemplo, 4095 para 12 bits).

Essa equação possui duas vantagens principais:

- Garante que o valor mínimo $x = \Delta$ seja mapeado para zero;
- Garante que o valor máximo $x = V_{\max}$ permaneça inalterado, preservando a linearidade da faixa útil.

Esse tipo de normalização é especialmente relevante em contextos de baixa luminosidade, nos quais grande parte da imagem pode estar concentrada nas faixas inferiores da escala de intensidade. Ao reescalar, a distribuição tonal da imagem é expandida, melhorando a eficácia de etapas como mapeamento de tons, ajuste de exposição e equalização de contraste.

4.1.3 Demosaico

A etapa de *demosaicing*, também denominada interpolação cromática, constitui um processo fundamental no pipeline de processamento de imagens digitais. Sua função é reconstruir uma imagem colorida completa (RGB) a partir da matriz de valores monocromáticos capturada pelo sensor, na qual cada pixel registra apenas uma componente espectral conforme o padrão do filtro de Bayer ([BAYER, 1976](#)). Este padrão, predominantemente utilizado em sensores de imagem, emprega uma distribuição de microfiltros sobre os fotodiodos, sendo o arranjo RGGB (*Red, Green, Green, Blue*) o mais comum. Nesta configuração, 50% dos pixels são sensíveis ao verde, enquanto 25% são sensíveis ao vermelho e 25% ao azul, priorizando o canal verde devido à maior sensibilidade do sistema visual humano à luminância, que é mais precisamente estimada por este comprimento de onda ([GONZALEZ; WOODS, 2018](#); [MALVAR; HE; CUTLER, 2004](#)).

- **Interpolação Bilinear:** Método baseline que utiliza média simples dos vizinhos mais próximos. É computacionalmente eficiente, mas introduz desfoque significativo e artefatos de cor em regiões de alto detalhe (GUNTURK; ALTUNBASAK, 2002).
- **Interpolação Baseada em Gradiente (Edge-Sensing):** Considera a direção de bordas locais para direcionar a interpolação, melhorando a preservação de contornos. Exemplos incluem os algoritmos de (HAMILTON; ADAMS, 1997).
- **Métodos Edge-Aware ou Edge-Directed:** Utilizam critérios de homogeneidade para adaptar o kernel de interpolação, reduzindo *zippering*. O algoritmo AHD (Adaptive Homogeneity-Directed), proposto por (HIRAKAWA; PARKS, 2005), é um representante desta classe e se baseia na seleção da direção (horizontal ou vertical) que maximiza a homogeneidade cromática em uma janela local.
- **Abordagens em Domínio de Frequência e Baseadas em Otimização:** Técnicas mais avançadas que modelam o processo como um problema de reconstrução, utilizando wavelets ou regularização (ZHANG; WU, 2005).

Neste trabalho, optou-se pela implementação do algoritmo AHD, disponibilizada pela biblioteca *rawpy* através de sua interface com a *Libraw*. Esta escolha fundamenta-se no equilíbrio comprovado entre qualidade perceptual e custo computacional, sendo amplamente adotada em pipelines profissionais e softwares de conversão RAW.

A saída deste estágio é uma imagem RGB linear de três canais completos, ainda no espaço de cor do sensor (camera RGB space), que serve como entrada para as correções colorimétricas e ajustes tonais subsequentes. A fidelidade desta reconstrução impacta diretamente a qualidade final, especialmente em regiões de alto contraste e padrões repetitivos, onde algoritmos mais simples falham visivelmente (HIRAKAWA; PARKS, 2005).

A importância do *demosaicing* está na fidelidade da reconstrução cromática. Um algoritmo mal ajustado pode introduzir artefatos de cor (*moiré*), falsas bordas ou desfoque. Por isso, sua escolha influencia diretamente a qualidade perceptiva da imagem

processada, principalmente em áreas de alto contraste ou com padrões geométricos finos.

4.1.4 Balanço de Branco

O Balanço de Branco (*White Balance*) constitui uma etapa crítica no pipeline de processamento digital de imagens, responsável pela compensação perceptual e física dos efeitos da temperatura de cor da fonte de luz incidente. Sensores de imagem são dispositivos radiométricos que registram a intensidade da radiação eletromagnética, sem a capacidade inerente de distinguir entre a refletância intrínseca de um objeto e o espectro da fonte de iluminação. Consequentemente, um objeto branco ou acromático pode ser registrado com uma dominante de cor indesejada (e.g., azulada sob luz diurna, avermelhada sob luz incandescente), fenômeno conhecido como *color cast* (GONZALEZ; WOODS, 2018; FARRELL et al., 2011).

A compensação é feita por meio da multiplicação dos canais RGB da imagem por fatores escalares distintos para cada canal. Isso corrige o desbalanceamento entre os valores de vermelho, verde e azul, originado pelas características espectrais da luz ambiente e pela resposta do sensor.

A operação pode ser expressa pela seguinte equação:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} g_R & 0 & 0 \\ 0 & g_G & 0 \\ 0 & 0 & g_B \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Neste trabalho, foi adotada a técnica conhecida como Gray World Assumption (BUCHSBAUM, 1980), ou hipótese do mundo cinza, um método automático que assume que, em média, uma cena natural contém a mesma quantidade de vermelho, verde e azul. Com base nisso, calcula-se a média de intensidade de cada canal de cor na imagem, e os canais com menor média recebem ganho proporcional para igualar-se à média global.

4.1.5 Correção de Exposição

Imagens capturadas em ambientes com baixa luminosidade tendem a apresentar subexposição severa, especialmente nos canais menos sensíveis à luz (vermelho e azul). A etapa de correção de exposição tem como objetivo realçar essas imagens, ampliando sua faixa de brilho de forma controlada, sem saturar regiões claras ou degradar áreas escuras.

A exposição de uma imagem digital é uma função direta da quantidade total de energia luminosa (integral de fótons) que atinge o sensor, controlada pelos parâmetros de captura: tempo de obturador, abertura da lente e sensibilidade (ISO). Uma das vantagens fundamentais do processamento sobre dados RAW lineares é a preservação da relação de proporcionalidade direta entre o valor digital do pixel e a quantidade de luz incidente. Esta linearidade permite que ajustes de exposição sejam realizados a posteriori através de uma simples operação de escalonamento, simulando com alta fidelidade o efeito de uma maior captura de luz no momento do *click* (FRASER, 2004; GONZALEZ; WOODS, 2018).

Neste trabalho, a correção de exposição é aplicada sobre a imagem RGB linear já corrigida no balanço de branco, por meio de uma multiplicação escalar simples entre a imagem e um valor de ganho simples de 0.3, valor esse de ganho que pode ser alterado dependendo da imagem de forma manual.

4.1.6 Correção de Cor – Matriz de Correção Cromática (CCM)

Após a aplicação do balanço de branco, a imagem ainda se encontra no espaço de cor nativo do sensor da câmera, o qual difere significativamente de espaços de cor padronizados como sRGB. Isso ocorre porque cada sensor possui uma resposta espectral específica, influenciada por fatores como a construção física do fotodiodo, os filtros de cor aplicados e a eletrônica de leitura. Para garantir que as cores capturadas sejam perceptualmente corretas e consistentes com as expectativas humanas, é necessário aplicar uma matriz de correção de cor – conhecida como *Color Correction Matrix (CCM)*.

A CCM é uma matriz 3×3 que transforma os valores RGB capturados pelo sensor

para um espaço de cor intermediário, como XYZ, ou diretamente para sRGB. Essa transformação pode ser expressa pela seguinte equação vetorial:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = CCM \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

A aplicação desta matriz permite ajustar o espectro de resposta do sensor para alinhar-se ao modelo de cor perceptual humano. Isso inclui a compensação de variações na sensibilidade espectral dos canais, resultando em uma representação mais precisa de tons de pele, vegetação, céu, entre outros elementos presentes na cena. Neste trabalho, a matriz CCM utilizada foi extraída diretamente dos metadados do arquivo RAW, por meio da biblioteca `rawpy`, que fornece acesso ao campo `color_matrix`. Essa matriz é definida pelo fabricante da câmera com base no modelo do sensor e na calibração de fábrica. Para compatibilidade com a estrutura da pipeline e consistência nos formatos de entrada de dimensões 3×3.

4.1.7 Redução de Ruído

O desafio central de um algoritmo de denoising é atenuar seletivamente as flutuações aleatórias enquanto preserva integralmente as estruturas geométricas e os contornos de alta frequência (bordas, texturas). Enquanto filtros lineares (e.g., Gaussiano) suavizam uniformemente, borrando estruturas importantes, métodos avançados buscam uma separação mais inteligente entre sinal e ruído.

Neste trabalho, optou-se pelo Filtro Bilateral, um método não-linear, não iterativo e *edge-preserving* proposto por (TOMASI; MANDUCHI, 1998). Sua principal vantagem reside em considerar dois domínios simultaneamente:

- Domínio Espacial: Proximidade geométrica entre pixels (como um filtro Gaussiano).
- Domínio da Intensidade (*Range*): Similaridade fotométrica entre os valores dos pixels.

4.1.8 Correção Gama

Após as etapas lineares de realce e redução de ruído, a imagem encontra-se em um espaço de cor linear (RGB linear), onde a relação entre os valores digitais e a intensidade luminosa é diretamente proporcional. No entanto, tanto a percepção visual humana quanto a resposta dos dispositivos de exibição padrão (monitores, telas) são inerentemente não-lineares. A percepção humana da luminosidade segue aproximadamente uma lei de potência com expoente 0.33 a 0.5 (lei de Stevens), sendo mais sensível a variações nas sombras do que nas altas-luzes (PATTANAIK et al., 1998). Simultaneamente, os dispositivos de exibição possuem uma resposta eletro-óptica não-linear, historicamente herdada dos tubos de raios catódicos (CRT).

Para compensar essas não-linearidades e codificar os dados de imagem de forma eficiente para armazenamento e exibição, aplica-se a correção gama (*gamma encoding*). Esta etapa é uma transformação ponto-a-ponto que comprime os tons claros e expande os tons escuros, alocando mais bits de profundidade às regiões onde o sistema visual humano é mais sensível, otimizando assim o uso da faixa dinâmica discreta (e.g., 8 bits por canal) (REINHARD et al., 2010).

Diferentemente de transformações logarítmicas ou mapeamentos tonais complexos, a correção gama é uma função de potência que ajusta as intensidades de maneira contínua e não linear. Sua forma geral é dada por:

$$I'(x, y) = I(x, y)^{1/\gamma} \quad (4.3)$$

Onde,

- $I(x, y)$: Representa a intensidade luminosa linear do pixel localizado na posição espacial (x, y) da imagem original, diretamente proporcional à quantidade de luz capturada pelo sensor.
- $I'(x, y)$ Corresponde à intensidade do pixel após a aplicação da correção gama, já ajustada para a percepção visual humana e adequada à exibição em dispositivos digitais.
- x, y : Coordenadas espaciais do pixel na imagem bidimensional.

- γ : Parâmetro de correção gama que controla o grau de não linearidade da transformação.

No entanto, neste trabalho, como os valores da imagem podem ultrapassar o intervalo $[0,1]$ após a amplificação da exposição, optou-se por uma versão normalizada da função gama, onde cada pixel é dividido pelo valor máximo da imagem antes da aplicação da potência:

$$I'(x, y) = \left(\frac{I(x, y)}{\max(I)} \right)^{\frac{1}{\gamma}} \quad (4.4)$$

Onde,

- $\max(I)$: Valor máximo de intensidade possível (ou observado) na imagem.

Esta abordagem assegura que:

- O valor máximo da imagem seja mapeado para 1.0 (branco máximo).
- A relação de proporcionalidade entre todos os tons seja preservada.
- A função seja aplicada de forma estável, independentemente da escala absoluta dos dados lineares de entrada.

4.1.9 Remoção de Névoa

Após a aplicação da correção gama, observou-se que a imagem resultante apresentava uma aparência perceptualmente "lavada", com baixo contraste local e perda de definição em regiões de baixa intensidade. Esse efeito é comum em imagens noturnas realçadas, especialmente quando a correção de exposição e gama ampliam não apenas os detalhes visuais, mas também os resíduos de luz difusa e ruído de fundo. Para mitigar esse efeito e restaurar a percepção de profundidade e contraste atmosférico da cena, foi incorporada à pipeline uma etapa adicional de remoção de névoa (dehazing).

O método utilizado baseia-se na técnica proposta por (HE; SUN; TANG, 2011), conhecida como Dark Channel Prior (DCP), originalmente concebida para remover neblina em imagens diurnas, mas aqui adaptada para recuperar contraste em cenas

noturnas amplificadas. A suposição fundamental do DCP é que, em áreas não afetadas pela neblina (ou névoa luminosa, neste caso), ao menos um dos canais de cor apresenta baixa intensidade. Isso permite estimar o canal escuro da imagem e inferir a presença de dispersão luminosa ou iluminação difusa excessiva.

