

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Lara Fischer

Detecção de Objetos e Segmentação Semântica em Imagens Degradadas pelo Efeito da Chuva

Florianópolis
2026

Lara Fischer

Detecção de Objetos e Segmentação Semântica em Imagens Degradadas pelo Efeito da Chuva

Dissertação de mestrado apresentada para o Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina, para a obtenção do grau de mestre em Engenharia Elétrica. Orientador: Rui Seara.

Florianópolis
2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Fischer, Lara
Detecção de objetos e segmentação semântica em imagens degradadas pelo efeito da chuva / Lara Fischer ; orientador, Rui Seara, 2026.
50 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2026.

Inclui referências.

1. Engenharia Elétrica. 2. Deraining. 3. Inteligência Artificial. 4. Detecção de Objetos. 5. Segmentação Semântica. I. Seara, Rui. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Lara Fischer

Detecção de Objetos e Segmentação Semântica em Imagens Degradadas pelo Efeito da Chuva

O presente trabalho em nível de mestrado foi avaliado e aprovado, em 10/12/2025, por banca examinadora composta pelos seguintes membros:

Prof. Rui Seara, Dr.
Universidade Federal de Santa Catarina
Campus Florianópolis

Prof. Ciro André Pitz, Dr.
Universidade Federal de Santa Catarina
Campus Blumenau

Prof. Marcos Vinicius Matsuo, Dr.
Universidade Federal de Santa Catarina
Campus Blumenau

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Engenharia Elétrica, na área de concentração Comunicações e Processamento de Sinais, pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.



Documento assinado digitalmente
Carlos Renato Rambo
Data: 10/03/2026 16:18:43-0300
CPF: ***.748.198-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Carlos Renato Rambo, Dr.
Coordenação do Programa de Pós-Graduação



Documento assinado digitalmente
Rui Seara
Data: 10/03/2026 07:40:54-0300
CPF: ***.107.869-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Rui Seara, Dr.
Orientador

Florianópolis, 10 de dezembro de 2025

Há coragem em imaginar uma realidade diferente. Que os sonhos floresçam em mentes dispostas e corajosas.

AGRADECIMENTO

A pesquisa dentro da sociedade brasileira não é nem de longe uma carreira glamourosa. Conceber este trabalho teve o atravessamento de diversas questões de natureza financeira, estrutural, pessoal. Houve muitos momentos em que pensei em desistir. Essas pessoas que estão nesta dedicatória fazem parte do grupo de pessoas que me acolheu e que me deu forças para concluir este grande projeto.

Agradeço primeiramente ao meu orientador, professor Rui Seara; sua força de vontade e persistência me inspiraram e me inspiram a querer fazer um trabalho de excelência. Ao professor Ênio dos Santos Silva, pelo apoio, incentivo e contribuições no desenvolvimento desta pesquisa. Por diversas vezes eu comentei que, apesar do trabalho ainda não estar acabado, os meus agradecimentos eu já sabia a quem direcionar, pois se não fossem os vários dias de reuniões debruçados sobre códigos e mais códigos, não sei onde essa pesquisa estaria. Ao Elton, pela agilidade, sinceridade e por construir parte das imagens utilizadas nesta pesquisa e no artigo derivado dela. Aos meus professores do curso de mestrado e da graduação, vocês engrandeceram minha jornada e formaram a pesquisadora que sou. Ao CNPq, pelo apoio financeiro, à UFSC e ao LINSE por toda infraestrutura disponibilizada para a realização deste trabalho.

Aos meus pais e a minha irmã, obrigada por me ouvirem e me apoiarem. Vocês sempre enxergaram em mim muito mais do que eu mesma fui capaz. Obrigada por acreditarem no meu potencial. Ao meu sobrinho lindo, espero que em algum momento você se depare com estes agradecimentos e saiba que é motivo de muito orgulho e alegria para esta dinda babona. Ao meu namorado, obrigada por estar ao meu lado nessa trajetória, me ouvindo reclamar, chorar e duvidar de mim mesma. Sei que você nunca duvidou. Sou muito grata por isso.

Aos meus amigos queridos, sou muito abençoada por tê-los na minha vida. São muitos nomes para citar, abençoada que sou, mas saibam que cada taça de vinho, cada praia, cada roda de conversa me deu mais força. Obrigada por serem quem são. A minha psicóloga, obrigada por transformar a minha vida e me aproximar, a cada sessão, da pessoa que quero ser.

E por fim, agradeço a mim mesma. Essa pesquisa têm muito esforço, muitas horas de dedicação, muito nervosismo e medos superados em cada linha escrita, apagada, corrigida e aprovada. Sou muito mais potente agora que passei por ela.

RESUMO

Em aplicações de visão computacional, a qualidade das imagens é essencial para processos de alta complexidade, como é o caso da detecção de objetos e/ou segmentação semântica. No entanto, perturbações causadas pela chuva podem comprometer a precisão da tarefa em questão. Isso acontece porque a chuva distorce o contorno dos objetos, muda a cor, distorce a perspectiva de distância, dentre outros aspectos. Para mitigar tal impacto, uma etapa de pré-processamento, baseada em um modelo de inteligência artificial, é usualmente considerada. Essa técnica é usualmente denominada na literatura por *deraining*. Este trabalho investiga o desempenho de um dos modelos do estado-da-arte de *deraining* em imagens submetidas a diferentes níveis de perturbações. As imagens resultantes do processo de *deraining* são posteriormente processadas por algoritmos de detecção de objetos e segmentação semântica, os quais verificam se as características semânticas relevantes da imagem têm sido recuperadas. Resultados de simulação computacional corroboram a aplicabilidade do modelo de *deraining* considerado.

Palavras-chave: Base de dados de imagem com chuva, *deraining* de uma imagem, detecção de objetos, segmentação semântica, visão computacional.

ABSTRACT

In computer vision applications, image quality is essential for highly complex processes, such as object detection and/or semantic segmentation. However, disturbances caused by rain can compromise the accuracy of the task at hand. This happens because rain distorts the outline of objects, changes color, distorts the perspective of distance, among other aspects. To mitigate such an impact, a preprocessing step, based on an artificial intelligence model, is usualy considered. This technique is referred to in the literature deraining. This work investigates the performance of one of the state-of-the-art models of deraining in images subjected to different levels of perturbations. The images resulting from the deraining process are further processed by object detection and semantic segmentation algorithms, which verify whether the relevant semantic characteristics of the image have been retrieved. Computer simulation results corroborate the applicability of the deraining model considered.

Keywords: Rain dataset, single image deraining, object detection, semantic segmentation, computer vision.

Lista de Figuras

1 Diferentes tipos de degradações provocadas pelo efeito da chuva. (a) Raias de chuva. (b) Gotas de chuva. (c) Chuva torrencial. (Figura adaptada de [1]). . . . 13

2 Exemplo de uma câmara bi-partida empregada na aquisição de imagens de cenas com e sem chuva. 16

3 Par de imagens confeccionado para este estudo e utilizado no treinamento do modelo RESCAN. (à esquerda) Imagem original. (à direita) Imagem com chuva artificialmente adicionada. 21

4 Ilustração de um bloco recorrente da arquitetura RESCAN. $\mathbf{O}_s$, $\hat{\mathbf{R}}_s$ e $\mathbf{B}_s$ representam, respectivamente, a imagem observada, as rais de chuva e a imagem após o processo de *deraining* na etapa **s**. SE representa a rede. 22

5 Exemplos mostrando diferentes níveis de perturbação por raias de chuva. 24

6 Histograma dos valores da PSNR após o processo de *deraining*, considerando o conjunto de dados de teste com perturbação de 12 dB, para as camadas recorrentes consideradas. 26

7 Exemplos mostrando imagens após o processo de *deraining*; imagens que antes possuíam perturbações de 12 dB advindas das 4 últimas camadas de recorrência. 26

8 Ilustração da utilização da rede de detecção de objetos YOLOv8n em diferentes imagens. (a) Imagem original, isto é, antes da adição de chuva. (b) Imagem com chuva artificial. (c) Imagem após o processo de *deraining*. 30

9 Exemplos de falhas de classificação do modelo YOLOv8n. (a) Classificação de um vaso como bola de esporte. (b) Classificação da Lua como uma bola de esporte e um camelo como um cavalo. 31

10 Ilustração da utilização da detecção de objetos pelo modelo YOLOv8n. (a) Imagem original, ou seja, antes da adição de chuva, identificada como pessoa com 98% de confiança. (b) Imagem com 12 dB de perturbação de chuva, sem qualquer identificação. (c) Imagem após o pré-processamento de *deraining*, identificada como pessoa com 99% de confiança 32
11 Ilustração da utilização da segmentação semântica pela rede *Mask* R-CNN. (a) Imagem original. (b) Imagem com 18 dB de perturbação de chuva. (c) Imagem após o pré-processamento de *deraining* advinda de uma perturbação de 12 dB. . 38
12 Ilustração da utilização da segmentação semântica pelo modelo *Mask* R-CNN. (a) Imagem original. (b) Imagem com 18 dB de perturbação de chuva. (c) Imagem após o pré-processamento de *deraining* advinda de uma perturbação de 12 dB. 39

Lista de Tabelas

1 Avaliação das Imagens com Chuva Artificial 27

2 Avaliação do Processo de *Deraining* 27

3 Comparação dos resultados de detecção dos modelos YOLOv8n e *Fast* R-CNN com diferentes níveis de PSNR, sob chuva artificial 33

4 Resultados de detecção do modelo YOLOv8n nas camadas #2, #3 e #4 com diferentes níveis de PSNR, após o processo de *deraining* 34

5 Resultados de detecção do modelo *Fast* R-CNN nas camadas #2, #3 e #4 com diferentes níveis de PSNR, após o processo de *deraining* 34

6 Avaliação da segmentação semântica dos modelos YOLOv8 e *Mask* R-CNN no conjunto de dados com chuva artificial 37

7 Avaliação da segmentação semântica do modelo YOLOv8 no conjunto de dados após o processo de *deraining* nas diferentes camadas selecionadas 37

8 Avaliação da segmentação semântica do modelo *Mask* R-CNN no conjunto de dados após o processo de *deraining* nas diferentes camadas selecionadas 38

