

# Detecção de Objetos em Imagens Degradadas pelo Efeito da Chuva

Lara Fischer, Ênio dos Santos Silva e Rui Seara

**Resumo**— Em aplicações de visão computacional, a qualidade das imagens é essencial para o processo de detecção de objetos. No entanto, perturbações causadas pela chuva podem comprometer a precisão da tarefa em questão. Para mitigar tal impacto, uma etapa de pré-processamento baseada em um modelo de inteligência artificial é utilizada. Essa técnica é usualmente denominada na literatura por *deraining*. Este artigo investiga o desempenho de um dos modelos do estado-da-arte de *deraining* em imagens submetidas a diferentes níveis de perturbações. Os resultados do processo de *deraining* são avaliados através de um algoritmo de detecção de objetos, o qual visa verificar se as características semânticas relevantes da imagem são recuperadas. Resultados de simulação computacional corroboram a aplicabilidade do modelo de *deraining* considerado.

**Palavras-Chave**— Base de dados de imagem com chuva, *deraining*, detecção de objetos em condição de chuva.

**Abstract**— In computer vision applications, image quality is essential to the object detection process. However, disturbances caused by rain can compromise the accuracy of the task at hand. To mitigate such an impact, a preprocessing step based on an artificial intelligence model is used. This technique is usually referred to in the literature as *deraining*. This paper investigates the performance of one of the state-of-the-art models of *deraining* subjected to different levels of perturbations. The results of the *deraining* process are evaluated by using an object detector algorithm, which aims to verify whether the relevant semantic characteristics of the image are recovered. Computer simulation results corroborate the applicability of the *deraining* model considered.

**Keywords**— Rain dataset, single image *deraining*, rain detection.

## I. 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 Fig. 1. Esses eventos prejudicam a eficácia de algoritmos de análise de imagens, principalmente quando tratam de tarefas de alta complexidade, como direção autônoma [2], serviços de vigilância [3], dentre outras aplicações. Para mitigar esses efeitos, métodos de pré-processamento vêm sendo desenvolvidos com o objetivo de 'restaurar' a nitidez e a fidelidade visual das cenas degradadas [1], [4].

Lara Fischer, Ênio dos Santos Silva e Rui Seara, LINSE Laboratório de Circuitos e Processamento de Sinais, Departamento de Engenharia Elétrica e Eletrônica, Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brasil, e-mails: lara.fischer@linse.ufsc.br; enio@linse.ufsc.br; seara@linse.ufsc.br. Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

O processo de remoção de chuva em uma imagem, conhecido por *deraining* ou *single image deraining (SID)*, busca reduzir as interferências causadas por essas perturbações, aprimorando a qualidade das imagens com vistas à percepção humana e/ou perspectiva das máquinas, facilitando sua utilização em tarefas de visão computacional [2].

Na literatura, diferentes abordagens de *deraining* vêm sendo discutidas [5]. Dentre elas, destacam-se aquelas baseadas em aprendizado profundo, com modelos envolvendo redes recorrentes [6], redes convolucionais [1], redes generativas [7], dentre outros [1]. Essas arquiteturas vêm demonstrando avanços significativos na reconstrução de imagens degradadas; no entanto, em aplicações do mundo real, quando perturbações de chuva combinadas interferem na imagem, o desempenho dessas técnicas é substancialmente afetado [8].

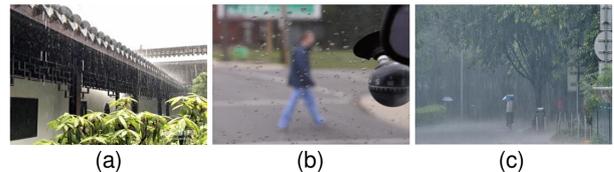


Fig. 1: Diferentes tipos de degradações provocadas pelo efeito da chuva. (a) Raias de chuva. (b) Gotas de chuva. (c) Chuva torrencial. (Figura adaptada de [1]).