A aplicação do algoritmo foi estruturada nas seguintes etapas:

- Cálculo do canal escuro: Obtém-se, para cada pixel, o valor mínimo entre os canais RGB dentro de uma vizinhança $\Omega(x)$, através da equação:

$$D(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{R, G, B\}} I^c(y) \right) \quad (4.5)$$

Onde,

- $D(x)$: Valor do canal escuro no pixel x , definido como o menor valor de intensidade entre os canais RGB dentro da vizinhança $\Omega(x)$.
- $\Omega(x)$: Janela local (vizinhança) centrada no pixel x , utilizada para calcular estatísticas locais, como o canal escuro.
- $I^c(y)$: Intensidade do canal de cor $c \in \{R, G, B\}$ no pixel localizado em y .
- Estimativa da luz atmosférica: Selecionam-se os pixels com maior intensidade no canal escuro para estimar a luz ambiente dominante A , que representa o véu de brilho presente na imagem.
- Estimativa da transmissão: A fração de luz que não foi espalhada é estimada por:

$$t(x) = 1 - \omega \cdot \min_{y \in \Omega(x)} \left(\min_{c \in \{R, G, B\}} \frac{I^c(y)}{A^c} \right) \quad (4.6)$$

Onde,

- $t(x)$: Função de transmissão no pixel x , que representa a fração de luz que não foi dispersada antes de atingir o sensor.
- ω : Parâmetro de controle do método ($0 < \omega < 1$), responsável por ajustar a intensidade da remoção de névoa.
- A^c : Componente da luz atmosférica correspondente ao canal de cor c .

- Recuperação da radiância da cena: Com a transmissão refinada $t(x)$ e a luz atmosférica A reconstrói-se a imagem:

$$J(x) = \left(\frac{I(x) - A}{\max(t(x), t_0)} \right) + A \quad (4.7)$$

Onde,

- $J(x)$: Radiância da cena recuperada no pixel x , correspondente à imagem final após a remoção da névoa.
- A : Componente da luz atmosférica correspondente ao canal de cor.
- t_0 : Valor mínimo imposto à transmissão para evitar instabilidades numéricas e amplificação excessiva de ruído em regiões muito escuras.

A aplicação do dehazing resultou em imagens com maior profundidade perceptual, melhor contraste local e recuperação eficaz de textura em áreas lavadas, contribuindo significativamente para a qualidade visual final das imagens processadas.

4.1.10 Equalização de Contraste Local – CLAHE

Mesmo após a correção gama, imagens provenientes de cenários de baixa luminosidade frequentemente exibem contraste local deficiente, com histogramas de intensidade concentrados em faixas estreitas. Essa compressão tonal dificulta a distinção perceptiva de bordas, texturas e microcontrastes, elementos essenciais para a interpretação visual e para a performance de algoritmos de visão computacional (ZUIDERVELD, 1994). Para mitigar este problema sem introduzir artefatos de realce excessivo ou amplificação de ruído, aplica-se a técnica CLAHE (*Contrast Limited Adaptive Histogram Equalization*).

Diferentemente da equalização de histograma global (AHE), que opera na imagem inteira e pode causar superamplificação do ruído em (PIZER et al., 1987). O algoritmo divide a imagem em uma grade de blocos retangulares (tiles). Para cada bloco, um histograma local é calculado, equalizado e, crucialmente, limitado (*clipped*). Este *clipping* prévio redistribui os pixels excedentes de um determinado nível de intensidade ao longo de todo o histograma, impedindo que regiões de baixa variância (como

céus ou sombras uniformes) sejam artificialmente *over-sharpened* e que o ruído seja realçado. Os histogramas equalizados de cada bloco são então combinados utilizando interpolação bilinear nos limites dos blocos, garantindo uma transição suave e evitando descontinuidades visíveis (*blocking artifacts*).

Para realizar o realce sem distorcer a cromaticidade da cena, o CLAHE é aplicado exclusivamente no canal de luminância. A imagem é, portanto, convertida do espaço RGB para o espaço de cor LAB, que desacopla de forma perceptual a informação de brilho (canal L^*) dos componentes cromáticos (a^* e b^*) (GONZALEZ; WOODS, 2018). Esta separação é matematicamente mais robusta para operações de realce de brilho do que a simples decomposição em HSV/HSL, pois o LAB é projetado para ser perceptualmente uniforme.

O fluxo da operação é o seguinte:

- Conversão de RGB para LAB: A imagem RGB (após gamma) é convertida para o espaço LAB.
- Extração e Normalização do Canal L^* : O canal de luminância (L^*) é extraído e normalizado para a faixa dinâmica esperada pela implementação (tipicamente 0-255 para inteiros de 8 bits).
- Aplicação do CLAHE: A função `cv2.createCLAHE` é aplicada ao canal L^* normalizado.
- Recomposição e Conversão: O canal L^* realçado é re combinado com os canais a^* e b^* originais, e a imagem é reconvertida para o espaço RGB.

O resultado final é uma imagem com contraste local significativamente aprimorado, preservando as cores originais e evitando realce artificial de ruído. Essa etapa reforça contornos e estruturas finas, contribuindo para a legibilidade da imagem em contextos técnicos e perceptuais.

4.1.11 Ajuste de saturação

A manipulação da saturação foi conduzida no espaço de cor HSV (*Hue, Saturation, Value*), uma representação cilíndrica que desacopla intuitivamente os componentes de matiz (*Hue*), saturação (*Saturation*) e brilho (*value*) (SMITH, 1978). Esta separação permite modificar a intensidade da cor sem alterar o tom fundamental ou a luminosidade percebida da imagem, uma operação que seria não trivial e potencialmente destrutiva no espaço RGB (GONZALEZ; WOODS, 2018).

O procedimento adotado segue três passos sequenciais:

- Conversão do Espaço de Cor: A imagem resultante do CLAHE (em RGB) é convertida para o espaço HSV utilizando a transformação padrão do OpenCV: `cv2.COLOR_RGB2HSV`.
- Modulação do Canal de Saturação (S): O canal S é escalado por um fator multiplicativo β , sendo o resultado limitado (clipped) para permanecer dentro da faixa válida [0, 255].
- Reconversão para RGB: Os canais são recombinaados e a imagem é convertida de volta ao espaço RGB para visualização e armazenamento.

É importante ressaltar que este ajuste opera de forma global e uniforme. Em cenários onde a saturação original varia significativamente entre diferentes regiões da imagem (ex: um objeto colorido contra um fundo neutro), uma abordagem adaptativa ou seletiva poderia ser considerada. Contudo, para o escopo deste trabalho e considerando a natureza geralmente homogênea da perda de saturação induzida pelo pipeline anterior, o ajuste global mostrou-se adequado e eficiente.

Esta etapa finaliza o pipeline de processamento, entregando uma imagem cujo contraste local, nitidez estrutural e vivacidade cromática foram otimizados para uma interpretação visual clara e perceptualmente balanceada.

5

DESENVOLVIMENTO

Esta capítulo apresenta, de forma descritiva e visual, o comportamento da pipeline de processamento de imagens implementada neste trabalho. Cada subetapa descrita na metodologia (Capítulo 4) será agora apresentada com sua respectiva função prática e a imagem resultante gerada após sua execução.

5.1 Estrutura da Pipeline Implementada

O pipeline de processamento de imagem noturna foi desenvolvida utilizando a linguagem de programação Python, devido à sua ampla adoção em tarefas de processamento de imagem e ao suporte robusto de bibliotecas especializadas. O projeto foi estruturado de forma modular, com cada bloco funcional implementado como uma função independente, permitindo o processamento sequencial e o salvamento das imagens após cada etapa.

O código principal do pipeline segue uma estrutura linear, onde a imagem é lida no formato *RAW* e processada passo a passo até a obtenção da imagem final melhorada. Cada função recebe como entrada uma imagem no formato *NumPy array*, realiza o processamento, e retorna a imagem transformada, que é imediatamente salva em disco no diretório *imagens_saida/*, para posterior análise visual e avaliação objetiva.

5.1.1 Organização do código

A organização geral do projeto segue a estrutura apresentada na Figura 2, a qual foi concebida para aprimorar a legibilidade do código e garantir o isolamento entre as etapas do pipeline. Dessa forma, modificações realizadas em um módulo não acarretam impactos nos demais. A pasta *utils* reúne os arquivos de código-fonte responsáveis pelo suporte e execução das etapas de processamento do pipeline.

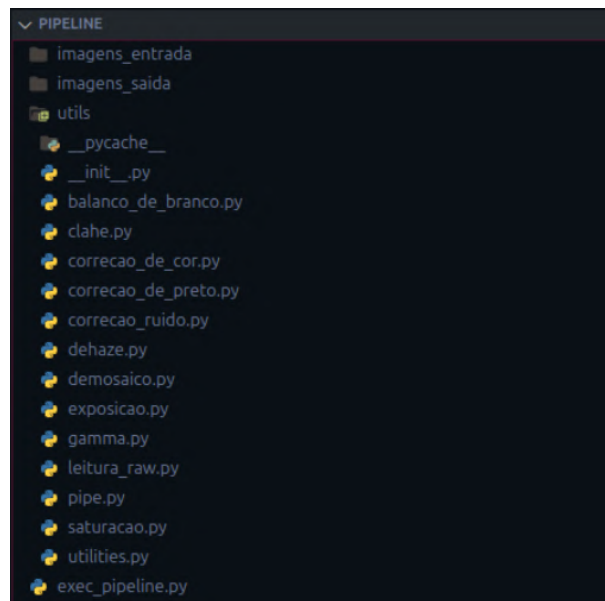


Figura 2 – Estrutura de Diretórios de projeto.

Fonte: Autoria Própria (2025).

5.2 Ambientes e Ferramentas Utilizadas

O desenvolvimento e execução do pipeline de processamento de imagens noturnas foram realizados com base em um conjunto de ferramentas e bibliotecas amplamente utilizadas na área de processamento de imagem e visão computacional. Esta seção descreve os principais recursos utilizados durante o projeto.

5.2.1 Ambiente de Desenvolvimento

O desenvolvimento do projeto foi realizado majoritariamente no sistema operacional Linux Ubuntu 22.04 LTS, escolhido por sua estabilidade, segurança e ampla compatibilidade com bibliotecas científicas em Python. A implementação e os testes do código foram conduzidos na IDE Visual Studio Code (VSCode), que oferece recursos como terminal integrado e depuração interativa, otimizando o fluxo de desenvolvimento. Adicionalmente, o sistema operacional Windows 11 foi utilizado para a execução de parte das métricas de avaliação de imagem no software MATLAB R2020, compatível com funcionalidades específicas exigidas em determinadas etapas do projeto.

A máquina utilizada para o desenvolvimento conta com um processador AMD Ryzen 7 3800X (8 núcleos e 16 threads), 16 GB de memória RAM, uma unidade de processamento gráfico (GPU) dedicada Nvidia GeForce GTX 1660Ti e um drive de estado sólido (SSD) com capacidade de 240GB, garantindo desempenho adequado para as demandas computacionais envolvidas.

A versão do *Python* utilizada foi a 3.8, instalada em um ambiente virtual (*venv*), com o objetivo de isolar as dependências do projeto, promovendo maior controle do ambiente de execução e facilitando a reprodutibilidade dos experimentos. Para isso, se fez uso na implementação do pipeline um conjunto de bibliotecas especializadas, cada uma contribuindo de forma complementar para o processamento das imagens.

A biblioteca *rawpy* (versão 0.21.0) foi empregada para a leitura e conversão dos arquivos *RAW*, servindo como ponto de entrada para o fluxo de processamento. As bibliotecas *numpy* (1.24.4) e *OpenCV* (4.12.0.88) foram utilizadas, respectivamente, para manipulação vetorial/matricial e para operações gerais de processamento e transformação de imagens. Para avaliação de qualidade, utiliza-se a biblioteca *imaquality* (1.2.7), responsável pelo cálculo da métrica BRISQUE, enquanto a *scikit-image* (0.18.0) oferece funções adicionais para filtragem e aprimoramento das imagens ao longo do pipeline, bem como ilustrado na tabela 3 temos um resumo das bibliotecas utilizadas.

Biblioteca	Versão utilizada	Finalidade principal
rawpy	0.21.0	Leitura e conversão de arquivos RAW
numpy	1.24.4	Manipulação vetorial e matricial
OpenCV	4.12.0.88	Manipulação de imagens
imaquality	1.2.7	Cálculo da métrica BRISQUE
skimage	0.18.0	Filtragem de ruído

Tabela 3 – Bibliotecas utilizadas e suas finalidades.

Fonte: Autoria Própria (2025).