Sumário

1 Introdução **13**

1.1 Modelos de *deraining* como etapa de pré-processamento 14

1.2 Objetivos do trabalho 18

1.3 Estrutura da dissertação 19

2 Processo de *Deraining* **20**

2.1 Introdução 20

2.2 *Recurrent squeeze-and-excitation context aggregation net* - RESCAN 20

2.3 Construção do *dataset* para os experimentos 22

2.4 Desempenho da RESCAN sob Diferentes Níveis de Degradação 24

2.5 Conclusões 27

3 Impacto do *Deraining* como Etapa de Pré-Processamento **29**

3.1 Introdução 29

3.2 Integração do Processo de *Deraining* com Detecção de Objetos 30

3.3 Impacto do Uso de *Deraining* na Segmentação Semântica 35

3.4 Conclusões 39

4 Considerações Finais **40**

4.1 Conclusões 40

4.2 Sugestões para trabalhos futuros 41

4.3 Trabalho publicado oriundo da dissertação 42

A Código para Geração de Chuva Sintética **43**

Apêndice A 45

Referências **50**

Capítulo 1

Introdução

A percepção visual de imagens é essencial para diversos sistemas de visão computacional, sendo um fator crítico para a interpretação precisa de cenas em diferentes contextos. No entanto, a aquisição de imagens em condições adversas, como, por exemplo, na presença de chuva, representa um desafio significativo para a sua interpretabilidade [1]. A chuva pode introduzir distorções ópticas, como dispersão da luz, redução do contraste e formação de diferentes padrões indesejados (artefatos), como ilustrado na Figura 1. Esses eventos prejudicam a eficácia de algoritmos de análise de imagens, principalmente quando tratam de tarefas nas quais a qualidade da cena retratada é essencial, como direção autônoma, sistemas de monitoramento de tráfego [2], serviços de vigilância e inspeções remotas por *drones* [3], aplicações de fotografia e videografia voltadas à restauração da qualidade estética e informativa das imagens, sistemas de visão para robótica móvel e dispositivos de assistência, dentre outras aplicações. Para mitigar esses efeitos, métodos de pré-processamento vêm sendo desenvolvidos com o objetivo de melhorar a nitidez e a fidelidade visual das cenas degradadas [1, 4]. Uma análise desses métodos está detalhada nas seções que se seguem.



(a) (b) (c)

Figura 1: Diferentes tipos de degradações provocadas pelo efeito da chuva. (a) Raias de chuva. (b) Gotas de chuva. (c) Chuva torrencial. (Figura adaptada de [1]).

1.1 Modelos de *deraining* como etapa de pré-processamento

O processo de remoção de chuva em imagens, denominado *deraining* ou *single image deraining* (SID) ou ainda *video deraining*, refere-se ao conjunto de técnicas e métodos voltados à atenuação ou eliminação de artefatos introduzidos por gotas e estrias de chuva no processo de captura tanto de imagens quanto de vídeos [5]. Tais interferências não apenas degradam a qualidade perceptual da imagem sob a ótica da visão humana, como também comprometem de maneira significativa o desempenho de algoritmos de visão computacional, que dependem de representações visuais consistentes para a extração de características discriminativas. Nesse sentido, o *deraining* desempenha papel central ao conciliar dois objetivos complementares: a restauração visual, que busca recuperar aspectos estruturais e texturais da cena, e o aprimoramento da qualidade da imagem visando aplicações em sistemas automáticos, tal qual detecção, rastreamento e segmentação [2].

Dessa forma, os métodos de *deraining* constituem um campo de investigação essencial para a ampliação da robustez e da aplicabilidade de sistemas de visão em condições ambientais adversas. Da perspectiva das técnicas de *deraining* em uma imagem usando modelos de aprendizado profundo, observa-se que a literatura destaca três pilares como essenciais para sua caracterização e avaliação; são eles: o tipo de modelo, o conjunto de dados e as métricas utilizadas para a avaliação [1]. Discutiremos esses três aspectos a seguir:

- **Modelos.** A remoção de artefatos de chuva em imagens tem se consolidado como um campo de pesquisa ativo na área de visão computacional, impulsionado pela necessidade de sistemas visuais mais robustos em cenários adversos. Nesse contexto, diferentes arquiteturas de aprendizado profundo vêm sendo exploradas, cada uma buscando capturar propriedades específicas das imagens degradadas e oferecer soluções eficazes ao problema. Diversas arquiteturas têm sido propostas para a construção de modelos de *deraining*, cada uma fundamentada em diferentes estratégias de aprendizado e extração de características visuais. Dentre elas, destacam-se as redes recorrentes (*recurrent neural network* - RNNs) [6], que exploram dependências temporais e relacionamentos sequenciais entre padrões de chuva, sendo particularmente úteis em cenários de vídeo, nos quais a correlação entre quadros auxilia na remoção de ruídos persistentes. As redes convolucionais (*convolutional neural networks* - CNNs) [7], por sua vez, figuram dentre as abordagens mais tradicionais, uma vez que sua capacidade de capturar padrões locais e hierárquicos as torna

adequadas para identificar texturas lineares e estruturas características das gotas ou trilhas de chuva. Modelos generativos, tais como *generative adversarial networks* (GANs) [8], são amplamente empregados por sua habilidade em produzir reconstruções fotorrealistas, explorando a competição entre gerador e discriminador para preservar detalhes estruturais e semânticos da cena "limpa". Mais recentemente, arquiteturas baseadas em *transformers* [9] têm ganhado destaque pela capacidade de modelar dependências globais de longo alcance na imagem, permitindo a remoção de chuva mesmo quando os artefatos se distribuem de maneira irregular e não local. Além disso, variações arquiteturais como *autoencoders*, convoluções dilatadas e estruturas *encoder-decoder* [10–12] vêm sendo empregadas com o objetivo de equilibrar eficiência computacional e preservação de detalhes finos, oferecendo soluções mais adaptáveis a diferentes condições de degradação. Apesar dos avanços significativos dessas abordagens, o problema permanece desafiador em aplicações do mundo real [13], discutidas com mais detalhes na sequência.

- **Conjuntos de imagens.** O desempenho de um modelo de inteligência artificial está fortemente ligado à natureza e à diversidade dos dados usados no treinamento. Os conjuntos de imagens utilizados nas arquiteturas de *deraining* se dividem em sintéticos e reais. Como representantes dos conjuntos de dados com imagens reais, destacamos o *SPA-Data* [14] e *Raidar* [15], os quais armazenam imagens de locais com chuva e sem chuva, não necessariamente correspondendo a mesma cena. Em contrapartida, [16] apresenta pares de imagens adquiridas por câmeras duplas sincronizadas com um painel transparente à frente, com e sem gotas d'água (visando, no primeiro caso, emular um cenário de chuva). A Figura 2, extraída de [16], mostra um exemplo de câmera bi-partida, utilizada na aquisição das imagens em [16]. Nela, painéis acrílicos transparentes são posicionados diante das lentes, mantendo-se a seção esquerda permanentemente seca, enquanto a seção direita é artificialmente submetida à pulverização de gotículas de água por meio de um bocal interno. Essa estratégia permite a obtenção de pares de imagens comparáveis da mesma cena, diferenciando-se, contudo, pela ligeira variação de ângulo decorrente da divisão física do sistema ótico. Apesar das referências supracitadas, a dificuldade em obter imagens reais pareadas representa um desafio importante. Pois, mesmo que pequenas, as alterações, como movimentos devido ao vento e leves trepidações da câmera, comprometem ainda a sua utilização por algoritmos que empregam a verificação *pixel* a *pixel*. Por isso, usualmente são considerados usar, para o treinamento, conjuntos de imagens

de natureza sintética. Nesses casos, o efeito da chuva é artificialmente incorporado às imagens sem chuva a partir de diferentes modelos matemáticos, que buscam reproduzir as características físicas do fenômeno da chuva. Dentre os mais utilizados, destacam-se os modelos aditivos, nos quais a chuva é representada como uma camada de intensidade luminosa somada à imagem original, e os modelos multiplicativos, os quais simulam a interação da chuva com a iluminação do ambiente, produzindo variações de transparência e brilho. Há ainda abordagens mais complexas baseadas em funções de espalhamento e difusão da luz, que procuram reproduzir efeitos atmosféricos adicionais, como "borrões" causados pelo movimento, refração ou dispersão da luminosidade, causadas pelas gotas. Essa diversidade de modelagens é essencial para gerar bancos de imagens mais realistas e desafiadores, permitindo que os algoritmos de *deraining* sejam expostos a diferentes padrões de degradação durante o treinamento. Dentre alguns bancos de imagens da literatura, destacam-se o *Rain100H* [17], que contém 100 imagens sem chuva, com cenas de alta complexidade estrutural e suas respectivas versões degradadas por chuva densa; o *Rain100L* [17], focado em cenas com estruturas mais simples e menos texturizadas; e o *Rain1400*, o qual possui 1400 imagens sintéticas geradas a partir de 100 imagens "limpas" com 14 padrões distintos de chuva [11]. Embora esses *datasets* ofereçam controle experimental e grande quantidade de dados, sua generalização para cenários reais é limitada. Em geral, a estrutura dos *datasets* para *deraining* varia em termos de resolução, diversidade de cenas, intensidade e tipo de chuva, sendo fundamental compreender essas diferenças para a seleção apropriada do conjunto de dados durante o treinamento e avaliação dos modelos.

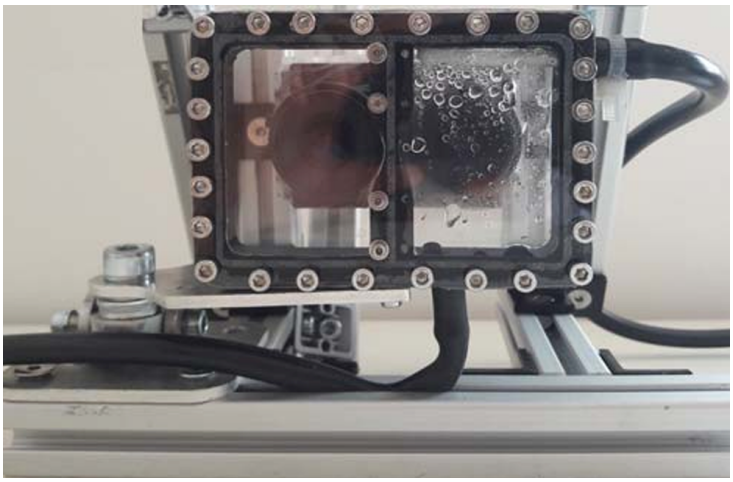


Figura 2: Exemplo de uma câmara bi-partida empregada na aquisição de imagens de cenas com e sem chuva.