Um modelo promissor recentemente evidenciado na literatura é o *RESCAN (recurrent squeeze-and-excitation context aggregation net)* [9]. 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). Aqui, é importante destacar que a presença da chuva oculta certas características visuais importantes, como bordas e texturas, comprometendo o desempenho de detectores tradicionais, como *faster regions with CNNs (Faster R-CNNs)*, *you only look once (YOLO)* e *single shot multi-box detector (SSD)* [10]. Para essas análises, um fator importante é o conjunto de dados utilizado. Ele deve conter imagens com e sem chuva da mesma cena [11], podendo manter atrelada a elas a informação de seus respectivos níveis de perturbações provocados pela chuva. Diante disso, propomos a construção de um *dataset*, com imagens com chuva sinteticamente adicionada, baseado nas imagens sem chuva do *dataset Rain100H* [12], adaptado especificamente para nossos objetivos experimentais.

Este estudo discute o desempenho da rede *RESCAN* sob diferentes níveis de perturbações introduzidas pelo efeito da chuva nas imagens. Neste trabalho, são também evidenciados os benefícios da integração do processo de *deraining* com o algoritmo de detecção de objetos, usado aqui o *YOLO-V4* (*you only look once*, versão 4) [13]. A eficácia da combinação proposta é verificada através dos resultados experimentais obtidos.

## II. METODOLOGIA

Nesta seção, são apresentados os principais passos de elaboração do *dataset* implementado, sendo levantados os aspectos relevantes da arquitetura da rede *RESCAN* e, na sequência, é discutido o uso dos modelos *RESCAN* e *YOLO-V4* nos processos propostos. Ressalta-se que todos os experimentos foram realizados na linguagem de programação *Python*, usando principalmente a biblioteca *PyTorch* (para o modelo de *deraining*) e o serviço de computação remota *Google Colaboratory*, que ofereceu uma máquina virtual *Tesla T4*.

### A. Dataset

Um dos fatores determinantes para o aprimoramento das técnicas de *deraining* está na escolha adequada do *dataset* utilizado para o treinamento do modelo. É a partir desses dados que a rede aprende a identificar e remover padrões de chuva presentes em imagens degradadas. Na literatura, há uma variedade abrangente de conjuntos de dados disponíveis. Esses conjuntos contêm imagens capturadas de ambientes naturais ou urbanos [14], com diferentes tipos de degradação [15], com imagens sintéticas [16] ou capturadas em condições reais de chuva [17], dentre outros [11]. No caso específico do treinamento da rede *RESCAN*, para realizar experimentos de forma ‘ideal’, é necessário um *dataset* composto por imagens pareadas — ou seja, capturas da mesma cena com e sem chuva. Por isso, é indicado que o treinamento da rede seja feito com base em um *dataset* com imagens com chuva sinteticamente adicionada. Contudo, os *datasets* disponíveis na literatura, em sua maioria, não contemplam uma separação estruturada que permita o treinamento de modelos em níveis diferentes de perturbações. Com base nesse aspecto, propõe-se a implementação de um *dataset* ad hoc, levando em conta as imagens sem chuva do *dataset Rain100H* [12].

Para a construção do *dataset*, é desenvolvida uma função que reproduz o efeito visual provocado pela chuva nas imagens. Essa função permite o controle de importantes variáveis para o processo, tais como inclinação, transparência e tamanho das raías de chuva. O modelo das perturbações causadas pelo efeito da chuva na iteração  $t$  é dado por

$$R^{(t)} = \sum_{i=1}^{100} G(P_i, s, 60, 2) \quad (1)$$

onde  $G(\cdot)$  caracteriza cada uma das 100 gotas de chuva, com  $P_i = (x_i, y_i)$  representando as coordenadas de origem das gotas no plano da imagem,  $s \in [-10, 10]$  denotando a inclinação da gota com 60 *pixels* de comprimento e 2 *pixels* de largura.