Além das bibliotecas citadas, foram utilizados arquivos .ARW (formato RAW da Sony) como entrada, obtidos de repositórios públicos de imagens RAW, do repositório *See in the Dark (SID)* (CHEN et al., 2018).

5.3 Resultados Intermediários do Processamento

Esta seção apresenta os resultados visuais obtidos após a aplicação de cada etapa da pipeline de processamento sobre as imagens RAW. O objetivo é demonstrar de forma progressiva como cada bloco influencia na reconstrução e aprimoramento da imagem, desde sua forma bruta até uma versão visualmente aprimorada e tecnicamente adequada para análise.

As imagens exibidas foram salvas automaticamente durante a execução da pipeline e representam as saídas parciais de cada função implementada. Essa abordagem facilita a avaliação individual de cada transformação aplicada, permitindo ajustes finos no fluxo e avaliações qualitativas.

5.3.1 Leitura da Imagem RAW e Correção de Nível de Preto

A etapa inicial do processo consistiu na leitura do arquivo RAW como podemos ver na Figura 3(a) por meio da biblioteca rawpy. Nessa fase, foram extraídos tanto a matriz CFA (*Color Filter Array*) quanto os metadados associados à captura, como níveis de

preto, *white level* e parâmetros específicos do sensor. Os dados foram convertidos para o tipo *double*, de modo a preservar a linearidade radiométrica e assegurar maior precisão nas operações subsequentes. A imagem bruta resultante foi armazenada para fins de documentação do fluxo e para permitir a análise comparativa entre as etapas da pipeline e em seguida a correção de nível de preto como observado na Figura 3(b) foi realizada com o objetivo de eliminar o *offset* eletrônico inerente ao sensor, responsável por valores residuais mesmo na ausência de iluminação. O valor do *black level* foi obtido diretamente dos metadados do arquivo e utilizado na normalização dos dados conforme:

$$I_{\text{norm}}(x, y) = \frac{(X - \Delta)}{V_{\text{max}} - \Delta} \times V_{\text{max}} \quad (5.1)$$

Onde:

- x representa o valor original do pixel;
- Δ é o valor do *black level* (*offset* eletrônico);
- V_{max} é o valor máximo da faixa de codificação do sensor (por exemplo, 4095 para 12 bits).



(a) Leitura RAW.



(b) Correção de níveis de preto.

Figura 3 – Imagens processadas.

Fonte: Autoria Própria (2025).

5.3.2 Demosaico e Balanço de branco

Posteriormente, na Figura 4(a) foi realizada a interpolação cromática da matriz CFA, convertendo-se os valores monocromáticos de cada pixel em uma imagem RGB completa. Para essa finalidade, empregou-se o algoritmo AHD (*Adaptive Homogeneity-Directed*), disponibilizado pela função `postprocess()` da biblioteca *rawpy*. O método foi selecionado por sua capacidade de preservar bordas e reduzir artefatos cromáticos em regiões de alto contraste. A imagem resultante Figura 4(b) foi convertida para o tipo *double* e registrada como etapa intermediária do pipeline, e em seguida balanço de branco foi aplicado com base no método *Gray World*, que parte da premissa de que a média global dos canais R, G e B em uma cena iluminada de forma neutra tende a apresentar valores similares. Inicialmente, foi calculada a média de cada canal, sendo então determinados os fatores de ganho.

Para evitar amplificação indevida de ruído, os valores de ganho foram restringidos ao intervalo de 0,5 a 3,0. Os ganhos foram aplicados multiplicativamente sobre cada canal, resultando na correção da dominante cromática presente na imagem. O resultado foi armazenado, possibilitando avaliação visual e validação da etapa.



(a) Demosaico.



(b) Balanço de branco.

Figura 4 – Imagens processado.

Fonte: Autoria Própria (2025).

5.3.3 Correção de exposição e Correção de Cor

A correção de exposição na Figura 5(a) foi aplicada com o objetivo de ajustar globalmente a luminância da imagem após o balanço de branco. Nessa etapa, cada pixel da imagem foi multiplicado por um fator escalar.

$$I'(x, y) = g \cdot I(x, y), \quad (5.2)$$

onde $g=0.8$, corresponde ao valor que apresentou melhor adequação nos experimentos conduzidos. Esse parâmetro mostrou-se eficaz nos casos de teste, pois não promove a elevação excessiva dos valores de cada pixel, evitando a superexposição da imagem. Em termos práticos, considerando a representação típica de imagens de 8 bits, escala de que vai 0 a 255, o uso de $g=0,8$ resulta em um aumento moderado da intensidade dos pixels, preservando detalhes nas regiões claras e evitando a saturação de valores próximos a 255. Além disso, essa escolha não contribui significativamente para a amplificação do ruído, especialmente em regiões de baixa iluminação. Após a multiplicação, os valores resultantes foram normalizados para o intervalo $[0,1]$, utilizando divisão por 255 quando necessário, e posteriormente truncados (*clipping*) para evitar saturação e preservar consistência radiométrica.

A etapa de correção cromática na Figura 5(b) foi executada utilizando uma matriz de correção de cor (CCM) derivada da propriedade `color_matrix` presente nos metadados do arquivo RAW. Essa matriz representa a transformação linear necessária para converter o espaço de cor da câmera, específico do sensor, para um espaço de cor padrão mais próximo do comportamento perceptivo humano. Matematicamente, a transformação foi aplicada pixel a pixel da seguinte forma:

$$C' = M_{CCM} \cdot C, \quad (5.3)$$

onde,

- $C = [R, G, B]^T$ representa o vetor cromático original do pixel.
- C' é o vetor corrigido
- M_{CCM} é a matriz 3×3 obtida via metadados da câmera.

$$\mathbf{M}_{\text{CCM}} = \begin{bmatrix} 0.5838 & -0.143 & -0.0246 \\ -0.3497 & 1.1477 & 0.2297 \\ -0.0748 & 0.1885 & 0.5778 \end{bmatrix} \quad (5.4)$$

E para viabilizar o cálculo matricial, a imagem foi reorganizada em um vetor de dimensões $(H \times W) \times 3$, possibilitando a aplicação vetorial da transformação. Em seguida, os dados foram reestruturados de volta para o formato matricial original (altura \times largura \times 3).



(a) Correção de exposição.



(b) Correção de cor.

Figura 5 – Imagens processadas.

Fonte: Autoria Própria (2025).

5.3.4 Redução de ruído e Correção de Gama

A etapa de redução de ruído Figura 6(a) foi implementada com o objetivo de mitigar artefatos de alta frequência decorrentes da captura em condições de baixa luminosidade. Para isso, empregou-se o filtro bilateral, amplamente utilizado em pipelines de processamento de imagem devido à sua capacidade de suavizar regiões homogêneas enquanto mantém bordas preservadas. No código, o método `denoise_bilateral` (da biblioteca *scikit-image*) foi aplicado no modo multicanal, o que garante que a filtragem considere as correlações cromáticas entre os canais RGB. A imagem resultante foi

então salva para documentar o impacto do processo de *denoising* antes da aplicação dos ajustes tonais subsequentes.

A correção gama Figura 6(b) foi aplicada com o objetivo de ajustar a resposta tonal da imagem para aproximá-la da sensibilidade não linear do sistema visual humano. Em sensores digitais, os dados são mantidos em domínio linear; entretanto, a percepção humana é mais sensível a variações tonais em regiões escuras do que em áreas iluminadas. A aplicação da função gama compensa essa diferença perceptual.

$$I'(x, y) = \left(\frac{I(x, y)}{\max(I)} \right)^{\frac{1}{\gamma}} \quad (5.5)$$

Onde,

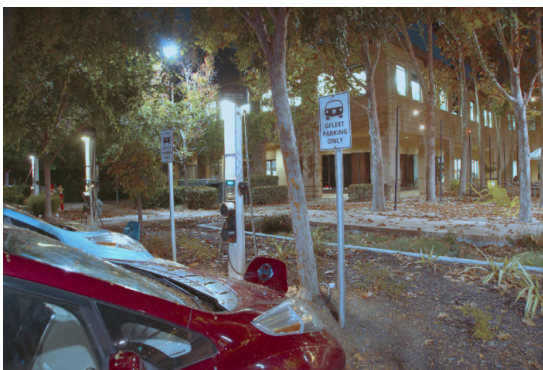
- $I(x,y)$: Representa a intensidade luminosa linear do pixel localizado na posição espacial (x,y) da imagem original, diretamente proporcional à quantidade de luz capturada pelo sensor.
- $I'(x,y)$:Corresponde à intensidade do pixel após a aplicação da correção gama, já ajustada para a percepção visual humana e adequada à exibição em dispositivos digitais.
- x,y : Coordenadas espaciais do pixel na imagem bidimensional.
- γ : Parâmetro de correção gama que controla o grau de não linearidade da transformação.

Adotou-se $\gamma = 2,2$ como parâmetro final do processo de correção de gama. Esse valor foi selecionado após a realização de testes preliminares com $\gamma = 1,0, 1,5, 2,0$ e $2,5$, empregados nas etapas iniciais de validação, mas que não apresentaram desempenho satisfatório para os objetivos do estudo. A escolha de $\gamma = 2,2$ também se justifica por sua correspondência direta com o padrão sRGB, amplamente utilizado em sistemas de captura, processamento e exibição de imagens (IMATEST, n.d.). Esse padrão assume uma resposta não linear aproximada de $\gamma = 2,2$, refletindo tanto características fisiológicas da percepção humana de luminância quanto limitações históricas e técnicas de dispositivos de exibição. Dessa forma, a adoção desse valor garante maior compatibilidade com fluxos de trabalho visuais contemporâneos e assegura que os resultados

obtidos sejam coerentes com práticas consolidadas na área. O procedimento envolve duas etapas fundamentais:

- Normalização: A imagem é inicialmente dividida pelo valor máximo presente no conjunto de pixels, garantindo que a entrada da função exponencial pertença ao intervalo $[0,1]$.
- Aplicação da transformação não linear: Cada pixel é elevado à potência $1/\gamma$, o que resulta na compressão das altas intensidades e expansão das baixas intensidades.

Esse mapeamento tonal contribui para a melhoria da percepção visual do conteúdo da cena, especialmente em regiões escuras típicas de cenas noturnas. Após a transformação, o resultado foi salvo como etapa intermediária, permitindo avaliar de forma objetiva e subjetiva o impacto do mapeamento não linear antes dos ajustes finais da pipeline.



(a) Redução de ruído.



(b) Correção de gama.

Figura 6 – Imagens processadas.

Fonte: Autoria Própria (2025).

5.3.5 Remoção de Névoa, CLAHE e Saturação

A etapa de remoção de névoa (*dehazing*) Figura 7 implementada baseia-se no *Dark Channel Prior (DCP)* proposto por (HE; SUN; TANG, 2011) e em seguida emprega

um refinamento de transmissão via *guided filter*. O objetivo é estimar e remover a componente atmosférica difusa que reduz o contraste e “lavado” da cena após as etapas de realce, recuperando a radiância direta dos objetos.

Parâmetros e observações práticas

- `window_size` (15): controla a escala local do *dark channel*; janelas maiores tornam a estimativa mais robusta a ruído, porém menos sensível a detalhes finos.
- `omega` (0.60): determina quanto do véu atmosférico será removido; valores menores deixam um pouco de véu para evitar super-contraste.
- `guided_filter_radius` (60) e **eps (1e-3)**: definem o raio e a regularização do filtro guiado — maiores raios proporcionam suavização mais ampla; `eps` controla o grau de suavização da regressão local.
- A conversão para `uint8` e a construção do guia em tons de cinza (`gray`) visam compatibilidade com funções *OpenCV*, mas recomenda-se preservar cópias em ponto flutuante para análise quantitativa.

Onde o método método DCP tende a recuperar contraste e saturação, mas pode amplificar ruído ou gerar artefatos em áreas homogêneas se os parâmetros não forem ajustados.



Figura 7 – Remoção de névoa.

Fonte: Autoria Própria (2025).

O uso do *guided filter* minimiza halos, porém o procedimento depende fortemente da estimativa inicial de A e t . Em seguida Figura 8 a equalização de histograma adaptativa com limitação de contraste (CLAHE) é empregada para reforçar o contraste local sem exagerar o ruído, sendo adequada após etapas que aumentam a exposição e a compressão tonal.