- **Métricas.** Por fim, a avaliação de desempenho é crucial para validar a eficácia de qualquer abordagem. Em *deraining*, essa avaliação é comumente feita por métricas quantitativas objetivas, tais como o *peak signal-to-noise ratio* (PSNR) e o *structural similarity index measure* (SSIM), que mensuram, respectivamente, a fidelidade do sinal reconstruído em relação à imagem de referência e a preservação de aspectos estruturais e perceptuais [18, 19]. Em contextos em que a análise perceptual é relevante, também são utilizadas métricas não referenciais, como o *natural image quality evaluator* (NIQE) e o *blind/referenceless image spatial quality evaluator* (BRISQUE), capazes de estimar a qualidade visual sem a necessidade de imagens de referência [20, 21]. Adicionalmente, em cenários que envolvem integração com tarefas de alto nível, como, por exemplo, detecção de objetos e/ou segmentação semântica, indicadores derivados do desempenho nessas tarefas — por exemplo, *mean intersection over union* (mIoU) e *average precision* (AP) — são utilizados para aferir o impacto do processo de *deraining* sobre aplicações subsequentes [22, 23].

Dentro deste panorama, um modelo promissor recentemente evidenciado na literatura é o RESCAN (*recurrent squeeze-and-excitation context aggregation net*) [24]. Sua estrutura combina redes convolucionais profundas com redes recorrentes. Seu desempenho já foi comprovado em diversas aplicações. No entanto, ainda existem lacunas importantes a serem exploradas, como, por exemplo, a ausência de um mapeamento indicando os níveis de ruído e/ou degradações nos quais o modelo poderia ser bem-sucedido ou apresentar falhas. E, também, de que forma o modelo beneficia aplicações posteriores das imagens tratadas (como no caso de detecção de objetos e/ou segmentação semântica).

A relação entre os processos de restauração e a utilização subsequente das imagens revela-se particularmente relevante, uma vez que, em cenários reais, a presença de chuva pode ocultar características visuais essenciais [13], tais como bordas e texturas, comprometendo de forma significativa o desempenho de algoritmos convencionais de detecção e segmentação semântica, como *fast regions with* CNNs (*Fast* R-CNNs [25]), *you only look once* (YOLO [26]), *mask region-based convolutional neural networks* (*Mask* R-CNN [27]) e *single shot multi-box detector* (SSD [28]). Alguns trabalhos na literatura buscam criar redes voltadas para essa conexão entre as tarefas de alta complexidade e o pré-processamento com *deraining* [29–32]. Essas abordagens geralmente seguem arquiteturas do tipo *end-to-end*, integrando módulos de restauração de imagem com detectores ou segmentadores semânticos, de modo a

preservar ou recuperar pistas visuais relevantes que seriam prejudicadas na presença de chuva. Essa integração mostra-se promissora, mas ainda enfrenta desafios, sobretudo pela carência de métricas e *datasets* projetados para avaliar simultaneamente restauração e interpreta- ção. Na literatura, grande parte dos *datasets* de *deraining* não têm sido concebida para uso direto com detectores ou segmentadores, e vice-versa, o que limita o avanço dessas abordagens integradas. Diante dessas considerações, observa-se que o tópico de pesquisa em métodos SID ainda apresenta lacunas importantes, especialmente no que diz respeito à integração eficaz entre restauração de imagens e aplicações subsequentes de alto nível. A ausência de *datasets* adequados para avaliação conjunta, limitações nas métricas existentes e a falta de estudos sistemáticos que relacionem a intensidade da degradação às capacidades dos modelos representam importantes desafios a serem suplantados. Nesse contexto, o presente trabalho propõe abordagens que respondam a essas limitações, explorando soluções que conciliem desempenho de restauração com impacto positivo em tarefas posteriores. A seguir, são apresentados os principais pontos de contribuição deste trabalho de pesquisa.

1.2 Objetivos do trabalho

Em resumo, o objetivo geral deste trabalho é investigar o impacto do pré-processamento por *deraining* na qualidade e no desempenho de sistemas de visão computacional. Para isso, iremos:

- Construir um *dataset* com imagens com chuva sinteticamente adicionada, geradas a partir das imagens sem chuva do *dataset Rain100H* [17]. Para nossa análise, o conjunto de dados deve conter imagens com e sem chuva da mesma cena, podendo manter vinculada a elas a informação de seus correspondentes níveis de perturbação provocados pela chuva, o que são características não encontradas nos *datasets* avaliados.

- Analisar o desempenho da arquitetura RESCAN sob diferentes intensidades de degradação provocadas pelo efeito da chuva, com o intuito de avaliar a robustez do modelo no processo de *deraining*.

- Avaliar os impactos da integração do processo de *deraining* com métodos de detecção de objetos, por meio da utilização dos algoritmos YOLO [26] e *Fast* R-CNN [25]).

- Investigar o impacto da incorporação do processo de *deraining* no desempenho de métodos de segmentação semântica, utilizando, para esse propósito, os algoritmos YOLO [33] e *Mask* R-CNN [27].

1.3 Estrutura da dissertação

O restante desta dissertação é organizado conforme descrito a seguir. O Capítulo 2 apresenta a construção do *dataset* empregado, bem como as principais observações decorrentes de sua aplicação na arquitetura RESCAN. O Capítulo 3, por sua vez, discute as melhorias do processo de detecção de objetos e segmentação semântica em imagens sob o efeito da chuva a partir da adoção do pré-processamento por meio de técnicas de *deraining*. Finalmente, o Capítulo 4 apresenta as conclusões e comentários finais desta dissertação, como também propostas para trabalhos futuros.

Capítulo 2

Processo de *Deraining*

2.1 Introdução

O aprimoramento das técnicas de *deraining* está diretamente relacionado à escolha adequada tanto do *dataset* quanto do modelo de rede neural, uma vez que a qualidade e a estrutura dos dados determinam a capacidade da rede em aprender padrões de chuva e removê-los de forma eficaz. No caso da arquitetura RESCAN, o treinamento ideal requer imagens pareadas, ou seja, capturas da mesma cena com e sem a presença de chuva, como ilustrado na Figura 3. Para assegurar a correspondência entre os pares e evitar possíveis distorções, sugere-se a utilização de um conjunto de imagens com chuva sinteticamente adicionada. Entretanto, observa-se que a maioria dos *datasets* disponíveis na literatura não oferece uma organização que permita controlar de maneira precisa diferentes níveis de degradação visual. Diante dessa dificuldade, este trabalho propõe a criação de um *dataset* ad hoc, baseado nas imagens sem chuva do *dataset Rain100H* [17], de modo a viabilizar experimentos controlados e alinhados aos requisitos do modelo.

As seções seguintes detalham a arquitetura da RESCAN, o processo de elaboração do *dataset* proposto e os resultados alcançados com o modelo treinado.

2.2 *Recurrent squeeze-and-excitation context aggregation net* - RESCAN

No contexto das redes neurais aplicadas ao *deraining*, a RESCAN [24] se destaca por adotar uma abordagem particularmente eficaz. O modelo baseia-se no aprendizado super-



Figura 3: Par de imagens confeccionado para este estudo e utilizado no treinamento do modelo RESCAN. (à esquerda) Imagem original. (à direita) Imagem com chuva artificialmente adicionada.

visionado a partir de pares de imagens permitindo que a rede aprenda de forma explícita a representação dos efeitos da chuva. Ao longo das camadas recorrentes, a RESCAN desenvolve uma compreensão progressiva das características visuais associadas à chuva, o que possibilita prever como seria a imagem original sem essas interferências.

De maneira simplificada, o modelo matemático de chuva utilizado pela rede é apresentado a seguir:

$$\mathbf{O} = \mathbf{B} + \sum_{i=1}^{n} \mathbf{R}^i \tag{2.1}$$

onde $\mathbf{O}$ representa a imagem observada, $\mathbf{B}$ corresponde ao fundo da cena (*background scene*), que é o objetivo final do processo de *deraining*, e $\mathbf{R}^i$ denota as raias de chuva aprendidas em cada camada de índice i. Durante o treinamento, a rede identifica padrões visuais característicos da presença de chuva e, *pixel* a *pixel*, remove essas perturbações da imagem de entrada, aproximando-a do fundo de cena verdadeiro. Esse mecanismo permite que a RESCAN modele a interferência causada pela chuva como um ruído visual específico, capaz de ser isolado e removido com precisão. Essa abordagem é viabilizada pela combinação entre o campo receptivo ampliado, obtido por meio de convoluções dilatadas das camadas de *squeeze-and-excitation* (SE) [34] e a recorrência temporal entre os estágios da rede. Em cada estágio, os mapas de ativação gerados a partir da imagem observada $\mathbf{O}$ são comparados com os padrões previamente internalizados durante o treinamento. Esse processo iterativo permite a identificação e supressão progressiva das regiões afetadas pela chuva, mesmo quando os traços são sutis ou se sobrepõem a elementos da cena.

A Figura 4, adaptada de [24], ilustra o funcionamento interno do bloco recorrente. Observa-se que a imagem $\mathbf{O}_s$, que é a imagem recuperada das camadas recorrentes anteriores,

passa por sucessivos estágios recorrentes, nos quais é processada e comparada com a imagem "limpa" correspondente dada na entrada da rede, i.e. a imagem **B**. Dessa forma, a RESCAN não busca reconstruir diretamente a imagem sem chuva; em vez disso, ela aprende a prever a camada de chuva, ou seja $\hat{\mathbf{R}}_s$, que é posteriormente subtraída da entrada para obter a imagem limpa, ou seja, $\mathbf{O}_s - \hat{\mathbf{R}}_s$. Como resultado, a saída final $\mathbf{B}_s$ tende a preservar com maior fidelidade os contornos e as texturas da cena original. Após, a imagem $\mathbf{B}_s$ é encaminhada para a próxima camada recorrente, sendo, portanto, $\mathbf{O}_{s+1} = \mathbf{B}_s$.