Considere agora  $I$  a imagem original (sem chuva),  $\tilde{I}^{(t)}$  a imagem com chuva na iteração  $t$ ,  $\tilde{I}$  a imagem resultante final do processo de adição de chuva e  $\alpha \in [0, 4; 0, 5]$  um parâmetro

modelando a transparência. A imagem com chuva é então uma associação (combinação convexa) entre a imagem original e as gotas modeladas por (1). Assim,

$$\tilde{I}^{(t)} = \alpha I + (1 - \alpha)R^{(t)}. \quad (2)$$

A função é aplicada iterativamente sobre as imagens até que o nível de perturbações desejado seja obtido ou atinja uma dada tolerância. Para tal, é usado como métrica de avaliação a *PSNR* (*peak signal-to-noise ratio*) [18] em relação à imagem original, i.e.,

$$\text{PSNR}^{(t)} = 10 \cdot \log \left[ \frac{255^2}{\text{MSE}(I, \tilde{I}^{(t)})} \right] \quad (3)$$

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

$$\text{MSE}(I, \tilde{I}^{(t)}) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (I_{i,j} - \tilde{I}_{i,j}^{(t)})^2 \quad (4)$$

com  $m$  e  $n$  denotando as dimensões da imagem, e  $I_{i,j}$  e  $\tilde{I}_{i,j}^{(t)}$  representando, respectivamente, os *pixels* da imagem original e sua correspondente imagem com chuva.

A métrica *PSNR* quantifica a distorção entre imagens. Assim, valores maiores após o processo de restauração indicam maior capacidade do modelo em lidar com os efeitos da chuva, ou seja, conjuntos de dados menos desafiadores para o processamento oriundo da rede.

As imagens geradas são então organizadas em cinco categorias distintas de intensidade de perturbações, correspondentes aos níveis 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, somando 500 amostras. A Fig. 2 ilustra uma amostra de cada categoria de imagens perturbadas advindas 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 este fim. Ressalta-se que todas as imagens utilizadas são geradas exclusivamente para este estudo e serão futuramente disponibilizadas à comunidade científica.



Fig. 2: Exemplos mostrando diferentes níveis de perturbações por raías de chuva. (de cima para baixo, da esquerda para direita) Imagem original e suas correspondentes imagens degradadas pela adição de diferentes níveis de perturbações, i.e., 20, 18, 16, 14 e 12 dB.

### B. Rede RESCAN

A rede *RESCAN* é composta por uma sequência de módulos convolucionais dilatados que ampliam o campo receptivo,

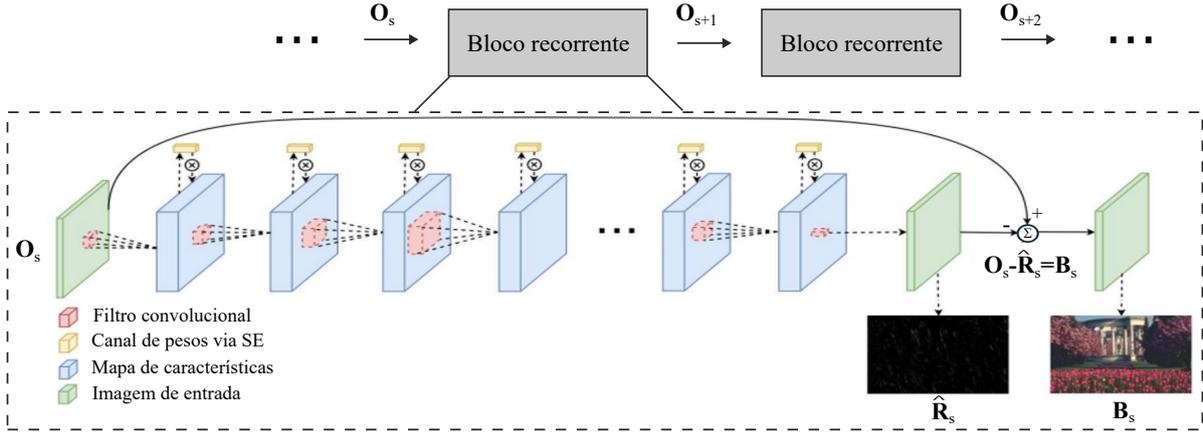


Fig. 3: Ilustração de um bloco recorrente da rede *RESCAN*.  $O_s$ ,  $\hat{R}_s$  e  $B_s$  representam, respectivamente, as imagens originais, as imagens com chuva e as imagens após o processo de *deraining*. SE representa a rede *squeeze-and-excitation* [19] (Figura adaptada de [9]).