Procedimento aplicado

- **Conversão para LAB:** a imagem é convertida do espaço RGB para LAB (`cv2.COLOR_RGB2LAB`) para separar informação de luminosidade (canal L) da informação cromática (canais a e b). Trabalhar sobre L evita alterações indesejadas na matiz das cores.
- **Aplicação do CLAHE no canal L :** Na aplicação do CLAHE ao canal L , adotou-se $clipLimit = 0,2$ e $tileGridSize = (2,2)$. O valor reduzido de $clipLimit$ foi selecionado com o objetivo de limitar a amplificação do histograma em cada *tile*, minimizando a intensificação de ruído, aspecto particularmente relevante em imagens capturadas sob baixa luminosidade. Por sua vez, o $tileGridSize = (2,2)$ foi definido de modo a proporcionar uma equalização de granularidade mais ampla, evitando a produção de *microcontraste* artificial e assegurando um realce mais suave e perceptualmente consistente. Testes preliminares com configurações como (4,4) e (8,8) evidenciaram excesso de detalhamento local e artefatos indesejados, reforçando a adequação da configuração adotada. Dessa forma, os parâmetros escolhidos mostraram-se eficazes para preservar a naturalidade da luminância ao mesmo tempo em que promovem uma melhoria local de contraste.
- **Reconstrução e conversão para RGB:** após a equalização do canal L , os canais são reunidos e reconvertidos para RGB. Os valores são normalizados para ponto flutuante no intervalo $[0, 1]$, garantindo continuidade no pipeline.



Figura 8 – CLAHE.

Fonte: Autoria Própria (2025).

E por fim a etapa de ajuste de saturação Figura 9 visa restabelecer a vivacidade cromática alterada pelas etapas anteriores (exposição, *dehaze*, CLAHE). O procedimento aplicado segue uma abordagem direta em espaço HSV:

- **Conversão e normalização:** a imagem é convertida para um formato compatível (`convert_image / BGR → HSV` conforme uso do OpenCV). Em seguida, utiliza-se `cv2.cvtColor` para converter BGR→HSV.
- **Escalonamento dos canais S e V:**
 - O canal de saturação *S* é multiplicado por `saturation_scale` e normalizado com `cv2.normalize` para o intervalo `[0, 255]`, restaurando intensidade das cores.
 - O canal de valor (“vividness”) *V* é multiplicado por `vividness_scale` e normalizado, aumentando saturação percebida e brilho de forma controlada.
- **Reconversão e salvamento:** após normalização, a imagem é convertida de volta para BGR (`cv2.COLOR_HSV2BGR`) e salva.



Figura 9 – Saturação.

Fonte: Autoria Própria (2025).

5.4 Considerações Técnicas

Durante o desenvolvimento e execução da pipeline de processamento de imagens noturnas, diversas decisões técnicas foram tomadas visando melhorar a robustez, qualidade dos resultados e compatibilidade com imagens RAW (.ARW) de diferentes câmeras. Esta seção destaca os principais desafios enfrentados, ajustes realizados e observações relevantes que impactaram diretamente na construção do fluxo de processamento.

5.4.1 Formato RAW e limitações dos metadados

A leitura dos arquivos no formato .ARW (Sony RAW) foi realizada por meio da biblioteca *rawpy*, que fornece acesso direto aos dados brutos do sensor da câmera. Essa biblioteca permite a aplicação de configurações personalizadas durante o processo de interpolação cromática, incluindo o uso de uma matriz de correção de cor (*Color Correction Matrix – CCM*). A obtenção e aplicação da CCM também foram viabilizadas com o auxílio do *rawpy*, visando um ajuste mais preciso das cores. Contudo, observou-se a necessidade de um trabalho conjunto com a função de balanço de branco, de modo a alcançar resultados visuais mais equilibrados e fielmente representativos da cena original.

5.4.2 Ajustes manuais nos ganhos de exposição e balanço de branco

As funções de correção de exposição e de balanço de branco automático (*Auto White Balance – AWB*) foram desenvolvidas de forma parametrizável, permitindo ajustes finos conforme as características da imagem. Em casos de imagens severamente subexpostas, a aplicação de ganhos elevados resultam em efeitos indesejados, enquanto ganhos insuficientes não eram capazes de revelar detalhes relevantes. Para mitigar esses efeitos, foram testados diversos fatores de amplificação, variando entre 1,0 e 2,0, além de ajustes no cálculo do ganho médio entre os canais de cor. Essa abordagem permitiu minimizar o surgimento de tonalidades artificiais e obter um equilíbrio cromático mais consistente.

5.4.3 Correção Gama e percepção visual

O uso evidenciou ainda que a aplicação da correção gama com $\gamma = 2,2$ produz um clareamento global da imagem, frequentemente associado a uma aparência lavada. Esse comportamento, já antecipado anteriormente, justificou a inclusão do módulo de *Dehazing*, que demonstrou capacidade de restaurar o contraste atmosférico e reforçar os níveis de preto, compensando adequadamente o efeito colateral da correção gama.

5.4.4 CLAHE

A análise final dos experimentos evidenciou que ajustes mais elevados de *clipLimit* ou *tileGridSize* intensificam a granulação, especialmente em regiões homogêneas. Assim, os resultados corroboram a escolha da configuração mais moderada adotada no trabalho (*clipLimit = 0,2; tileGridSize = (2,2)*).

5.4.5 Saturação e estética da imagem

A etapa de ajuste de saturação mostrou-se relevante para restaurar a naturalidade das cores após as transformações realizadas durante o processamento da imagem. O incremento na saturação foi aplicado de forma controlada no espaço de cor HSV,

com o objetivo de evitar exageros que poderiam introduzir distorções perceptíveis, especialmente em tons de pele ou elementos naturais. Embora não constitua uma etapa essencial do ponto de vista técnico, esse ajuste contribui significativamente para a qualidade estética da imagem final.

5.4.6 Modularização e salvamento por etapas

Uma decisão técnica fundamental foi a modularização completa do pipeline, com salvamento automático de imagens após cada etapa. Isso não apenas facilitou a análise qualitativa e identificação de problemas, como também permitiu que o código fosse ajustado em partes sem afetar o funcionamento geral do pipeline. Todas as imagens foram armazenadas no diretório *imagens_saida/*, com nomes numerados de forma incremental e padronizada

5.5 Preparação para Avaliação da Qualidade

A etapa final da pipeline compreende a avaliação da qualidade das imagens processadas por meio de métricas objetivas sem referência, considerando a ausência de uma imagem original perfeita. Essa avaliação é fundamental para validar, de forma quantitativa, os efeitos das transformações aplicadas sobre a imagem capturada em condições de baixa luminosidade.

5.5.1 Métricas de Qualidade Utilizadas

Neste trabalho, optou-se pela utilização de métricas sem referência, adequadas a contextos onde não há uma imagem ideal conhecida para comparação. As métricas adotadas conforme visto na metodologia proposta foram:

- **BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator)**: Analisa a qualidade da imagem com base em estatísticas naturais de cenas. Um valor mais

baixo indica melhor qualidade perceptual. BRISQUE é sensível a ruído, desfoque, compressão e outros artefatos comuns em imagens degradadas.

- **NIQE (*Naturalness Image Quality Evaluator*)**: Avalia o grau de naturalidade da imagem, também com base em estatísticas naturais, porém sem necessidade de treinamento prévio. NIQE estima a distância entre a imagem de entrada e um modelo estatístico de imagens de boa qualidade. Valores mais baixos indicam maior naturalidade e qualidade visual.

Essas métricas foram escolhidas por serem não supervisionadas, interpretáveis e amplamente utilizadas na literatura para avaliação de imagens realçadas ou capturadas em ambientes adversos.

5.5.2 Implementação da Avaliação

As imagens geradas pelo pipeline foram submetidas a um processo de avaliação estruturado:

- As métricas BRISQUE foram calculadas diretamente via Python para todas as imagens finais;
- As métricas NIQE foram obtidas no MATLAB, utilizando os mesmos arquivos exportados da saída pipeline;

Os resultados de ambas as métricas foram registrados e organizados e serão discutidos no Capítulo 6.

6

RESULTADOS E DISCUSSÕES

Este capítulo apresenta e analisa os resultados obtidos com a aplicação da pipeline de processamento, detalhada anteriormente, às imagens noturnas no formato RAW. As imagens de entrada, provenientes do *dataset See in the Dark (SID)* (CHEN et al., 2018), consistem em cenas externas capturadas com baixa luminosidade, tempo de exposição de 10 segundos e padrão de *Bayer* RGB. Os resultados são organizados para evidenciar o desempenho da pipeline em diferentes cenários e condições presentes no conjunto de dados.

As imagens originais utilizadas neste trabalho, como podem ser vistas no anexo A, são todas capturadas em ambientes com luminosidade baixa, mas sem um rigor técnico específico usado para a escolha destas as imagens em geral, são compostas por cenas de ambiente externo no período da noite e com uma diversidade nas composições de cenas.

O presente trabalho, tem como objetivo mensurar, de forma quantitativa, os impactos das transformações aplicadas em cada etapa do pipeline, com ênfase na recuperação de detalhes, na melhoria da exposição e na qualidade visual geral das imagens processadas.

Foram empregadas duas métricas objetivas de avaliação de qualidade de imagem sem referência: (i) **BRISQUE** e (ii) **NIQE**. Essas métricas são aplicadas às imagens resultantes da pipeline, permitindo a análise dos efeitos cumulativos das transformações implementadas. Destaca-se, nesse contexto, a importância do bloco de redução de ruído, cuja influência é investigada para verificar se sua inclusão contribui positivamente ou

negativamente para os escores obtidos pelas métricas mencionadas.

Na etapa subsequente, realiza-se uma análise comparativa entre os resultados fornecidos pelas duas métricas, com o intuito de avaliar a consistência entre elas e, conseqüentemente, validar a confiabilidade das avaliações quantitativas realizadas.

Por fim, a análise é complementada por uma avaliação visual qualitativa subjetiva, conduzida com o objetivo específico de identificar detalhes recuperados nas imagens, em consonância com a proposta de processamento de imagens em cenários de baixa luminosidade. Trata-se de uma apreciação subjetiva realizada pelo próprio autor, sem a participação de um grupo avaliador e sem a intenção de classificar ou ranquear os resultados. Nessa etapa, destacam-se casos para o *pipeline* demonstrar a capacidade de recuperar informações anteriormente imperceptíveis, reforçando a eficácia prática do método proposto.

6.1 Métricas BRISQUE e NIQE

A métrica BRISQUE (*Blind/Referenceless Image Spatial Quality Evaluator*) como previamente estabelecida foi utilizada para estimar a qualidade perceptiva das imagens processadas por sua vez a métrica NIQE (*Naturalness Image Quality Evaluator*) foi calculada utilizando MATLAB, devido à sua maior precisão na implementação. NIQE avalia a naturalidade da imagem em relação a um modelo estatístico ideal de imagens naturais. .

6.1.1 Pontuações

Observou-se que a inclusão do bloco de *denoise* não ocasionou incremento nas pontuações das métricas; em certas amostras, verificou-se uma leve piora dos valores após a aplicação do filtro, o que sugere que BRISQUE e NIQE pode interpretar a suavização introduzida pelo método de redução de ruído como perda de detalhes estruturais relevantes.

Em decorrência desses pontos, na Tabela 4 apresentam-se os resultados quan-

titativos obtidos com e sem a etapa de *denoise*, visando analisar comparativamente o impacto dessa etapa nas medidas de qualidade.

Imagens	BRISQUE com ruído		BRISQUE sem ruído		NIQE com ruído		NIQE sem ruído	
	Pontuação%		Pontuação%		Pontuação%		Pontuação%	
Quadra	43,62	56,38	40,12	59,88	4,97	50,28	4,19	58,09
Lago	42,53	57,47	34,10	65,90	3,95	60,55	3,95	61,23
Piscina	43,70	56,30	37,24	62,76	4,36	56,38	3,94	60,57
Placa	44,73	55,27	36,35	63,65	4,52	54,78	3,82	61,76
Rua	49,88	50,12	43,32	56,68	4,78	52,22	4,52	54,77
Vaso	47,98	52,02	37,88	62,12	4,28	57,18	4,22	57,78
Cadeira	29,86	70,14	22,83	77,17	3,67	63,28	3,25	67,54
Telefone	43,90	56,10	33,69	66,31	3,73	62,71	3,15	68,54
Carros	32,96	67,04	23,97	76,03	3,73	62,72	2,90	70,98
Banco	45,95	54,05	23,52	76,48	4,59	54,14	2,77	62,27

Tabela 4 – Resultados BRISQUE e NIQE com e sem bloco de ruído.