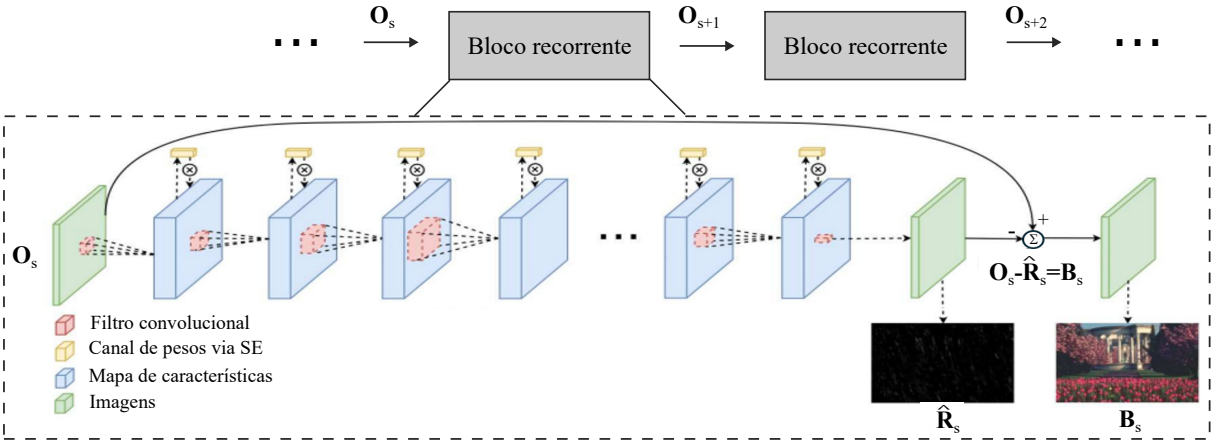


Figura 4: Ilustração de um bloco recorrente da arquitetura RESCAN. $\mathbf{O}_s$, $\hat{\mathbf{R}}_s$ e $\mathbf{B}_s$ representam, respectivamente, a imagem observada, as rais de chuva e a imagem após o processo de *deraining* na etapa **s**. SE representa a rede.

2.3 Construção do *dataset* para os experimentos

Para a construção do *dataset*, foi desenvolvido um procedimento computacional capaz de reproduzir o efeito visual provocado pela chuva nas imagens. Tal procedimento permite o controle explícito de variáveis relevantes do fenômeno, como inclinação, transparência, comprimento e largura das raias de chuva. O modelo das perturbações introduzidas pelo efeito da chuva na iteração t, denotado por R^t, é definido como

$$R^t = \sum_{i=1}^{N^t} G(P_i, s, l, w) \tag{2.2}$$

onde N^t representa o número de gotas na iteração t e $G(\cdot)$ é uma matriz esparsa que caracteriza o modelo geométrico de uma gota de chuva posicionada na imagem. O termo $P_i = (x_i, y_i)$ corresponde às coordenadas de origem da i-ésima gota no plano da imagem, $s \in [-s_{\max}, s_{\max}]$ denota a inclinação da gota, e l e w representam, respectivamente, o comprimento e a largura das raias.

A chuva sintética é gerada por meio da inserção estocástica dessas estruturas lineares inclinadas sobre a imagem original. O número de gotas é ajustado iterativamente, de modo a controlar o nível de degradação introduzido. O código completo da implementação é apresentado no Apêndice A. Ao longo do processo de desenvolvimento, foram conduzidos experimentos com distintas configurações de parâmetros, mantendo-se aquelas que, sob análise qualitativa, produziram resultados visualmente mais condizentes com o padrão de chuva almejado.

Considere agora I a imagem original (sem chuva), $\tilde{I}^t$ a imagem com chuva na iteração t, e $\tilde{I}$ a imagem final obtida ao término do processo. Seja ainda $\alpha \in [0{,}4; 0{,}5]$ um parâmetro que modela a transparência das gotas. A imagem degradada na iteração t é obtida por meio de uma combinação convexa entre a imagem original e o padrão de chuva modelado em (2.2), isto é,

$$\tilde{I}^t = \alpha I + (1-\alpha)R^t. \tag{2.3}$$

O processo é realizado iterativamente até que o nível de perturbação desejado seja atingido, considerando uma tolerância previamente estabelecida. Para quantificar o grau de degradação introduzido, utiliza-se a métrica PSNR (*Peak Signal-to-Noise Ratio*) [35], calculada em relação à imagem original com profundidade de 8 bits por canal, sendo representadas no intervalo discreto [0,255], isto é

$$\mathrm{PSNR}^t = 10 \cdot \log\left(\frac{255^2}{\mathrm{MSE}(I, \tilde{I}^t)}\right), \tag{2.4}$$

onde $\mathrm{MSE}(I, \tilde{I}^t)$ é o erro quadrático médio definido por

$$\mathrm{MSE}(I, \tilde{I}^t) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(I_{i,j} - \tilde{I}^t_{i,j}\right)^2, \tag{2.5}$$

em que m e n denotam as dimensões espaciais da imagem, e $I_{i,j}$ e $\tilde{I}^t_{i,j}$ representam, respectivamente, os valores dos *pixels* da imagem original e da imagem com chuva na posição (i, j). A métrica PSNR quantifica a distorção entre duas imagens. Dessa forma, valores mais elevados indicam menor degradação em relação à imagem original. No contexto deste trabalho, o valor alvo de PSNR determina o nível de severidade da chuva sintética, permitindo a construção de cinco grupos distintos, de 12, 14, 16, 18 e 20 dB. Cada categoria no conjunto de treinamento contém 1254 imagens, totalizando 6270 amostras. O conjunto de teste é composto por 100 imagens por categoria, totalizando 500 amostras. A Figura 5 ilustra uma amostra de cada categoria de imagens perturbadas provenientes do conjunto de teste. A separação do conjunto

de validação foi realizada de forma aleatória durante o processo de treinamento, reservando-se 30% do conjunto de treinamento para esse fim. Ressalta-se que todas as imagens utilizadas são geradas exclusivamente para este estudo e serão futuramente disponibilizadas à comunidade científica.



Figura 5: Exemplos mostrando diferentes níveis de perturbação por raias de chuva.

2.4 Desempenho da RESCAN sob Diferentes Níveis de Degradação

Esta seção dedica-se à análise do impacto de diferentes níveis de perturbação, ocasionados pelo efeito da chuva, sobre o desempenho da arquitetura RESCAN. Para tal, realiza-se uma comparação sistemática entre três conjuntos de dados: as imagens originais (sem chuva), suas versões degradadas (com chuva) e as versões resultantes do processo de *deraining*. Ressalta-se que os experimentos foram implementados em *Python*, utilizando-se majoritariamente a biblioteca *PyTorch* para a execução do modelo de *deraining*, em ambiente de computação remota

disponibilizado pelo *Google Colaboratory*, o qual forneceu acesso a uma unidade de processamento gráfico *Tesla T4*.

A arquitetura do modelo RESCAN foi implementada conforme as configurações sugeridas na literatura e os pesos do modelo foram treinados utilizando, para isso, o conjunto de dados construído conforme a seção anterior. Para essa análise, foram utilizadas métricas de qualidade perceptual, a saber: PSNR, dada em (2.4), e SSIM (*structural similarity index measure*) [36], definida como segue:

$$\mathrm{SSIM}(I,\tilde{I}) = \frac{(2\mu_I\mu_{\tilde{I}}+\varepsilon_1)(2\sigma_{I\tilde{I}}+\varepsilon_2)}{(\mu_I^2+\mu_{\tilde{I}}^2+\varepsilon_1)(\sigma_I^2+\sigma_{\tilde{I}}^2+\varepsilon_2)} \tag{2.6}$$

onde μ_I e $\mu_{\tilde{I}}$ denotam, respectivamente, as médias de I e $\tilde{I}$; σ_I e $\sigma_{\tilde{I}}$ são as variâncias de I e $\tilde{I}$, respectivamente; $\sigma_{I\tilde{I}}$ representa a covariância entre I e $\tilde{I}$; e ε_1 e ε_2 são parâmetros de regularização para prevenir divisão por zero. A métrica SSIM mensura a preservação da estrutura da imagem, sendo que valores próximos de 1 sugerem elevada similaridade com a imagem de referência, refletindo alta fidelidade na reconstrução visual.

Na fase de teste do modelo, o processo de avaliação é configurado para a arquitetura RESCAN, salvando quatro imagens correspondentes às últimas quatro camadas recorrentes. Essas camadas são aqui denominadas #1, #2, #3 e #4, sendo a #4 a última camada do modelo. A Figura 6 mostra o histograma da PSNR das imagens após o processo de *deraining* para as diferentes camadas recorrentes consideradas. Verifica-se, dessa figura, que mesmo em um *dataset* com um nível maior de perturbações, como é o caso do *dataset* de 12 dB, as camadas #2, #3 e #4 possuem desempenho semelhante. Além disso, a Figura 7 representa uma amostra dessas imagens advindas do *dataset* com 12 dB de perturbação após o processo de *deraining*; subjetivamente constatamos que há pouca diferença entre as imagens. Por esse motivo, consideram-se aqui os resultados finais das 3 últimas camadas de recorrência.

Levando em conta o uso das métricas MSE e SSIM, as Tabelas 1 e 2 apresentam, respectivamente, os valores obtidos com essas métricas aplicadas ao conjunto das imagens com chuva (artificialmente adicionada) e ao conjunto das imagens após o processo de *deraining*. Observa-se que, para todas as camadas recorrentes consideradas, a aplicação da arquitetura RESCAN proporcionou um aumento substancial nos valores da SSIM e uma redução nos valores de MSE, quando comparados às imagens degradadas. Por exemplo, a SSIM média para o conjunto de imagens com 12 dB, nas diferentes camadas recorrentes, aumentou de 0,3214 para no mínimo 0,9702, enquanto o MSE foi reduzido de 3805,97 para no máximo 14,48. Além

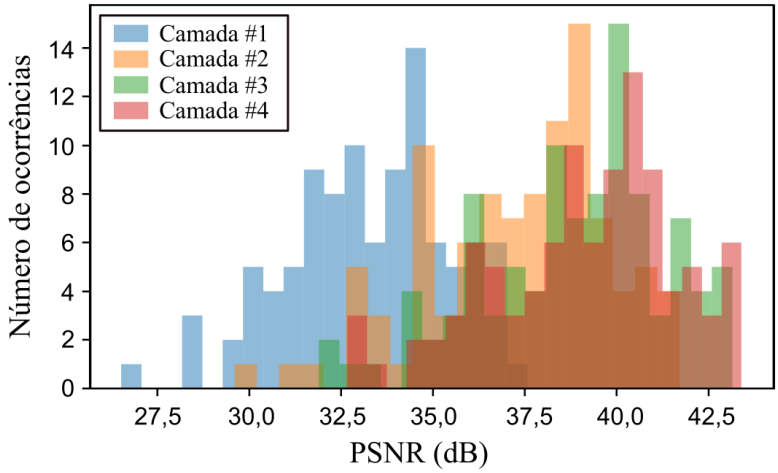


Figura 6: Histograma dos valores da PSNR após o processo de *deraining*, considerando o conjunto de dados de teste com perturbação de 12 dB, para as camadas recorrentes consideradas.