permitindo capturar informações contextuais em diferentes escalas — aspecto crucial para tarefas de remoção de chuva [9]. Além disso, ela incorpora blocos *squeeze-and-excitation* [19], que recalibram dinamicamente os canais de ativação, ajustando o peso relativo de cada mapa de características (*feature map*) segundo sua transparência e intensidade. A arquitetura é organizada em múltiplos estágios sequenciais, nos quais uma rede neural recorrente (*recurrent neural network - RNN* [20], detalhada na Fig. 3) armazena e propaga informações relevantes de etapas anteriores, refinando progressivamente o processo de remoção de chuva ao longo do treinamento.

### C. Avaliação do Modelo de *Deraining*

A análise experimental foi estruturada em duas etapas principais, focando-se no processo de *deraining* e na detecção de objetos, ambas discutidas nas seções que seguem.

1) *Desempenho da RESCAN sob Diferentes Níveis de Degradação*: A arquitetura da rede *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 (3), e *SSIM* (*structural similarity index measure*) [21], definida como segue:

$$SSIM(I, \tilde{I}) = \frac{(2\mu_I\mu_{\tilde{I}} + \epsilon_1)(2\sigma_{I\tilde{I}} + \epsilon_2)}{(\mu_I^2 + \mu_{\tilde{I}}^2 + \epsilon_1)(\sigma_I^2 + \sigma_{\tilde{I}}^2 + \epsilon_2)} \quad (5)$$

onde  $\mu_I$  e  $\mu_{\tilde{I}}$  denotam, respectivamente, as médias de  $I$  e  $\tilde{I}$ ;  $\sigma_I$  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  $\epsilon_1$  e  $\epsilon_2$  são parâmetros de regularização para prevenir divisão por 0.

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 e, portanto, menor complexidade no cenário avaliado.

2) *Integração do Processo de *Deraining* com a Detecção de Objetos*: Aqui, o objetivo é avaliar o desempenho da rede de detecção de objetos *YOLO-V4* ao operar sobre imagens submetidas previamente ao processo de *deraining*, como

ilustrado na Fig. 4. O intuito é verificar se a reconstrução realizada pela rede *RESCAN* introduz artefatos visuais que possam comprometer a eficácia de tarefas posteriores, como a detecção de objetos. Dessa forma, busca-se avaliar se o pré-processamento das imagens contribui positivamente para a robustez dos modelos de visão computacional ou, ao contrário, interfere no reconhecimento semântico, como discutido em [8] (que trata de um procedimento semelhante ao *deraining*).

Nesta implementação, foi empregada uma versão pré-treinada do modelo *YOLO-V4* no *dataset COCO* [22]. Para a avaliação das detecções de objetos, comparam-se os resultados obtidos nas imagens sem chuva (imagens originais), com chuva (imagens artificialmente perturbadas) e pré-processadas utilizando o algoritmo de *deraining*, como está ilustrado na Fig. 5. Na sequência, são utilizadas as métricas *Precision* (6), *Recall* (7) e *F1-score* (8) para comparar o desempenho do modelo *YOLO-V4*. Para fins deste estudo, as imagens sem chuva são processadas pelo modelo 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.

$$Precision = \frac{TP}{TP + FP}, \quad (6)$$

$$Recall = \frac{TP}{TP + FN}, \quad (7)$$

$$F1-score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (8)$$

## III. RESULTADOS

Nesta seção, primeiramente, apresentam-se os resultados obtidos a partir da aplicação do modelo *RESCAN* em imagens submetidas a diferentes níveis de perturbações. Na sequência, considera-se a combinação do pré-processamento de *deraining* com a detecção de objetos, avaliando os impactos dessa integração.