O impacto específico do bloco de redução de ruído no resultado quantitativo final após o experimento controlado comparando duas configurações do pipeline: (1) Configuração Completa, incluindo o bloco de *denoise* de filtragem bilateral, também podem ser observados na Figura 10 e Figura 11;

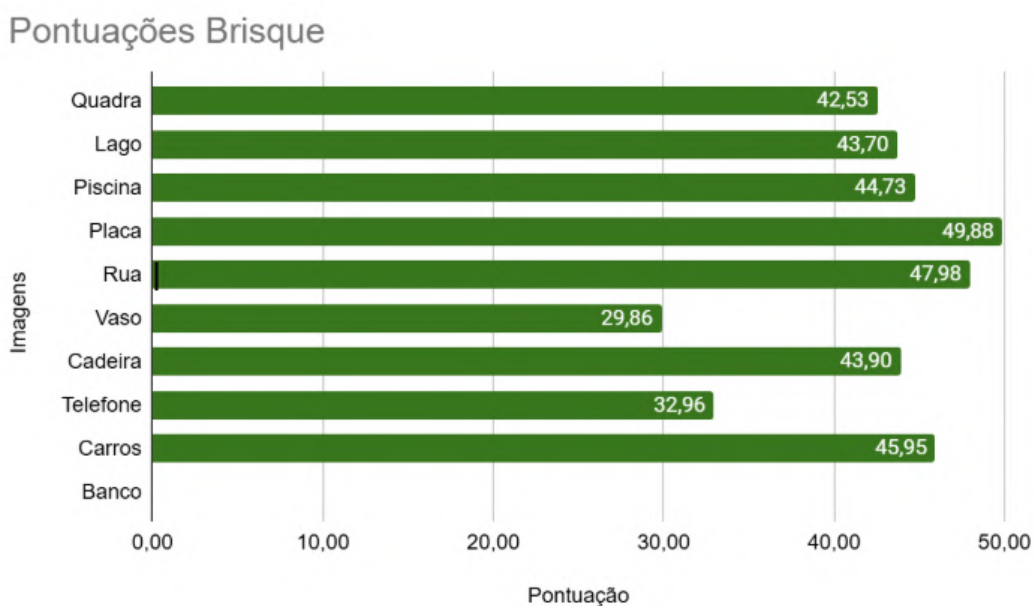


Figura 10 – BRISQUE com bloco de ruído.

Fonte: Autoria Própria (2025).

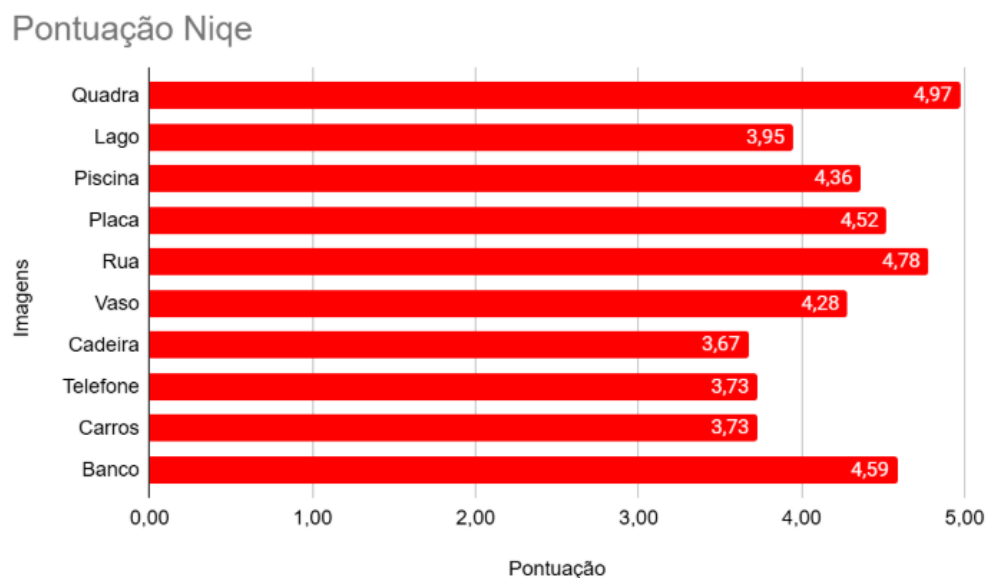


Figura 11 – NIQE com bloco de ruído.

Fonte: Aatoria Própria (2025).

e (2) Configuração sem *Denoising*, com a exclusão deste bloco. Pode ser notado em ambos BRISQUE e NIQE, valores observado nas Figura 12 e Figura 13.

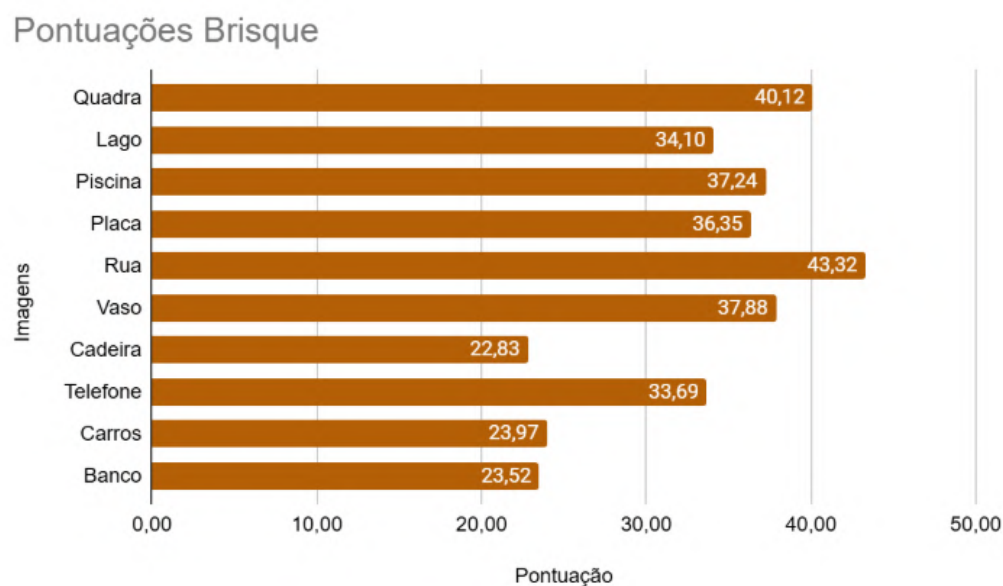


Figura 12 – BRISQUE sem bloco de ruído.

Fonte: Aatoria Própria (2025).

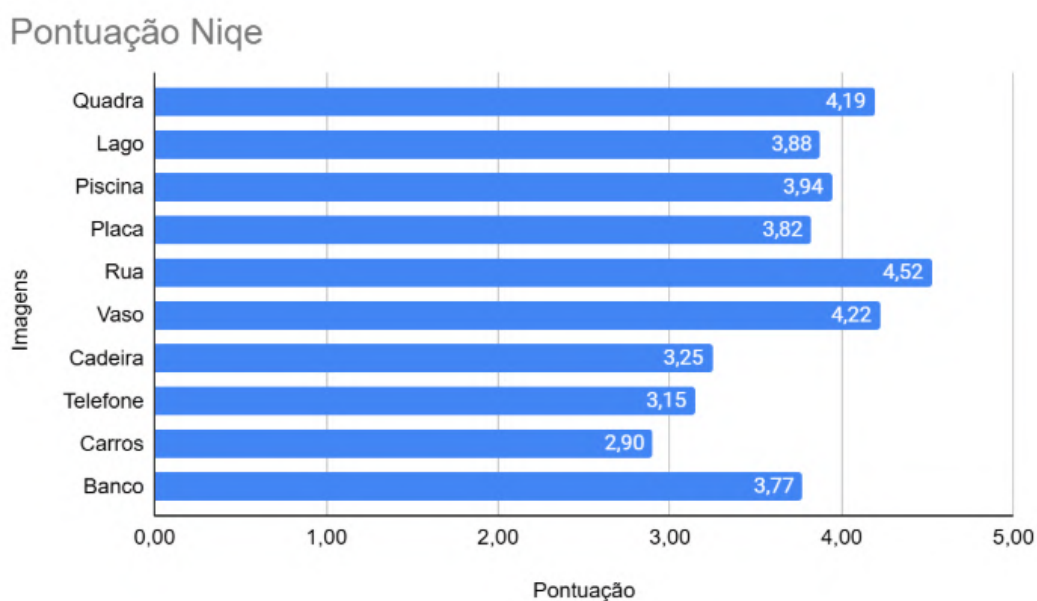


Figura 13 – NIQE sem bloco de ruído.

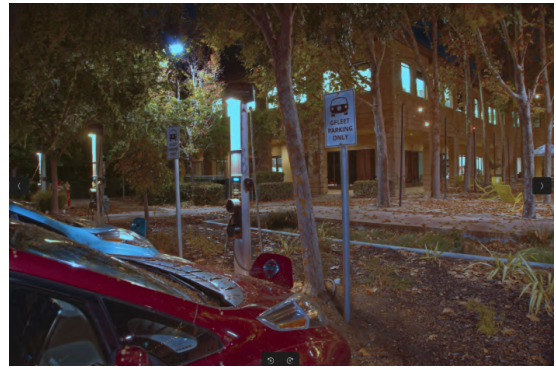
Fonte: Autoria Própria (2025).

para todo o conjunto de 10 imagens representativas de cenários de baixa luminosidade. Os resultados, apresentados na Tabela 4, revelam um padrão consistente de leve degradação quando o módulo de redução de ruído é empregado fazendo com que a pontuação sofra reduções.

Ao analisar os dois contextos — o pipeline completa e o pipeline sem o bloco de redução de ruído — observa-se que as imagens Figura 14(a) e Figura 14 (b) , "Cadeira" e "Carros" respectivamente apresentam desempenho consistente em ambas as combinações de métricas, sendo bem avaliadas em todos os casos. Isso indica que essas cenas se beneficiaram significativamente do processamento, exibindo características que são favoráveis tanto à modelagem estatística do NIQE quanto aos critérios perceptivos do BRISQUE.



(a) Cadeira - com denoise



(b) Carro - com denoise

Figura 14 – Imagens processadas.

Fonte: Autoria Própria (2025).

Imagens	BRISQUE		NIQE	
	Pontuação	%	Pontuação	%
Cadeira	22,83	77,17	3,25	67,54
Banco	23,52	76,48	–	–
Carros	23,97	76,30	2,90	70,98
Telefone	–	–	3,15	68,54

Tabela 5 – Comparação BRISQUE e NIQE.

Na Tabela 5 pode-se notar que há uma concordância parcial entre as métricas, com as imagens da Figura 14(a) e Figura 14(b) aparecendo em ambas as listas, ainda que em posições diferentes. O BRISQUE favorece a imagem da Figura 15(c) (Banco) como terceira melhor, enquanto o NIQE mantém a Figura 15(b) (Telefone) nessa posição.

Por outro lado, as diferenças nas posições ou nas imagens restantes como as da Figura 15(a), Figura 15(b) e Figura 15(c) revelam o comportamento distinto de cada métrica em relação a diferentes tipos de cena. Por exemplo, imagens com maior complexidade de fundo ou menor definição podem ser penalizadas de forma diferente por cada algoritmo.

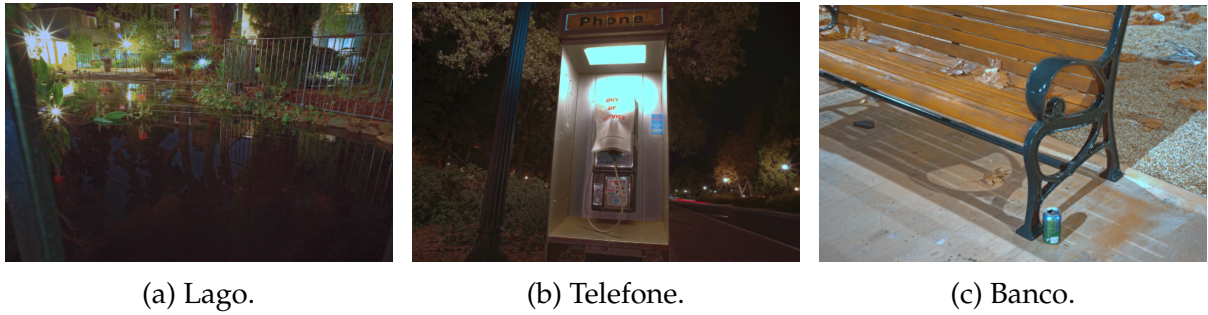


Figura 15 – Imagens processadas.

Fonte: Autoria Própria (2025).

Essa análise sugere que, embora as métricas não estejam perfeitamente alinhadas, há uma convergência geral nas imagens consideradas de melhor qualidade, o que fortalece a consistência dos resultados e valida o desempenho da pipeline desenvolvida.

Quanto ao impacto do bloco de redução de ruído, os resultados indicam uma leve perda quantitativa, conforme observado na Figura 16 e Figura 17 nas métricas avaliadas.