Figura 7: Exemplos mostrando imagens após o processo de *deraining*; imagens que antes possuíam perturbações de 12 dB advindas das 4 últimas camadas de recorrência.

do mais, através dos resultados das métricas consideradas, constata-se o comportamento consistente da arquitetura RESCAN para os diferentes níveis de perturbação. Isso é evidenciado por valores progressivamente mais altos da SSIM e menores de MSE. Outro ponto relevante a ser observado é a estabilidade dos resultados entre as diferentes camadas recorrentes. Como conclusão, constata-se que a arquitetura RESCAN é eficaz e robusta para a tarefa de restauração de imagens degradadas por chuva com diferentes níveis de perturbação.

Tabela 1: Avaliação das Imagens com Chuva Artificial

PSNR (dB)	**MSE**	**SSIM**
12	3805,97	0,3214
14	2524,10	0,3512
16	1607,22	0,4169
18	1025,18	0,5051
20	659,98	0,6016

Tabela 2: Avaliação do Processo de *Deraining*

PSNR (dB)	**Camada #2**		**Camada #3**		**Camada #4**	
	MSE	**SSIM**	**MSE**	**SSIM**	**MSE**	**SSIM**
12	14,48	0,9702	10,42	0,9785	9,55	0,9802
14	11,39	0,9726	8,67	0,9797	8,16	0,9811
16	8,96	0,9778	6,91	0,9832	6,54	0,9843
18	6,53	0,9834	5,10	0,9871	4,84	0,9878
20	4,87	0,9875	3,78	0,9903	3,58	0,9908

2.5 Conclusões

Neste capítulo, foi detalhado o processo de remoção de chuva (*deraining*) em imagens utilizando a RESCAN, evidenciando suas principais características estruturais e o modo como a arquitetura recorrente atua na identificação e supressão progressiva das raias de chuva. Além disso, foi descrito o processo de construção de um *dataset* sintético voltado ao treinamento e avaliação de métodos de *deraining*. Esse conjunto de dados foi gerado a partir da combinação de imagens limpas com modelos matemáticos de chuva, resultando em amostras com diferentes níveis de degradação e complexidade. A apresentação de exemplos representativos do *dataset* permitiu visualizar as variações nos padrões de chuva e a diversidade de cenas simuladas, assegurando a adequação do conjunto para o treinamento supervisionado da RESCAN. Por fim, os experimentos realizados com o *dataset* proposto demonstraram que a RESCAN apresentou

desempenho consistente na remoção de chuva, especialmente em cenários onde as raias são finas ou parcialmente sobrepostas aos elementos da cena. Os resultados quantitativos e qualitativos indicaram que o modelo é capaz de preservar contornos e texturas relevantes, reduzindo de forma significativa a interferência causada pela chuva. Esses achados reforçam a eficácia da arquitetura RESCAN e validam a utilidade do *dataset* sintético desenvolvido como ferramenta de avaliação controlada.

Os resultados obtidos neste capítulo estabeleceram a base para as análises posteriores, nas quais serão investigadas abordagens complementares ao *deraining*, como a avaliação do impacto do pré-processamento em tarefas subsequentes, a exemplo da segmentação semântica e da detecção de objetos em ambientes degradados.

Capítulo 3

Impacto do *Deraining* como Etapa de Pré-Processamento

3.1 Introdução

Este capítulo tem como objetivo avaliar o impacto da aplicação do modelo RESCAN como etapa de pré-processamento às tarefas de detecção de objetos e segmentação semântica. Serão avaliadas as variações de desempenho dos algoritmos nas tarefas consideradas ao processar imagens com chuva, sem chuva e com chuva removida pela RESCAN. Todos os experimentos aqui detalhados foram realizados com a mesma infraestrutura definida na seção anterior.

Ressalta-se que, em aplicações práticas, o acesso a imagens totalmente livres do efeito da chuva constitui uma situação idealizada e, portanto, de difícil ocorrência. Assim, essas imagens sem chuva são consideradas apenas como referência para propósito de comparação. O objetivo central é verificar em que medida o processo de *deraining* possibilita resultados de detecção próximos a esse cenário de referência, de modo que sua atuação seja imperceptível aos detectores de objetos. Em outras palavras, busca-se compreender se o pré-processamento de imagens com chuva contribui efetivamente para o aprimoramento do desempenho das etapas subsequentes de detecção ou, ao contrário, se introduz distorções que prejudicam o reconhecimento semântico, problemática também abordada em [13], em contexto semelhante ao do *deraining*. Dessa forma, procura-se avaliar se a reconstrução realizada pela RESCAN, embora concebida para remover ruídos atmosféricos, pode gerar artefatos visuais que comprometam a eficácia das tarefas de detecção de objetos e/ou segmentação semântica.

3.2 Integração do Processo de *Deraining* com Detecção de Objetos

Para a etapa de detecção de objetos, utilizou-se os modelos *Fast* R-CNN pré-treinados no *dataset* COCO [37] e YOLO versão 8 nano (YOLOv8n), também pré-treinada no *dataset* COCO. O modelo *Fast* R-CNN trata-se de um detector de objetos baseado em duas etapas (*two-stage detector*), no qual inicialmente são propostas regiões de interesse e, em seguida, refinadas por uma rede de classificação e regressão. Esse algoritmo de detecção costuma oferecer maior acurácia em tarefas de detecção quando comparada a métodos de etapa única, embora apresente maior custo computacional e, consequentemente, menor velocidade de inferência. Em contraste, o modelo YOLOv8n segue a abordagem de detecção em etapa única. Essa estratégia prioriza a velocidade de processamento, permitindo inferências em tempo real, ainda que, por se tratar da variante "nano", apresente menor capacidade de representação e, portanto, acurácia inferior em relação a arquiteturas mais complexas como o *Fast* R-CNN. Por outro lado, o YOLOv8n destaca-se em cenários que demandam baixa latência e execução em dispositivos com recursos limitados, enquanto o *Fast* R-CNN tende a ser mais adequado quando a acurácia da detecção é fator prioritário.



(a) (b) (c)

Figura 8: Ilustração da utilização da rede de detecção de objetos YOLOv8n em diferentes imagens. (a) Imagem original, isto é, antes da adição de chuva. (b) Imagem com chuva artificial. (c) Imagem após o processo de *deraining*.

Para esta avaliação, comparam-se os resultados das detecções obtidos nas imagens sem chuva (imagens originais), com chuva (imagens artificialmente perturbadas) e pré-processadas utilizando o algoritmo de *deraining*, como ilustrado na Figura 8. Foram consideradas as imagens geradas após o processo de *deraining* aplicado nas três últimas camadas recorrentes do modelo RESCAN (camadas #2, #3 e #4), conforme descrito anteriormente. Na sequência, são utilizadas as métricas *Precision* (3.1), *Recall* (3.2) e F1-*score* (3.3) para comparar o desempenho dos modelos selecionados. Para fins deste estudo, as imagens sem chuva são processadas

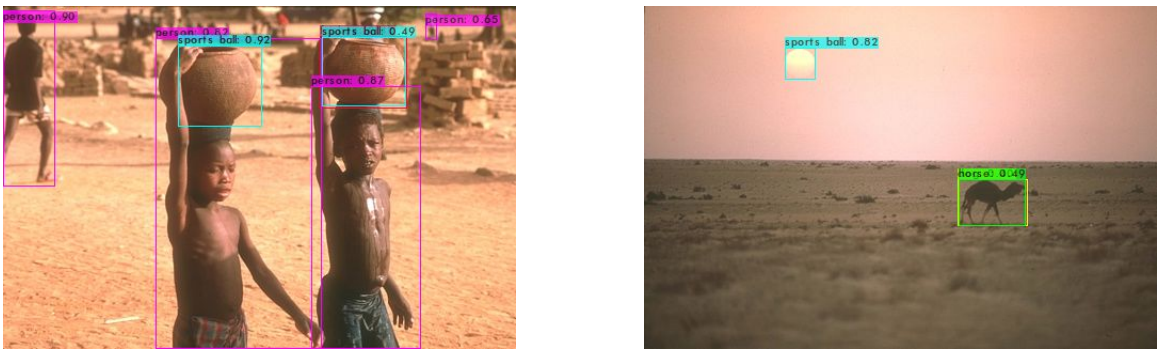
pelo modelo de *deraining* e servem como referência para a formulação das categorias *true positive* (TP), *false positive* (FP) e *false negative* (FN), utilizadas no cálculo das três métricas consideradas, enunciadas a seguir:

$$Precision = \frac{\text{TP}}{\text{TP + FP}}, \tag{3.1}$$

$$Recall = \frac{\text{TP}}{\text{TP + FN}}, \tag{3.2}$$

$$\text{F1-}score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.3}$$

Observou-se que, mesmo nas imagens originais, sem qualquer tipo de perturbação induzido pela chuva, alguns objetos são identificados de forma incorreta. Esse fato decorre de limitações intrínsecas aos próprios modelos de detecção de objetos selecionados, que, em determinados casos, realizam uma catalogação equivocada, atribuindo rótulos incorretos a elementos da cena (veja Figura 9). É importante ressaltar que tais inconsistências não estão relacionadas ao processo de degradação das imagens ou às etapas de *deraining*, mas sim à capacidade dos modelos de reconhecimento em distinguir corretamente as classes de interesse. Assim, os erros de classificação observados são tratados neste estudo como resultados aceitáveis, uma vez que o foco da pesquisa não está em aprimorar ou recalibrar os modelos de detecção, mas em avaliar o impacto da remoção da chuva sobre seu desempenho. Melhorias nesse aspecto estão fora do escopo da presente investigação.