### A. Modelo *RESCAN* com Diferentes Níveis de Perturbações

Esta seção investiga o impacto de diferentes níveis de perturbações, induzidos pelo efeito da chuva, no desempenho

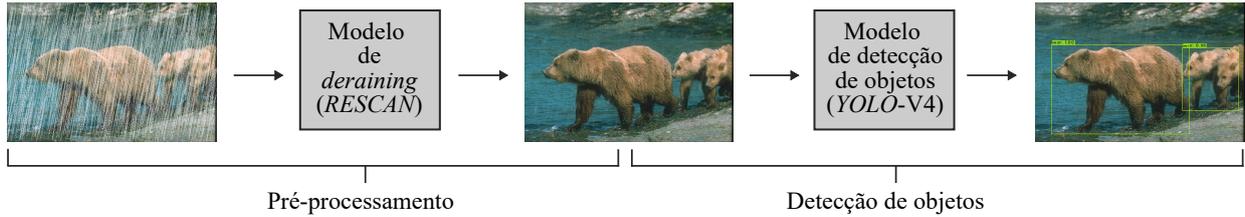


Fig. 4: Sequência de processamento de imagens, partindo da imagem perturbada por chuva, passando pelo pré-processamento de remoção de chuva (*deraining*) e concluindo com a etapa de detecção de objetos.

da rede *RESCAN*. Para tal, utiliza-se a comparação entre as versões originais das imagens (sem chuva) com suas respectivas versões degradadas (com chuva) e versões após o processo de *deraining*. Para efeito de comparação, utilizam-se as métricas *PSNR*, *SSIM* e *MSE*, anteriormente definidas.

Na fase de testes, o processo de avaliação é configurado usualmente, na rede *RESCAN*, para salvar 4 imagens referentes às últimas quatro camadas recorrentes. Essas camadas são aqui denominadas #1, #2, #3 e #4, sendo a #4 a última camada do modelo. A Fig. 6 mostra o histograma do *PSNR* das imagens após o processo de *deraining* para as diferentes camadas recorrentes consideradas. Verifica-se dessa figura que mesmo em um *dataset* com mais perturbações, como é o caso do *dataset* de 12 dB, as camadas #2, #3 e #4 possuem desempenho semelhante. Por esse motivo, consideram-se aqui os resultados finais das 3 últimas camadas de recorrência.

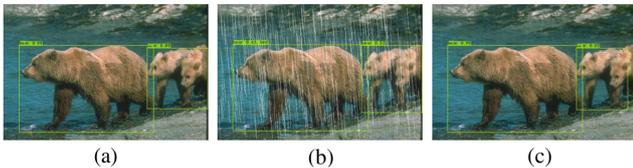


Fig. 5: Ilustração da detecção de objetos usando o *YOLO-V4*. (a) Imagem original. (b) Imagem com chuva artificial. (c) Imagem após o processo de *deraining*.

Considerando o uso das métricas *MSE* e *SSIM*, as Tabelas I e II 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 rede *RESCAN* proporcionou um aumento substancial nos valores de *SSIM* e uma redução nos valores de *MSE*, quando comparados às imagens degradadas. Por exemplo, o *SSIM* médio 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 do mais, através dos resultados das métricas consideradas, constata-se o comportamento consistente da rede *RESCAN* para os diferentes níveis de perturbações. Isto é evidenciado por valores progressivamente mais altos de *SSIM*, e menores de *MSE*. Outro ponto relevante a ser observado é a estabilidade dos resultados entre as diferentes camadas recorrentes. Como conclusão, comprova-se que a rede *RESCAN* é eficaz e robusta para a tarefa de restauração de imagens degradadas por chuva com diferentes níveis de perturbações.

TABELA I: Avaliação das imagens com chuva artificial.

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 II: Avaliação do processo de *deraining*.

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

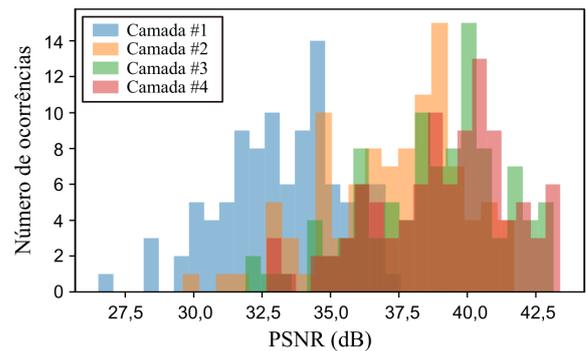


Fig. 6: Histograma dos valores de *PSNR* após o processo de *deraining*, considerando o conjunto de dados de teste com perturbações de 12 dB, para as camadas recorrentes consideradas.