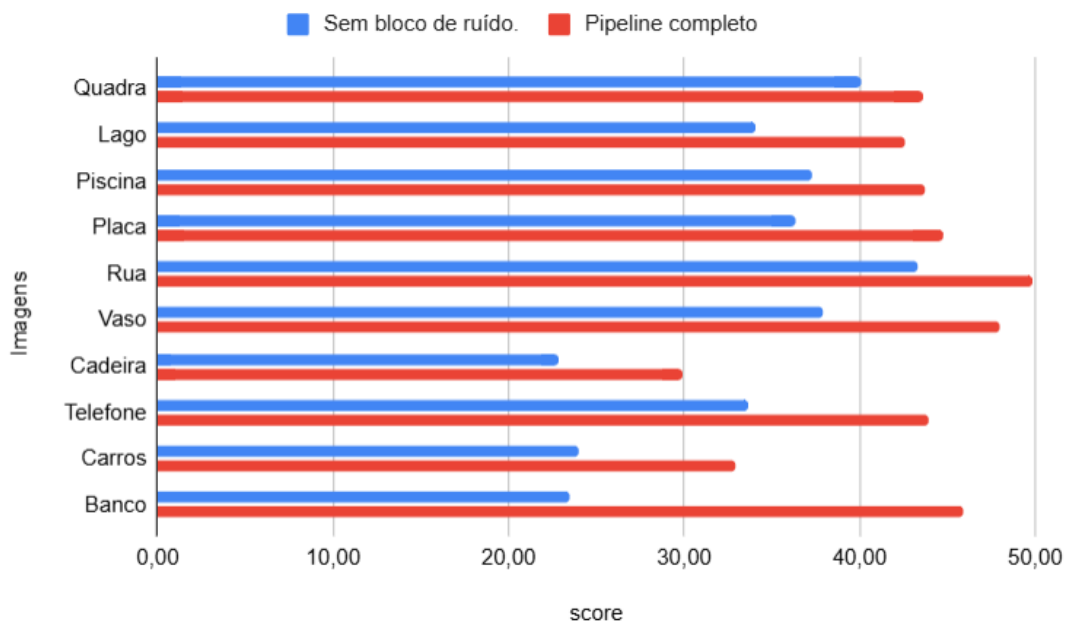


Figura 16 – BRISQUE - Com e sem denoise.

Fonte: Autoria Própria (2025).

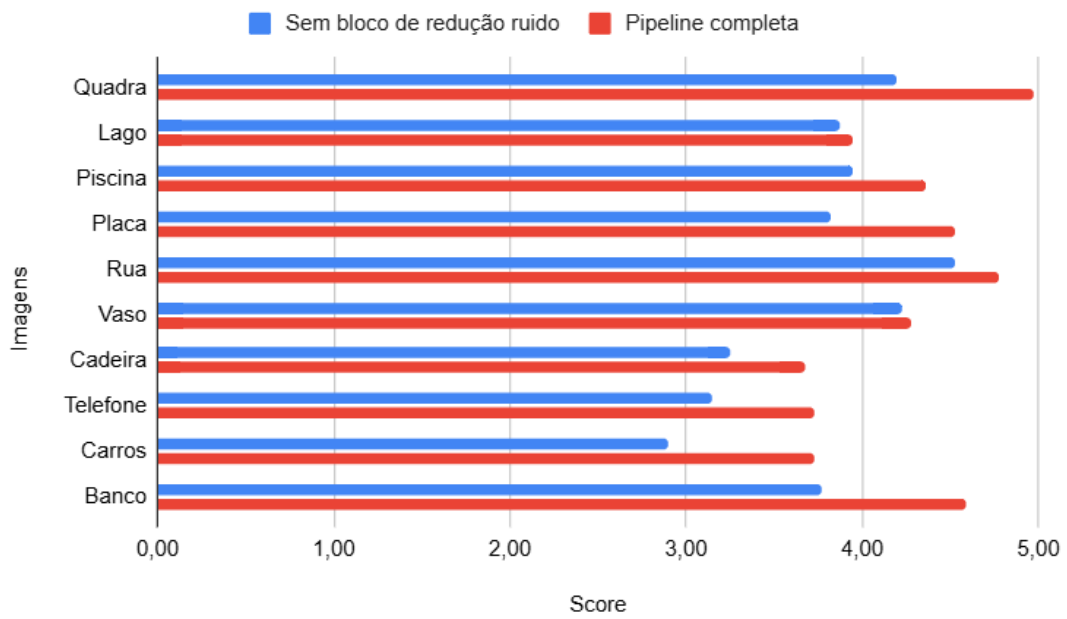


Figura 17 – NIQE- Com e sem denoise.

Fonte: Autoria Própria (2025).

Especificamente, o BRISQUE mostrou uma redução de 16,32%, enquanto o NIQE apresentou uma perda menor, de 8,70%. Apesar dessa degradação dos valores numéricos, não foi possível perceber uma deterioração significativa na qualidade visual das imagens como visto na sessão seguinte. Isso pode ser explicado pela importância do bloco de redução de ruído em cenários de baixa luminosidade, onde ele desempenha um papel crucial na melhoria da visibilidade das imagens.

Com intuito de comparar de forma objetiva os resultados das duas abordagens e mensurar o impacto percentual foi utilizado o cálculo de melhoria percentual, onde a variação percentual entre as configurações com e sem o módulo de *denoising* foi calculada utilizando a fórmula padrão de diferença relativa, conforme práticas estabelecidas em análise estatística comparativa (FIELD, 2009).

$$\Delta\% = \left(\frac{B - A}{B} \right) \times 100\%$$

Onde,

- A é Pontuação em porcentagem da primeira execução com a pipeline completa:

- **B** Pontuação em porcentagem da segunda execução sem o bloco de redução de ruído no pipeline

Com isso, podemos então observar na Tabela 6 o percentual de melhora obtido para cada uma das imagens, onde os dados demonstram que, para a maioria absoluta das imagens testadas, a exclusão do módulo de denoising resultou em uma melhoria significativa nas métricas. Em média, a pontuação BRISQUE melhorou em 16,32% e a NIQE em 8,70%. Esta tendência é particularmente pronunciada em cenas com texturas complexas e alto detalhamento, como "Quadra" em NIQE com 15,54% e "Banco" em BRISQUE e NIQE. Apenas na imagem "Lago" a métrica NIQE permaneceu com um valor baixo em comparação aos outros. Este fenômeno pode ser explicado pelas características operacionais do filtro bilateral e pela fundamentação teórica das duas métricas. O filtro bilateral, ao buscar suavizar o ruído através da ponderação espacial e de intensidade, pode introduzir um leve borramento afetando diretamente o sistema de pontuação resultante.

Imagens	BRISQUE		NIQE	
	B-A	%	B-A	%
Quadra	3,50	6,21	0,78	15,54
Lago	8,43	14,67	0,07	1,13
Piscina	6,46	11,47	0,42	7,44
Placa	8,38	15,17	0,70	12,75
Rua	6,56	13,10	0,25	4,88
Vaso	10,10	19,41	0,06	1,04
Cadeira	7,03	10,03	0,43	6,73
Telefone	10,21	18,19	0,58	9,30
Carros	8,99	13,41	0,83	13,17
Banco	22,44	41,51	0,81	15,03

Tabela 6 – Comparação BRISQUE e NIQE - Variação Percentual

Fonte: Autoria Própria (2025).

6.2 Avaliação Visual da Recuperação de Cena

Além das análises quantitativas conduzidas pelas métricas BRISQUE e NIQE, uma avaliação visual foi realizada com o objetivo de identificar melhorias perceptíveis nas imagens ao longo da aplicação da pipeline. Essa abordagem é essencial, pois permite observar se as transformações realizadas realmente possibilitam a recuperação de informações e objetos que antes estavam ocultos ou imperceptíveis, especialmente em contextos de iluminação extremamente reduzida.

6.2.1 Recuperação de detalhes em regiões escuras

As imagens originais, captadas no formato RAW (ARW), apresentavam severa subexposição em diversas áreas da cena, impossibilitando a identificação de elementos básicos como estruturas, objetos ou superfícies. Após o processamento completo da pipeline, observou-se que:

Na Figura 18 (Quadra), áreas onde anteriormente não se distinguia passaram a exibir detalhes ao fim do processamento do pipeline.



Figura 18 – Quadra de basquete com detalhe ampliado.

Fonte: Autoria Própria (2025).

Como evidenciado pelos resultados, foi possível notar que objetos contido na cena com característica desafiadoras após o processamento foi possível ser recuperado

de modo que os detalhes e o entendimento que agora nos permitem a identificação de aspectos como cor, detalhes, formas e objetos adjacentes.



Figura 19 – Cabine telefônica com detalhe ampliado.

Fonte: Autoria Própria (2025).

Assim como na imagem anterior, na Figura 19 temos a imagem de uma cabine telefônica, há uma região com a presença de um poste de iluminação urbana a esquerda, que originalmente se encontra com pouca ou quase nenhuma visibilidade, que só se fez visível na imagem processada final.

Em síntese, a análise visual confirma que a pipeline proposta foi bem-sucedida em seu objetivo principal de transformar dados brutos do sensor em uma representação visualmente inteligível. A recuperação de detalhes textuais, a definição de bordas e a abertura de sombras foram consistentemente observadas através das diversas cenas testadas. Estes resultados qualitativos, aliados aos dados quantitativos anteriores, validam a abordagem modular e controlada como uma estratégia eficaz para o realce de imagens em baixa luz, oferecendo um equilíbrio reproduzível entre recuperação de informação, naturalidade perceptual e viabilidade computacional.

7

CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal o desenvolvimento de uma pipeline de processamento de imagem voltada para cenas capturadas em ambientes de baixa luminosidade, utilizando imagens no formato RAW. A proposta foi fundamentada em técnicas clássicas de *Image Signal Processing (ISP)*, organizadas de forma modular, e com foco na obtenção de uma imagem final perceptivamente melhorada, sem o uso de redes neurais ou modelos de aprendizado profundo. A pipeline implementada foi composta por diversos blocos de processamento, incluindo leitura e linearização de imagem RAW, *demosaiçing*, correção de cor, balanço de branco, ajuste de exposição, correção gama, redução de névoa (*dehazing*), contraste local via CLAHE, redução de ruído e ajuste de saturação. Cada etapa foi projetada para atuar em um aspecto específico da imagem degradada, com o intuito de recuperar detalhes, corrigir distorções e aprimorar a visibilidade geral da cena. A avaliação da qualidade das imagens processadas foi conduzida com base em métricas objetivas sem referência, neste caso o BRISQUE e NIQE, onde é extremamente relevante destacar que, em todos os resultados obtidos, os valores de qualidade ficaram consistentemente acima de 50% segundo os critérios dos dois métodos de avaliação adotados. Esse resultado atesta a efetividade da pipeline em cumprir seu objetivo de aprimorar imagens capturadas em ambientes de baixa luminosidade, preservando, ao mesmo tempo, características que permitem que as imagens sejam consideradas de boa qualidade pelos métodos utilizados neste trabalho.

Na análise qualitativa das imagens finais, observou-se de forma nítida a eficácia da pipeline: cenas anteriormente praticamente indistintas tornaram-se legíveis, permi-

tindo a identificação de objetos, elementos cenográficos e outras informações relevantes presentes nas imagens. Essa transformação refletiu-se em melhorias perceptíveis na visibilidade e na distinção de detalhes estruturais, evidenciando o ganho visual alcançado pelo processamento aplicado.

Um aspecto interessante observado nas análises foi que, embora a inclusão do bloco de redução de ruído tenha melhorado a nitidez perceptiva em algumas regiões, em termos de métricas objetivas, sua presença nem sempre resultou em melhores pontuações, sugerindo que a suavização pode comprometer estatísticas naturais da imagem. Ainda assim, a modularidade da pipeline permitiu comparar os cenários com e sem esse bloco, enriquecendo a avaliação crítica. Entre as contribuições deste trabalho, destacam-se

- A construção de uma pipeline robusta e interpretável para imagens *RAW* noturnas, sem uso de técnicas de *deep learning*;
- A análise detalhada dos efeitos de cada etapa da pipeline sobre a qualidade da imagem. A disponibilização de um sistema reproduzível que pode ser utilizado em diversos cenários de captura em baixa luz.

7.1 Limitações

A pipeline proposta foi testada apenas com um conjunto restrito de imagens, não sendo avaliada em uma escala mais ampla ou em diferentes sensores e modelos de câmeras, o que limita a sua generalização para outros cenários e dispositivos.

A calibração dos parâmetros foi realizada de forma manual, o que pode impactar a transferibilidade e robustez do método em contextos diversos, sugerindo a necessidade de automatização ou adaptação dinâmica.

Além disso, algumas técnicas, como a matriz de correção de cor (CCM), foram capturadas de forma prévia e não de maneira automatizada, o que poderia ser otimizado para melhorar a precisão e a flexibilidade do processo de calibração.