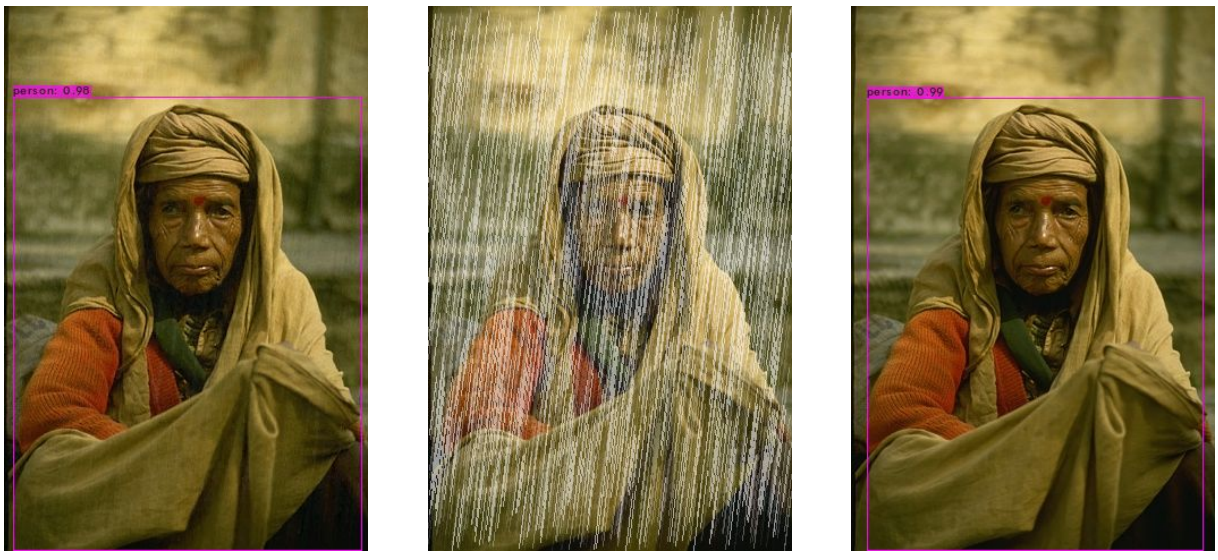


(a) (b)

Figura 9: Exemplos de falhas de classificação do modelo YOLOv8n. (a) Classificação de um vaso como bola de esporte. (b) Classificação da Lua como uma bola de esporte e um camelo como um cavalo.

Em contrapartida, também notou-se que, após o processo de *deraining*, o modelo

apresentou, em algumas imagens, maior confiança na identificação dos objetos, como ilustrado na Figura 10. Esse comportamento pode estar associado a diferentes fatores. Uma primeira hipótese é que a remoção das gotas de chuva e dos borramentos provocados pela precipitação reduziu a quantidade de ruído visual, resultando em contornos mais nítidos e padrões texturais mais consistentes, o que favorece a extração de características discriminativas. Fato é, o *deraining* pode ter atuado como uma forma de realce de bordas e contrastes locais, produzindo imagens que, embora diferentes da original, se aproximem mais das representações que o modelo aprendeu durante o treinamento [38]. Adicionalmente, pode-se levantar a hipótese de que a rede de *deraining* tenha introduzido pequenas alterações estruturais na imagem, por exemplo, suavização de áreas não homogêneas ou redução de detalhes irrelevantes, as quais eliminaram informações que poderiam confundir o detector. Assim, o aumento da confiança na classificação, em determinados casos, não estaria apenas relacionado à recuperação da cena degradada, mas também ao fato de o *deraining* gerar uma representação mais próxima de um cenário idealizado, simplificado em relação ao ruído do mundo real. Essa observação abre espaço para investigações futuras, no sentido de avaliar se a melhoria de confiança corresponde necessariamente a uma melhoria na acurácia, ou se pode, em alguns casos, refletir um viés introduzido pelo processo de *deraining*.



(a) (b) (c)

Figura 10: Ilustração da utilização da detecção de objetos pelo modelo YOLOv8n. (a) Imagem original, ou seja, antes da adição de chuva, identificada como pessoa com 98% de confiança. (b) Imagem com 12 dB de perturbação de chuva, sem qualquer identificação. (c) Imagem após o pré-processamento de *deraining*, identificada como pessoa com 99% de confiança

Os resultados obtidos com os modelos YOLOv8n e *Fast* R-CNN permitem discutir

de forma comparativa o impacto da degradação causada pela chuva, bem como a melhoria de desempenho do detector de objetos após as imagens serem restauradas. De maneira geral, como sugere a Tabela 3, ambos os modelos apresentaram queda expressiva nas métricas de detecção (*Precision*, *Recall* e *F1-score*) na presença de chuva, sobretudo em níveis mais altos de ruído (PSNR na faixa de 12–14 dB). Esse comportamento evidencia a vulnerabilidade dos detectores de objetos a perturbações visuais, confirmando que a precipitação constitui um fator crítico na redução da acurácia dos sistemas de visão computacional. A aplicação do *deraining* mostrou-se determinante para a recuperação do desempenho em ambas as arquiteturas, como evidenciam as Tabelas 4 e 5, elevando os indicadores a patamares próximos aos observados em condições sem chuva. No modelo YOLOv8n, por exemplo, verificou-se um aumento de F1-*score* de 0,241 para 0,953 na camada #2 em 12 dB, enquanto que o modelo *Fast* R-CNN no mesmo cenário resultou em crescimento de 0,313 para 0,929. Esses resultados reforçam a hipótese de que a remoção da chuva atua como pré-processamento essencial, mitigando as falhas de detecção associadas à degradação visual.

Tabela 3: Comparação dos resultados de detecção dos modelos YOLOv8n e *Fast* R-CNN com diferentes níveis de PSNR, sob chuva artificial

PSNR (dB)	**YOLOv8n**			***Fast* R-CNN**		
	Precision	**Recall**	**F1-*score***	**Precision**	**Recall**	**F1-*score***
12	0,392	0,174	0,241	0,367	0,273	0,313
14	0,439	0,252	0,320	0,429	0,330	0,373
16	0,562	0,357	0,436	0,526	0,498	0,511
18	0,590	0,426	0,495	0,530	0,551	0,540
20	0,728	0,513	0,602	0,630	0,555	0,590

Tabela 4: Resultados de detecção do modelo YOLOv8n nas camadas #2, #3 e #4 com diferentes níveis de PSNR, após o processo de *deraining*

PSNR (dB)	**Camada #2**			**Camada #3**			**Camada #4**		
	Precision	**Recall**	**F1-*score***	**Precision**	**Recall**	**F1-*score***	**Precision**	**Recall**	**F1-*score***
12	0,941	0,965	0,953	0,940	0,957	0,948	0,922	0,930	0,926
14	0,889	0,974	0,929	0,949	0,965	0,957	0,956	0,948	0,952
16	0,966	0,983	0,974	0,966	0,974	0,970	0,966	0,983	0,974
18	0,957	0,965	0,961	0,973	0,957	0,965	0,982	0,965	0,974
20	0,949	0,965	0,957	0,966	0,974	0,970	0,933	0,965	0,949

Tabela 5: Resultados de detecção do modelo *Fast* R-CNN nas camadas #2, #3 e #4 com diferentes níveis de PSNR, após o processo de *deraining*

PSNR (dB)	**Camada #2**			**Camada #3**			**Camada #4**		
	Precision	**Recall**	**F1-*score***	**Precision**	**Recall**	**F1-*score***	**Precision**	**Recall**	**F1-*score***
12	0,933	0,925	0,929	0,934	0,930	0,932	0,935	0,952	0,943
14	0,917	0,930	0,923	0,908	0,912	0,910	0,929	0,925	0,927
16	0,915	0,952	0,933	0,910	0,934	0,922	0,900	0,947	0,923
18	0,917	0,930	0,923	0,934	0,943	0,939	0,951	0,943	0,947
20	0,960	0,947	0,953	0,947	0,947	0,947	0,951	0,947	0,949

Entretanto a análise comparativa revela diferenças importantes entre os modelos. O YOLOv8n demonstrou maior sensibilidade em cenários extremos de degradação, como sugere a Tabela 3, apresentando valores bastante reduzidos em presença de chuva. Por outro lado, também foi o modelo que mais se beneficiou do processo de *deraining*, alcançando ganhos mais expressivos após a remoção do ruído. Já o modelo *Fast* R-CNN exibiu maior robustez relativa em níveis intermediários de ruído (16–18 dB), mantendo desempenho levemente superior ao YOLOv8n antes do pré-processamento, mas com ganhos menos acentuados após o *deraining*. Essa distinção sugere que arquiteturas de natureza *one-stage*, como o modelo YOLO, são mais vulneráveis à degradação, mas apresentam maior capacidade de recuperação quando a qualidade da imagem é restaurada. Já arquiteturas *two-stage*, como o *Fast* R-CNN, preservam melhor a robustez inicial, ainda que com menor responsividade aos ganhos do pré-processamento.

No que diz respeito à análise por camadas, verificou-se um padrão semelhante entre os modelos considerados. Nos cenários degradados, as camadas intermediárias (#2, #3 e #4) apresentaram desempenhos comparáveis, indicando que a degradação afeta de maneira homogênea as diferentes profundidades de representação. Contudo, após o *deraining*, observou-se uma vantagem sistemática das camadas mais profundas (#4) em níveis mais altos de ruído (18–20 dB), atingindo os maiores F1-*scores* em ambos os modelos (0,974 para o modelo YOLOv8n e 0,947 para o modelo *Fast* R-CNN). Essa tendência sugere que representações mais profundas conseguem consolidar e refinar a informação quando a qualidade da imagem é recuperada, potencializando a detecção de objetos.

3.3 Impacto do Uso de *Deraining* na Segmentação Semântica

No campo da visão computacional, diversos modelos vêm sendo propostos para tarefas de segmentação de objetos. Dentre os mais utilizados encontram-se o *Mask* R-CNN [27], baseado em redes convolucionais profundas. O modelo YOLOv8 [33] pertencente à família de modelos *you only look once*, cuja proposta é priorizar eficiência e velocidade sem comprometer significativamente a acurácia. Ambos apresentam abordagens distintas para a resolução do mesmo problema, o que justifica uma análise comparativa. O modelo *Mask* R-CNN é uma extensão do *Fast* R-CNN, incorporando uma ramificação adicional responsável pela predição de máscaras de segmentação para cada objeto detectado. Sua arquitetura é composta por uma *backbone* convolucional, comumente a ResNet-50 combinado ao *feature pyramid network* (FPN) [39], que extrai representações multi-escala das imagens. A partir disso, a rede gera regiões de interesse (RoIs) e aplica refinamentos sucessivos para detecção de *bounding boxes*, classificação de categorias e previsão de máscaras binárias. Apesar de sua alta acurácia em tarefas de segmentação por *pixel* (*instance-level*), o modelo *Mask* R-CNN apresenta um custo computacional elevado, demandando maior poder de processamento e tempo de inferência mais longo quando comparado a modelos de última geração.

Por outro lado, o modelo YOLOv8 representa a evolução de uma família de arquiteturas caracterizada por sua abordagem de detecção em tempo real. Em contraste ao *Mask* R-CNN, o YOLOv8 emprega uma arquitetura unificada de detecção e segmentação, em que *bounding boxes* e máscaras são previstas de forma conjunta e direta. Essa estratégia confere ao modelo uma eficiência significativamente maior, com tempos de inferência reduzidos, o que o torna

mais adequado para aplicações em cenários que exijam processamento em tempo real, como sistemas de monitoramento, robótica e veículos autônomos. Entretanto, embora exiba adequados níveis de desempenho, pode apresentar limitações em relação à granularidade e precisão das máscaras quando comparado a arquiteturas mais complexas, como o *Mask* R-CNN.