7.2 Trabalhos Futuros

Apesar dos resultados promissores, este trabalho abre caminho para diversas investigações futuras que podem expandir, otimizar e robustecer a pipeline proposta, mantendo-se dentro do paradigma do Processamento Digital de Imagens (PDI) clássico. As seguintes direções são sugeridas:

1. Automação e Otimização de Parâmetros: A calibração manual dos parâmetros (e.g., ganho de exposição, intensidade do CLAHE, força do filtro bilateral) limita a adaptabilidade do sistema. Trabalhos futuros podem investigar métodos de otimização automática baseados em métricas da própria imagem. Por exemplo, a utilização do histograma global para inferir o ganho de exposição ótimo, ou a análise da entropia local para regular a intensidade do CLAHE e do denoising de forma adaptativa por região, criando uma pipeline com parâmetros dinâmicos e dependentes do conteúdo.

2. Derivação Automatizada da Matriz de Correção de Cor (CCM): Para aumentar a portabilidade entre diferentes sensores, é fundamental desenvolver um método para derivar a CCM de forma automatizada e não supervisionada. Isto poderia ser realizado através da captura de um chart de cores de referência (como um ColorChecker) em condições controladas de iluminação, seguida pela aplicação de um algoritmo de ajuste por mínimos quadrados ou otimização não-linear para calcular a matriz que melhor mapeia os valores do sensor para um espaço de cor padrão (e.g., sRGB).

3. Expansão para Diferentes Sensores e Formatos RAW: A validação com um único tipo de sensor (Sony .ARW) é uma limitação. Um passo natural é estender a pipeline para uma variedade de sensores (CCD e CMOS de diferentes fabricantes) e formatos RAW (e.g., .CR2, .NEF, .DNG). Isso exigiria a criação de um módulo de interpretação de metadados genérico e a realização de testes de calibração radiométrica (flat-field, dark frame) para caracterizar e compensar as particularidades de cada sensor.

4. Investigação de Algoritmos Avançados de Denoising Preservador de Bordas: Dado o trade-off identificado entre redução de ruído e preservação de textura, futuras iterações devem testar e comparar algoritmos de denoising mais sofisticados do estado da arte do PDI. Técnicas como Non-Local Means (NLM), Block-Matching and 3D Filtering (BM3D) e variantes de filtros wavelet oferecem potencial para uma

supressão de ruído mais eficaz com menor perda de detalhes finos, podendo melhorar simultaneamente a qualidade perceptual e as métricas NR-IQA.

5. Refinamento do Fluxo Tonal e do Mapeamento de Faixa Dinâmica: A sequência atual (Exposição Linear \rightarrow Gamma) é eficaz mas básica. Pesquisas futuras podem integrar um módulo dedicado de Tone Mapping (TM) para cenários de Alto Alcance Dinâmico (HDR). Algoritmos clássicos como os de Reinhard, Durand, ou Mantiuk poderiam ser adaptados para operar no espaço linear, comprimindo a faixa dinâmica de forma mais perceptual e controlada do que uma simples correção gama, possivelmente recuperando mais detalhes simultâneos em sombras e altas-luzes.

6. Otimização Computacional para Aplicações em Tempo Real: Para aplicações embarcadas ou de vídeo (e.g., vigilância, sistemas avançados de assistência ao condutor), a otimização da eficiência computacional é crucial. Isso envolveria a reimplementação de estágios críticos (como demosaicing AHD e filtro bilateral) em linguagens de baixo nível (C/C++), a exploração de paralelismo (GPU, multi-threading) e a avaliação de algoritmos mais leves que mantenham a qualidade, visando atingir taxas de quadros adequadas para processamento em tempo real.

7. Criação de um Benchmark Público para Pipelines PDI em Baixa Luz: Para fomentar comparações justas e reprodutíveis na comunidade, um trabalho de grande valor seria a criação e publicação de um dataset benchmark padronizado. Este conjunto deveria incluir pares de imagens RAW (baixa exposição) e referências de alta qualidade, abrangendo diversos cenários, sensores e níveis de ruído, acompanhados de scripts de avaliação baseados em métricas objetivas e subjetivas.

Ao explorar essas direções, a pesquisa em pipelines PDI clássicas e interpretáveis para baixa luminosidade pode continuar a evoluir, oferecendo soluções robustas, eficientes e transparentes que complementam ou servem como baseline para abordagens baseadas em aprendizado profundo

REFERÊNCIAS

- AL-HAYTHAM, I. *Kitāb al-Manāir (The Book of Optics)*. [S.l.: s.n.], 1021. Modern edition: A. I. Sabra (Ed.), *The Optics of Ibn al-Haytham*, London: The Warburg Institute, 2001. 2 vols. [19](#)
- BAYER, B. E. *Color Imaging Array*. 1976. U.S. Patent No. 3,971,065. [32](#)
- BOUTELL, T. *PNG (Portable Network Graphics) Specification Version 1.0*. 1997. RFC 2083, W3C. [21](#)
- BUCHSBAUM, G. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, v. 310, n. 1, p. 1–26, 1980. ISSN 0016-0032. [34](#)
- CHEN, C. et al. Learning to see in the dark. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 3291–3300, 2018. [25](#), [31](#), [46](#), [60](#), [90](#), [91](#)
- DEBEVEC, P.; MALIK, J. Recovering high dynamic range radiance maps from photographs. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. Los Angeles, CA: ACM, 1997. p. 369–378. [23](#)
- FARRELL, J. E. et al. Digital camera image processing chain design. In: SINGHAL, A. et al. (Ed.). *Computational Photography: Methods and Applications*. Boca Raton: CRC Press, 2011. p. 67–102. [22](#), [23](#), [34](#)
- FIELD, A. *Descobrimdo a estatística usando o SPSS*. 2. ed. Porto Alegre: Artmed, 2009. Capítulo 2: Estatística descritiva. [67](#)
- FRASER, B. *Real World Camera Raw with Adobe Photoshop CS*. Berkeley, CA: Peachpit Press, 2004. [21](#), [22](#), [23](#), [30](#), [31](#), [35](#)
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4. ed. London: Pearson, 2018. [18](#), [19](#), [20](#), [21](#), [22](#), [23](#), [30](#), [31](#), [32](#), [34](#), [35](#), [41](#), [42](#)
- GUNTURK, B. K.; ALTUNBASAK, Y. Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, v. 11, n. 9, p. 997–1013, 2002. ISSN 1057-7149. [33](#)
- GUO, G. et al. Zero-reference deep curve estimation for low-light image enhancement. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. Disponível em: <https://arxiv.org/abs/2001.06826>. [14](#), [26](#)

- HAMILTON, J. F.; ADAMS, J. E. *Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera*. 1997. U.S. Patent No. 5,629,734. 33
- HANNAN, C. The camera obscura: A chronological history. *Journal of Photographic Science*, v. 64, n. 5, p. 456–467, 2016. 19
- HE, K.; SUN, J.; TANG, X. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, p. 2341–2353, 2011. 38, 52
- HIRAKAWA, K.; PARKS, T. W. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Transactions on Image Processing*, v. 14, n. 3, p. 360–369, 2005. ISSN 1057-7149. 33
- HOLST, G. C.; LOMHEIM, T. S. *CMOS/CCD Sensors and Camera Systems*. 2. ed. Winter Park, FL: JCD Publishing, 2011. 20, 22, 31
- IMATEST. *Gamma, Tonal Response Curve, and Related Concepts*. n.d. Acesso em: 10 dez. 2025. Disponível em: <<https://www.imatest.com/docs/gamma>>. 51
- KRIZ, J.; HUBBARD, B.; CHEN, J. Scientific imaging vs. photographic imaging: The raw file advantage. *Journal of Forensic Identification*, v. 66, n. 3, p. 201–215, 2016. 23
- LANDGREBE, D. A. *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, NJ: Wiley-Interscience, 2003. 21
- LORENTZ, V. F. et al. Benchmarking low-light image enhancement methods: A critical review. *Journal of Imaging Science and Technology*, v. 65, n. 4, p. 040408–1–040408–15, 2021. 25
- MALVAR, H. S.; HE, L.; CUTLER, R. High-quality linear interpolation for demosaicing of bayer-patterned color images. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. Montreal: IEEE, 2004. p. 485–488. 32
- MURRAY, J. D.; VANRYPER, W. *Encyclopedia of Graphics File Formats*. 2. ed. Sebastopol, CA: O'Reilly, 1996. 21
- NAYAR, S. K.; BRANZOI, V. Adaptive dynamic range imaging. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2003. 20
- NEDERMEIJER, J. et al. *WebP: Web done right*. 2013. Google Developers Blog. Accessed: 2023-08-15. Disponível em: <<https://developers.google.com/speed/webp>>. 21
- NORTON, P. R. Infrared image sensors. *Opto-Electronics Review*, v. 9, n. 4, p. 297–318, 2001. 20
- PATTANAIK, S. N. et al. A multiscale model of adaptation and spatial vision for realistic image display. In: *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. Orlando, FL: ACM, 1998. p. 287–298. 37
- PIZER, S. M. et al. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, v. 39, n. 3, p. 355–368, 1987. ISSN 0734-189X. 40
- RAMANATH, R. et al. Color image processing pipeline. *IEEE Signal Processing Magazine*, v. 22, n. 1, p. 34–43, 2005. 20, 22

- RASHEED, M. T.; SHI, D.; KHAN, H. A comprehensive experiment-based review of low-light image enhancement methods and benchmarking low-light image quality assessment. *Signal Processing*, 2023. [14](#), [15](#), [27](#), [28](#)
- REINHARD, E. et al. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. 2. ed. San Francisco, CA: Morgan Kaufmann, 2010. [22](#), [23](#), [37](#)
- ROGALSKI, A. History of infrared detectors. *Opto-Electronics Review*, v. 20, n. 3, p. 279–308, 2012. [21](#)
- SHEIKH, H. R.; BOVIK, A. C. Image information and visual quality. *IEEE Transactions on Image Processing*, v. 15, n. 2, p. 430–444, 2006. [24](#)
- SKODRAS, A.; CHRISTOPOULOS, C.; EBRAHIMI, T. The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, v. 18, n. 5, p. 36–58, 2001. [21](#)
- SMITH, A. R. Color gamut transform pairs. *ACM SIGGRAPH Computer Graphics*, v. 12, n. 3, p. 12–19, 1978. [42](#)
- SULLIVAN, G. J. et al. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, p. 1649–1668, 2012. [21](#)
- TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: *Proceedings of the Sixth International Conference on Computer Vision*. Bombay, India: IEEE, 1998. p. 839–846. [36](#)
- WALLACE, G. K. The jpeg still picture compression standard. *Communications of the ACM*, v. 34, n. 4, p. 30–44, 1991. [21](#)
- WANG, Z.; BOVIK, A. C. *Modern Image Quality Assessment*. [S.l.]: Morgan & Claypool Publishers, 2006. v. 2. 1–156 p. (Synthesis Lectures on Image, Video, and Multimedia Processing, 1). [23](#), [24](#)
- XU, X. et al. Snr-aware low-light image enhancement. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2022. p. 17714–17724. [14](#), [15](#), [26](#)
- ZAMIR, S. W. et al. Learning digital camera pipeline for extreme low-light imaging. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA: IEEE, 2019. [14](#), [25](#)
- ZHANG, L.; WU, X. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Transactions on Image Processing*, v. 14, n. 12, p. 2167–2178, dez. 2005. ISSN 1057-7149. [33](#)
- ZHANG, Y. et al. A comprehensive survey on deep learning-based low-light image enhancement. *Computational Visual Media*, v. 7, n. 3, p. 329–365, 2021. [25](#)
- ZUIDERVELD, K. Contrast limited adaptive histogram equalization. In: HECKBERT, P. S. (Ed.). *Graphics Gems IV*. San Diego: Academic Press, 1994. p. 474–485. [40](#)

A

PRIMEIRO APÊNDICE

O presente apêndice reúne todas as imagens finais resultantes do pipeline de processamento desenvolvido neste trabalho. O objetivo deste capítulo é fornecer um registro visual completo dos resultados obtidos. Este apêndice, portanto, funciona como um complemento essencial à discussão técnica e aos resultados quantitativos apresentados, consolidando os aspectos visuais fundamentais para a compreensão da eficácia do pipeline proposto



Figura 20 – Quadra de basquete e Lago

Fonte: Autoria Própria (2025).



Figura 21 – Rua e Piscina.
Fonte: Autoria Própria (2025).

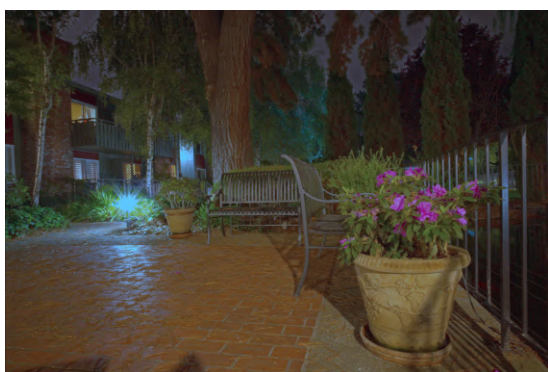


Figura 22 – Vaso e Cadeira.
Fonte: Autoria Própria (2025).

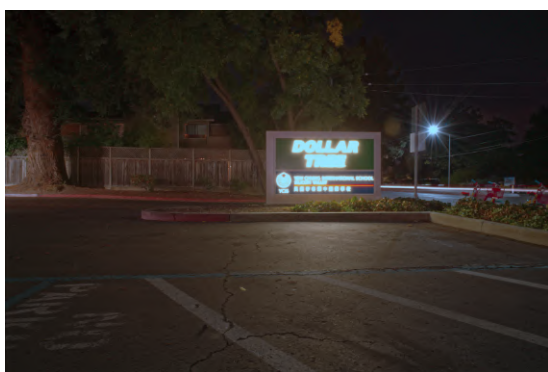


Figura 23 – Placa e banco.
Fonte: Autoria Própria (2025).



Figura 24 – Carro e Telefone.
Fonte: Autoria Própria (2025).

A

SEGUNDO APÊNDICE

Nesse capítulo está contido todo o código fonte utilizado para a realização do processamento do pipeline.

```
1 from utils.pipe import *
2
3 if __name__ == '__main__':
4
5     print(' Basquete')
6     pipe(raw_path1, 'Basquete')
7     move_dir(origem, '1-Basquete')
8
9     print(' Lago')
10    pipe(raw_path2, 'Lago')
11    move_dir(origem, '2-Lago')
12
13    print(' Piscina')
14    pipe(raw_path3, 'Piscina')
15    move_dir(origem, '3-Piscina')
16
17    print(' Placa')
18    pipe(raw_path4, 'Placa')
19    move_dir(origem, '4-Placa')
20
21    print(' Rua')
22    pipe(raw_path5, 'Rua')
```

```
23     move_dir(origem, '5-Rua')
24
25     print(' Vaso')
26     pipe(raw_path6, 'Vaso')
27     move_dir(origem, '6-Vaso')
28
29     print(' Lixo')
30     pipe(raw_path7, 'Cadeira')
31     move_dir(origem, '7-Cadeira')
32
33     print(' Telefone')
34     pipe(raw_path9, 'Telefone')
35     move_dir(origem, '9-Telefone')
36
37     pipe(raw_path14, 'Carros')
38     move_dir(origem, '14-Carros')
39
40     print(' Banco')
41     pipe(raw_path15, 'Banco')
42     move_dir(origem, '15-Banco')
```

```
1  from utils.utilities import *
2  from utils.balanco_de_branco import *
3  from utils.clahe import *
4  from utils.correcao_de_cor import *
5  from utils.correcao_ruido import *
6  from utils.correcao_de_preto import *
7  from utils.dehaze import *
8  from utils.exposicao import *
9  from utils.demosaico import *
10 from utils.gamma import *
11 from utils.leitura_raw import *
12 from utils.saturacao import *
13
14 def pipe(caminho_arquivo, name):
```

```
15
16     print("\n Iniciando pipeline de processamento...\n")
17     image, raw = leitura_raw(caminho_arquivo)
18     image = correcao_de_preto(image, raw)
19     image = demosaico(image, raw)
20     image = balanco_de_branco(image)
21     image = exposicao(image)
22     image = correcao_de_cor(image)
23     image = gamma(image)
24     image = correcao_ruído(image)
25     image = dehaze(image)
26     image = clahe(image)
27     image = saturacao(image)
28     run_brisque(image, name)
```

```
1  from .utilities import *
2
3  # 1. Leitura da imagem RAW
4  def leitura_raw(path):
5      raw = rawpy.imread(path)
6      image = np.array(raw.raw_image, dtype=np.double)
7      save_image("01_raw", image)
8      return image, raw
```

```
1  from .utilities import *
2
3  # Balanço de branco
4  def balanco_de_branco(image):
5
6      mean_channels = np.mean(image, axis=(0, 1))
7      gain = mean_channels.mean() / mean_channels
8      gain = np.clip(gain, 0.5, 3.0)
9
10     balanced = image * gain.reshape(1, 1, 3)
11
```

```
12     save_image("04_balanco_de_branco", balanced)
13     return b alanced
```

```
1  from .utilities import *
2
3  # 3. Demosaico
4  def demosaico(image, raw):
5      """Aplica demosaicing usando o algoritmo AHD."""
6      image = raw.postprocess(
7          demosaic_algorithm=rawpy.DemosaicAlgorithm.AHD
8      )
9
10     image = np.asarray(image, dtype=np.double)
11     save_image("03_demosaic", image)
12     return image
```

```
1  from .utilities import *
2
3  # 5 corrigir_exposição
4  def exposicao(imagem, ganho=0.8):
5      imagem_corrigida = imagem * ganho
6      images = imagem_corrigida.astype(np.float32) / 255.0
7      final = np.clip(images, 0, 1)
8      save_image("05_exposição", final)
9      return final
```

```
1  from .utilities import *
2  from .balanco_de_branco import *
3
4  # 6
5  def correcao_de_cor(img, ccm=None):
6      """
7      Aplica matriz de correção de cor (CCM) à imagem RGB normalizada.
8      img: numpy array float [0,1], shape (H,W,3)
9      ccm: matriz 3x3 numpy array, se None usa CCM padrão
```

```
10
11     Retorna: imagem corrigida, float [0,1]
12     """
13     if ccm is None:
14
15         ccm = np.array([[ 0.5838, -0.143, -0.0246],
16                        [-0.3497,  1.1477,  0.2297],
17                        [-0.0748,  0.1885,  0.5778]])
18
19     # Aplicar correção pixel a pixel
20     h, w, _ = img.shape
21     img_reshaped = img.reshape(-1, 3)
22     corrected = np.dot(img_reshaped, ccm.T)
23     corrected = corrected.reshape(h, w, 3)
24     corrected = np.clip(corrected, 0, 1)
25     corrected = balanco_de_branco(corrected)
26     save_image("06_color-correction", corrected)
27     return corrected
```

```
1 from .utilities import *
2
3 # 8 Gamma
4 def gamma(image, gamma=2.2):
5     image = np.power(image / image.max(), 1 / gamma)
6     save_image("08_gamma_corrected", image)
7     return image
```

```
1 from .utilities import *
2
3 def correcao_ruido(imagem):
4     # image = denoise_bilateral(imagem, channel_axis=-1)
5     image = denoise_bilateral(imagem, multichannel=True)
6     save_image("0_denoise", image)
7     return image
```

```
1 from .utilities import *
2
3 # 9 Dehaze
4 def dark_channel(imagem, window_size=15):
5     min_channel = np.min(imagem, axis=2)
6     kernel = cv2.getStructuringElement(cv2.MORPH_RECT,
7         (window_size, window_size))
8     dark = cv2.erode(min_channel, kernel)
9     return dark
10
11 def atmospheric_light(imagem, dark):
12     h, w = dark.shape
13     num_pixels = h * w
14     num_brightest = int(max(num_pixels * 0.001, 1)) #top 0.1% pixels
15
16     dark_vec = dark.reshape(num_pixels)
17     img_vec = imagem.reshape(num_pixels, 3)
18
19     indices = np.argsort(dark_vec)[-num_brightest:]
20     atmospheric = np.max(img_vec[indices], axis=0)
21     return atmospheric
22
23 def transmission_estimate(imagem, atmospheric,
24 omega=0.95, window_size=15):
25     norm_img = imagem / atmospheric
26     transmission = 1 - omega * dark_channel(norm_img, window_size)
27     return np.clip(transmission, 0.1, 1)
28
29 def guided_filter(I, p, radius=60, eps=1e-3):
30     mean_I = cv2.boxFilter(I, cv2.CV_64F, (radius, radius))
31     mean_p = cv2.boxFilter(p, cv2.CV_64F, (radius, radius))
32     corr_I = cv2.boxFilter(I * I, cv2.CV_64F, (radius, radius))
33     corr_Ip = cv2.boxFilter(I * p, cv2.CV_64F, (radius, radius))
34
```

```
35     var_I = corr_I - mean_I * mean_I
36     cov_Ip = corr_Ip - mean_I * mean_p
37
38     a = cov_Ip / (var_I + eps)
39     b = mean_p - a * mean_I
40
41     mean_a = cv2.boxFilter(a, cv2.CV_64F, (radius, radius))
42     mean_b = cv2.boxFilter(b, cv2.CV_64F, (radius, radius))
43
44     q = mean_a * I + mean_b
45     return q
46
47 def recover_scene_radiance(imagem, transmission, atmospheric,
48 t0=0.1):
49     transmission = np.clip(transmission, t0, 1)
50     J = (imagem - atmospheric) /
51     transmission[..., np.newaxis] + atmospheric
52     return np.clip(J, 0, 1)
53
54 def dehaze(imagem, save_path=None, omega=0.60, t0=0.2,
55 window_size=15, guided_filter_radius=60, eps=1e-3):
56     """
57     Remove névoa de uma imagem RGB normalizada [0,1].
58     Se save_path for fornecido, salva a imagem processada.
59     """
60     imagem = np.clip(imagem, 0, 1)
61     dark = dark_channel(imagem, window_size)
62     A = atmospheric_light(imagem, dark)
63     transmission = transmission_estimate(imagem, A, omega,
64                                         window_size)
65
66     gray = cv2.cvtColor((imagem *255).astype(np.uint8),
67 cv2.COLOR_RGB2GRAY) / 255.0
68     transmission_refined = guided_filter(gray,
69     transmission, guided_filter_radius, eps)
```

```
70
71     recovered = recover_scene_radiance(imagem,
72                                       transmission_refined, A, t0)
73
74     if save_path:
75         img_save = (recovered * 255).astype(np.uint8)
76         cv2.imwrite(save_path, cv2.cvtColor(img_save,
77                                             cv2.COLOR_RGB2BGR))
78
79     save_image("09_De haze", recovered)
80
81     return recovered
```

```
1  from .utilities import *
2
3  # 10. CLAHE
4  def clahe(imagem):
5      lab = cv2.cvtColor((imagem * 255).astype(np.uint8),
6                       cv2.COLOR_RGB2LAB)
7      l, a, b = cv2.split(lab)
8
9      clahe = cv2.createCLAHE(clipLimit=0.2, tileGridSize=(2,2))
10     l_clahe = clahe.apply(l)
11
12     lab_clahe = cv2.merge((l_clahe, a, b))
13     imagem_clahe = cv2.cvtColor(lab_clahe, cv2.COLOR_LAB2RGB)
14     imagem_clahe = imagem_clahe.astype(np.float32) / 255.0
15     save_image("10_CLAHE", imagem_clahe)
16     return imagem_clahe
```

```
1  from .utilities import *
2
3  def saturacao(image, saturation_scale=2.0, vividness_scale=3.0):
4
5      image = convert_image(image)
6
7      hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

```
8
9     hsv_image[:, :, 1] = cv2.normalize(
10         hsv_image[:, :, 1] * saturation_scale, None,
11         0, 255, cv2.NORM_MINMAX
12     )
13
14     hsv_image[:, :, 2] = cv2.normalize(
15         hsv_image[:, :, 2] *
16         vividness_scale, None, 0, 255,
17         cv2.NORM_MINMAX
18     )
19
20     result_image = cv2.cvtColor(hsv_image, cv2.COLOR_HSV2BGR)
21     save_image("11_saturation", result_image)
22     return result_image
```

A

PRIMEIRO ANEXO.

Imagens do *dataset See in the Dark* (CHEN et al., 2018), utilizadas como entrada para o processamento do pipeline.



Figura 25 – Quadra de basquete e Lago

Fonte: *dataset See in the Dark* (CHEN et al., 2018)



Figura 26 – Placa e Piscina

Fonte: *dataset See in the Dark* (CHEN et al., 2018)

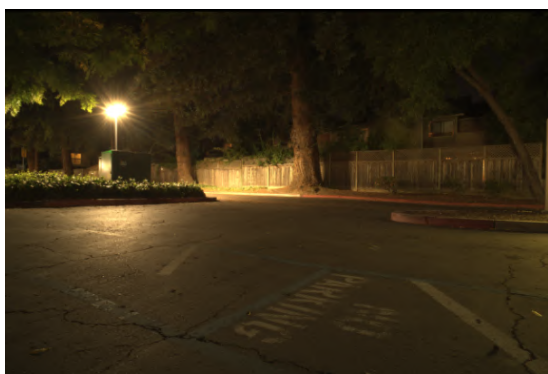


Figura 27 – Rua e Vaso

Fonte: dataset *See in the Dark* (CHEN et al., 2018)



Figura 28 – Cadeiras e Telefone

Fonte: dataset *See in the Dark* (CHEN et al., 2018)



Figura 29 – Carro e Banco

Fonte: dataset *See in the Dark* (CHEN et al., 2018)