Nesta seção, consideraremos os modelos anteriormente discutidos e, além das métricas já consideradas, analisamos a métrica *intersection over union* (IoU) [40]. Ela mede a sobreposição entre a região prevista pelo modelo e a região de referência (*ground truth*), quantificando a proporção de área comum em relação à área total coberta pelas duas regiões. Quanto maior o valor da IoU, melhor a qualidade da predição, sendo que um valor igual a 1 indica sobreposição perfeita. Matematicamente, a IoU é definida como

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{3.4}$$

em que A representa a área da máscara de predição, B corresponde à área da máscara de referência, $|A \cap B|$ é a área da interseção entre elas e $|A \cup B|$ é a área da união. Assim, a IoU expressa a fração de coincidência espacial entre as duas regiões avaliadas.

A comparação dos resultados das segmentações semânticas das arquiteturas consideradas evidencia diferenças importantes no comportamento frente ao ruído atmosférico e ao processo de restauração aqui comentado, o *deraining*. Nos cenários com chuva artificial, ambos os modelos apresentaram degradação significativa, mas com padrões distintos. A Tabela 6 indica que o YOLOv8 possui maior sensibilidade à intensidade do ruído, exibindo métricas muito baixas em baixos níveis de perturbação (12 dB: F1-*score* = 0,067, IoU = 0,056), com recuperação progressiva apenas em níveis mais elevados (20 dB: F1-*score* = 0,313, IoU = 0,278). Já o *Mask* R-CNN, como observado na Tabela 6, embora também impactado, manteve valores mais estáveis ao longo dos diferentes níveis de perturbação, oscilando em torno de F1-*score* = 0,122-0,136 e IoU = 0,102-0,118. Essa estabilidade sugere que a abordagem *two-stage*, baseada em *region proposals*, apresente maior resiliência inicial ao ruído, mesmo que com desempenho global reduzido.

Tabela 6: Avaliação da segmentação semântica dos modelos YOLOv8 e *Mask* R-CNN no conjunto de dados com chuva artificial

PSNR (dB)	**YOLOv8**				***Mask* R-CNN**			
	Precision	**Recall**	**F1-*score***	**IoU**	**Precision**	**Recall**	**F1-*score***	**IoU**
12	0,092	0,061	0,068	0,056	0,148	0,144	0,127	0,108
14	0,153	0,120	0,125	0,105	0,136	0,146	0,122	0,102
16	0,187	0,146	0,154	0,138	0,149	0,145	0,132	0,114
18	0,229	0,210	0,210	0,191	0,139	0,145	0,127	0,109
20	0,361	0,298	0,314	0,278	0,149	0,150	0,136	0,118

Com a aplicação do *deraining*, o cenário se inverte. O YOLOv8 apresentou ganhos muito mais expressivos, como mostra a Tabela 7, alcançando F1-*scores* superiores a 0,74 em todos os níveis de perturbação e IoU próximo a 0,76, com variações mínimas entre as camadas avaliadas. Em contraste, o *Mask* R-CNN obteve apenas ganhos marginais após a remoção da chuva (Tabela 8), mantendo métricas em torno de F1-*score* = 0,134-0,141 e IoU = 0,126-0,134. Isto é, enquanto o modelo YOLO apresentou um salto superior a 500% em relação ao conjunto degradado, o *Mask* R-CNN não demonstrou evolução significativa.

Tabela 7: Avaliação da segmentação semântica do modelo YOLOv8 no conjunto de dados após o processo de *deraining* nas diferentes camadas selecionadas

PSNR (dB)	**Camada #2**				**Camada #3**				**Camada #4**			
	Precision	**Recall**	**F1-*score***	**IoU**	**Precision**	**Recall**	**F1-*score***	**IoU**	**Precision**	**Recall**	**F1-*score***	**IoU**
12	0,760	0,768	0,757	0,754	0,737	0,753	0,738	0,741	0,767	0,779	0,768	0,764
14	0,745	0,762	0,744	0,745	0,748	0,748	0,741	0,733	0,754	0,758	0,748	0,745
16	0,761	0,779	0,763	0,761	0,759	0,771	0,758	0,756	0,757	0,764	0,753	0,751
18	0,761	0,773	0,759	0,756	0,755	0,778	0,759	0,765	0,758	0,783	0,764	0,768
20	0,760	0,774	0,762	0,761	0,746	0,771	0,752	0,761	0,744	0,771	0,751	0,762

Tabela 8: Avaliação da segmentação semântica do modelo *Mask* R-CNN no conjunto de dados após o processo de *deraining* nas diferentes camadas selecionadas

PSNR (dB)	Camada #2				Camada #3				Camada #4			
	Precision	**Recall**	**F1-*score***	**IoU**	**Precision**	**Recall**	**F1-*score***	**IoU**	**Precision**	**Recall**	**F1-*score***	**IoU**
12	0,142	0,145	0,136	0,127	0,144	0,147	0,139	0,131	0,145	0,150	0,141	0,133
14	0,144	0,146	0,138	0,130	0,142	0,144	0,136	0,128	0,141	0,144	0,136	0,128
16	0,143	0,146	0,139	0,130	0,142	0,146	0,138	0,130	0,146	0,148	0,141	0,133
18	0,139	0,143	0,135	0,127	0,138	0,143	0,134	0,126	0,140	0,144	0,136	0,128
20	0,145	0,146	0,139	0,131	0,148	0,148	0,141	0,134	0,143	0,143	0,137	0,129

Estes resultados indicam que, embora o *Mask* R-CNN seja mais estável em condições adversas, ele é menos capaz de aproveitar os benefícios proporcionados pela restauração. O YOLOv8, por sua vez, é altamente dependente da qualidade da entrada, sofrendo forte degradação com o efeito da chuva, mas alcançando um desempenho muito bom após o pré-processamento. Essa diferença pode estar relacionada à forma como cada arquitetura lida com as representações intermediárias: no *Mask* R-CNN, a etapa de geração de *region proposals* pode manter redundâncias que limitam o ganho após o *deraining*, enquanto no YOLOv8 a extração direta de características globais é mais sensível à qualidade da entrada e, portanto, mais beneficiada pela remoção do ruído. Por fim, destaca-se que a camada #4 no processo de *deraining* forneceu os melhores resultados para o YOLOv8, sugerindo que representações mais profundas são essenciais para uma restauração eficaz em arquiteturas *one-stage*. Já no *Mask* R-CNN, as diferenças entre as camadas foram praticamente irrelevantes, reforçando sua baixa sensibilidade ao refinamento das imagens pré-processadas. Alguns resultados para análise subjetiva se encontram nas Figuras 11 e 12.



(a) (b) (c)

Figura 11: Ilustração da utilização da segmentação semântica pela rede *Mask* R-CNN. (a) Imagem original. (b) Imagem com 18 dB de perturbação de chuva. (c) Imagem após o pré-processamento de *deraining* advinda de uma perturbação de 12 dB.



(a) (b) (c)

Figura 12: Ilustração da utilização da segmentação semântica pelo modelo *Mask* R-CNN. (a) Imagem original. (b) Imagem com 18 dB de perturbação de chuva. (c) Imagem após o pré-processamento de *deraining* advinda de uma perturbação de 12 dB.

3.4 Conclusões

Neste capítulo, foi apresentada uma análise abrangente sobre o impacto do processo de *deraining* — por meio da arquitetura RESCAN — no desempenho de diferentes modelos de detecção de objetos e segmentação semântica. Buscou-se compreender em que medida a remoção de chuva influencia a robustez e a precisão desses algoritmos.

Inicialmente, a RESCAN foi integrada a dois modelos de detecção de objetos, permitindo avaliar o efeito do pré-processamento de remoção de chuva sobre a capacidade desses detectores em reconhecer e localizar instâncias em imagens afetadas. As análises demonstraram que o uso prévio da RESCAN tende a melhorar a acurácia das detecções, reduzindo falsos positivos e negativos, sobretudo em cenários de maior densidade de chuva. Essa constatação reforça a relevância de estratégias de restauração de imagem na etapa de pré-processamento, como forma de potencializar o desempenho de modelos de alto nível. Em seguida, a mesma abordagem foi aplicada a dois modelos de segmentação semântica, com o objetivo de investigar o efeito do processo de *deraining* sobre a consistência espacial e a preservação de fronteiras entre regiões segmentadas. Os resultados mostraram que, de modo geral, a aplicação da RESCAN contribui para a melhoria das métricas de sobreposição e uniformidade das classes, evidenciando uma recuperação mais fiel das estruturas de cena.

De forma geral, os experimentos abordados neste capítulo evidenciam que a etapa de *deraining* não apenas melhora a qualidade visual das imagens, mas também exerce influência direta sobre o desempenho de tarefas subsequentes de visão computacional. Assim, a associação entre a RESCAN e modelos de detecção e segmentação se mostra promissora, abrindo caminho para investigações mais aprofundadas sobre estratégias integradas de aprendizado conjunto entre restauração e interpretação de imagens.

Capítulo 4

Considerações Finais

Nesta dissertação, foi realizada uma investigação sobre o desempenho da arquitetura RESCAN no processo de remoção de chuva (*deraining*) em imagens, bem como sobre o impacto desse pré-processamento em tarefas subsequentes de visão computacional, especialmente nas áreas de detecção de objetos e segmentação semântica. O trabalho foi estruturado em dois capítulos principais, cada um abordando etapas complementares da pesquisa. Neste capítulo, trataremos das considerações finais do presente trabalho de pesquisa. Inicialmente, é realizada uma discussão sobre os principais e mais relevantes resultados obtidos. Em seguida, algumas sugestões para trabalhos futuros na área são apresentadas. Por fim, é listado o trabalho publicado decorrente da pesquisa desenvolvida.

4.1 Conclusões

No Capítulo 1, foi apresentado o funcionamento da arquitetura RESCAN, destacando seus principais componentes e discutindo seu papel na modelagem das estruturas de chuva. Em seguida, descreveu-se o processo de construção de um *dataset* sintético, desenvolvido a partir da combinação controlada de imagens limpas com padrões artificiais de chuva. Essa abordagem permitiu gerar diferentes níveis de degradação, criando um ambiente experimental reprodutível para o treinamento e avaliação do modelo. Por fim, o capítulo apresentou os resultados do desempenho da RESCAN nesse conjunto de dados, demonstrando sua eficácia na remoção progressiva das raias de chuva e na restauração de detalhes visuais.

No Capítulo 2, o foco voltou-se para a análise do impacto do *deraining* sobre o desempenho de modelos de alto nível. A arquitetura RESCAN foi integrada a dois modelos

de segmentação semântica e a dois modelos de detecção de objetos, possibilitando avaliar de forma comparativa como a remoção de chuva interfere na qualidade das previsões. Foram utilizadas métricas quantitativas consolidadas da literatura para mensurar o desempenho antes e após o pré-processamento, permitindo uma análise objetiva dos ganhos obtidos. Os resultados evidenciaram que a presença de chuva compromete significativamente o desempenho das diferentes arquiteturas avaliadas, sobretudo na precisão da detecção e na coerência espacial das segmentações. Por outro lado, o uso do *deraining* mostrou-se essencial para restaurar a eficácia desses modelos, resultando em melhorias consistentes nos indicadores de desempenho. De modo particular, observou-se que o modelo YOLOv8n apresentou maior sensibilidade às degradações atmosféricas; no entanto exibiu maior capacidade de recuperação após o processo de remoção de chuva. Em contraste, as arquiteturas baseadas em R-CNN (*Fast* R-CNN e *Mask* R-CNN) demonstraram maior robustez em cenários degradados. Essa diferença de comportamento revela uma relação de complementaridade entre os modelos: enquanto o modelo YOLOv8n se beneficia fortemente da etapa de restauração, os modelos R-CNN mantêm desempenho estável mesmo sob degradação.

Em síntese, os resultados obtidos ao longo desta dissertação evidenciam a importância do uso de técnicas de restauração de imagem, como o *deraining*, para aprimorar o desempenho de sistemas de visão computacional em ambientes adversos. Diante do exposto, acredita-se que os objetivos inicialmente estabelecidos foram alcançados com sucesso no desenvolvimento deste trabalho de pesquisa, evidenciando que a integração entre redes de remoção de chuva e modelos de detecção ou segmentação abre perspectivas promissoras para o desenvolvimento de soluções mais robustas, capazes de operar de forma confiável em condições atmosféricas variadas.

4.2 Sugestões para trabalhos futuros

Como proposta para trabalhos de pesquisa futuros, sugere-se:

- Investigar abordagens de *multi-task learning* que integrem a restauração (*deraining*) e a detecção/segmentação em um único modelo, permitindo que a rede aprenda representações otimizadas simultaneamente para “limpeza” e interpretação da cena.
- Expansão do *dataset* sintético, incorporando novos padrões de chuva, diferentes condições de iluminação, variação de profundidade e movimento de câmera, de modo a aproximar

o cenário sintético das condições reais de captura.

- Avaliar o desempenho da RESCAN e dos modelos subsequentes em bases de dados reais, contendo chuva natural, para verificar a generalização dos resultados obtidos em ambientes sintético.

- Confrontar o desempenho da RESCAN com outras redes analisando a relação entre complexidade arquitetural, tempo de inferência e qualidade da restauração.

4.3 Trabalho publicado oriundo da dissertação

Durante a realização deste trabalho de pesquisa, o seguinte artigo foi publicado:

- L. Fischer, R. Seara, E. Silva, ”Detecção de Objetos em Imagens Degradadas pelo Efeito da Chuva”, *XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2025)*, Natal - RN, Brasil, Out. 2025.

Apêndice A

Código para Geração de Chuva Sintética

Código A: Implementação do método de geração sintética de chuva com controle de PSNR

```python
import numpy as np
import cv2
from skimage.metrics import mean_squared_error as mse
from skimage.metrics import peak_signal_noise_ratio as psnr


def generate_random_lines(imshape, slant, drop_length, num_drops):
    drops = []
    for i in range(num_drops):
        if slant < 0:
            x = np.random.randint(-slant, imshape[1])
        else:
            x = np.random.randint(0, imshape[1] - slant)
        y = np.random.randint(0, imshape[0] - drop_length)
        drops.append((x, y))
    return drops


def add_rain(image, target_psnr, slant_extreme=10,
             drop_length=60, drop_width=2,
             num_drops=100, max_iterations=500):
```

```
22
23      imshape = image.shape
24      image_rainy = image.copy()
25      slant = np.random.randint(-slant_extreme, slant_extreme)
26
27      for _ in range(max_iterations):
28          rain_drops = generate_random_lines(
29              imshape, slant, drop_length, num_drops)
30
31          for rain_drop in rain_drops:
32              start_point = (rain_drop[0], rain_drop[1])
33              end_point = (rain_drop[0] + slant,
34                           rain_drop[1] + drop_length)
35              color = (255, 255, 255)
36              image_rainy = cv2.line(
37                  image_rainy,
38                  start_point,
39                  end_point,
40                  color,
41                  drop_width
42              )
43
44          transparency = (np.random.randint(40, 50)) / 100
45          cv2.addWeighted(
46              image_rainy,
47              1 - transparency,
48              image,
49              transparency,
50              0,
51              image_rainy
52          )
53
54          current_mse = mse(image, image_rainy)
55          current_psnr = psnr(image, image_rainy)
```

```
56
57          if (target_psnr - 0.2 <= current_psnr <= target_psnr + 0.2
58                  or current_psnr <= target_psnr - 0.2):
59              break
60
61          num_drops += 5
62
63      return image_rainy, current_mse, current_psnr
```

Referências Bibliográficas

[1] Z. Zhang, Y. Wei, H. Zhang, Y. Yang, S. Yan, and M. Wang, "Data-driven single image deraining: A comprehensive review and new perspectives," *Pattern Recognition*, vol. 143, Nov. 2023, art. no. 109740.

[2] M. Hnewa and H. Radha, "Object detection under rainy conditions for autonomous vehicles: A review of state-of-the-art and emerging techniques," *IEEE Signal Process. Mag.*, vol. 38, no. 1, pp. 53–67, Jan. 2021.

[3] P. Baiju and S. N. George, "An automated unified framework for video deraining and simultaneous moving object detection in surveillance environments," *IEEE Access*, vol. 8, pp. 128 961–128 972, 2020.

[4] S. Li, W. Ren, F. Wang, I. B. Araujo, E. K. Tokuda, R. H. Junior, R. M. Cesar-Jr, Z. Wang, and X. Cao, "A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives," *Int. J. Comput. Vis.*, vol. 129, pp. 1301–1322, 2021.

[5] H. Wang, Y. Wu, M. Li, Q. Zhao, and D. Meng, "Survey on rain removal from videos or a single image," *Science China Information Sciences*, vol. 65, no. 1, p. 111101, 2022.

[6] D. Ren, W. Shang, P. Zhu, Q. Hu, D. Meng, and W. Zuo, "Single image deraining using bilateral recurrent network," *IEEE Trans. Image Process.*, vol. 29, pp. 6852–6863, 2020.

[7] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1708–1716.

[8] J. Pan, J. Dong, Y. Liu, J. Zhang, J. Ren, J. Tang, Y. Tai, and M. Yang, "Physics-based generative adversarial models for image restoration and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. t, pp. 2449–2462, Jul. 2021.

[9] J. M. J. Valanarasu, R. Yasarla, and V. M. Patel, "Transweather: Transformer-based restoration of images degraded by adverse weather conditions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 2353–2363.

[10] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao, "Conditional variational image deraining," *IEEE Trans. Image Process.*, vol. 29, pp. 6288–6301, 2020.

[11] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1377–1393, 2020.

[12] G. Wang, C. Sun, and A. Sowmya, "Context-enhanced representation learning for single image deraining," *Int. J. Comput. Vis.*, vol. 129, pp. 1650–1674, 2021.

[13] Y. Pei, Y. Huang, Q. Zou, Y. Lu, and S. Wang, "Does haze removal help cnn-based image classification?" in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 682–697.

[14] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 12 270–12 279.

[15] J. Jin, A. Fatemi, W. M. P. Lira, F. Yu, B. Leng, R. Ma, A. Mahdavi-Amiri, and H. Zhang, "Raidar: A rich annotated image dataset of rainy street scenes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 2951–2961.

[16] H. Porav, T. Bruls, and P. Newman, "I can see clearly now: Image restoration via deraining," in *Proc. IEEE Int. Conf. Robotics Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 7087–7093.

[17] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1357–1366.

[18] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *20th International Conference on Pattern Recognition*, 2010, pp. 2366–2369.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[20] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind"image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.

[21] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[22] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[23] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.

[24] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 262–277.

[25] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[26] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[27] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[28] P. Poirson, P. Ammirato, C.-Y. Fu, W. Liu, J. Kosecká, and A. C. Berg, "Fast single shot detection and pose estimation," in *Proc. 4th Int. Conf. 3D Vision (3DV)*, 2016, pp. 676–684.

[29] K. Wang, T. Wang, J. Qu, H. Jiang, Q. Li, and L. Chang, "An end-to-end cascaded image deraining and object detection neural network," *IEEE Robot. Autom. Lett.*, pp. 9541–9548, 2022.

[30] Y. Wei, Z. Zhang, H. Zheng, R. Hong, Y. Yang, and M. Wang, "Sginet: Toward sufficient interaction between single-image deraining and semantic segmentation," in *Proc. 30th ACM Int. Conf. Multimedia (ACM MM)*. ACM, 2022, pp. 6202–6210.

[31] M. Guo, M. Chen, C. Ma, Y. Li, X. Li, and X. Xie, "High-level task-driven single image deraining: Segmentation in rainy days," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*. Springer, 2020, pp. 350–362.

[32] K. Zhang, W. Luo, Y. Yu, W. Ren, F. Zhao, C. Li, L. Ma, W. Liu, and H. Li, "Beyond monocular deraining: Parallel stereo deraining network via semantic prior," *Int. J. Comput. Vis.*, pp. 1754–1769, 2022.

[33] C. H. Kang and S. Y. Kim, "Real-time object detection and segmentation technology: An analysis of the yolo algorithm," *JMST Adv.*, pp. 69–76, 2023.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[35] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, 2008.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[37] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 1209–1218.

[38] J. Gilg, T. Teepe, F. Herzog, and G. Rigoll, "The box size confidence bias harms your object detector," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 1471–1480.

[39] M. Ye, J. Ouyang, G. Chen, J. Zhang, and X. Yu, "Enhanced feature pyramid network for semantic segmentation," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 3209–3216.

[40] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.