### B. Impacto do Uso de *Deraining* na Detecção de Objetos

Esta seção apresenta a análise dos resultados obtidos a partir da integração entre os modelos de *deraining* e de detecção de objetos. Para a avaliação, foram consideradas as imagens geradas após o processo de *deraining* aplicado nas três últimas camadas recorrentes da rede (camadas #2, #3 e

TABELA III: Resultados das métricas de detecção de objetos. Métricas: I) *Precision*, II) *Recall*, e III) *F1-score*.

Imagens (dB)	TP	FP	FN	I	II	III
Com chuva (12 dB)	20	31	95	0,392	0,174	0,241
Após <i>deraining</i>	105	12	10	0,897	0,913	0,905
Com chuva (14 dB)	29	37	86	0,439	0,252	0,320
Após <i>deraining</i>	103	12	12	0,896	0,896	0,896
Com chuva (16 dB)	41	32	74	0,562	0,357	0,436
Após <i>deraining</i>	109	7	6	0,940	0,948	0,944
Com chuva (18 dB)	49	34	66	0,590	0,426	0,495
Após <i>deraining</i>	108	6	7	0,947	0,939	0,943
Com chuva (20 dB)	59	22	56	0,728	0,513	0,602
Após <i>deraining</i>	110	7	5	0,940	0,957	0,948

#4), conforme descrito anteriormente. Tanto as imagens degradadas pela chuva quanto aquelas restauradas foram submetidas ao detector de objetos. A Tabela III organiza os resultados do processo utilizando a camada #4. Os dados apresentados referem-se especificamente às detecções realizadas pelo modelo *YOLO-V4* utilizando as imagens restauradas oriundas do processo de *deraining*. Nessa tabela, verifica-se que, em todos os níveis de perturbações considerados, a presença da chuva prejudicou significativamente a capacidade do modelo *YOLO-V4* de classificar corretamente os objetos presentes nas imagens. Por exemplo, com o nível de 12 dB de degradação da imagem, o *F1-score* atingiu 0,241, indicando grande impacto visual nas imagens. No entanto, após o *deraining*, o desempenho foi significativamente melhorado, atingindo um *F1-score* de 0,905 para o referido conjunto de imagens. Esse comportamento se repetiu para todos os demais níveis, com o *deraining* sempre promovendo um aumento considerável de *TP* e uma redução de *FP* e *FN*, o que reforça a eficácia da técnica no restabelecimento da qualidade semântica da imagem. Tal melhoria é particularmente mais evidenciada nas faixas de menor razão sinal-ruído (12–16 dB), onde a presença de perturbações introduzidas pela chuva gera maiores distorções visuais. Nessas situações, a métrica *Precision* nas imagens com chuva varia entre 0,392 e 0,562, enquanto, após o processo de *deraining*, os valores sobem para acima de 0,896, indicando que o algoritmo é altamente eficaz na restauração de características visuais importantes para o modelo na etapa de detecção de objetos. A métrica *Recall* também segue essa tendência, mostrando que o número de objetos identificados corretamente se aproxima do verdadeiro, mesmo em condições mais adversas (com maiores níveis de perturbações).

#### IV. CONCLUSÕES

Neste trabalho de pesquisa, foi avaliado o desempenho do modelo *RESCAN* na remoção de chuva em imagens com diferentes níveis de perturbações. Além disso, verificou-se o impacto do processo de *deraining* na tarefa de detecção de objetos. Para efeito de avaliação, foi construído um conjunto de imagens com chuva artificialmente adicionada. A abordagem adotada permitiu o controle de diversos parâmetros, gerando imagens com diferentes níveis de perturbações por chuva. A integração com o modelo de detecção *YOLO-V4* evidenciou que o *deraining* contribuiu significativamente para a recuperação da acurácia de detecção de objetos nas imagens,

confirmando a eficácia do pré-processamento com *deraining* na restauração de informações semânticas.

#### REFERÊNCIAS

- [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] W. Yang, R. T. T. S. Wang, Y. Fang, and J. Liu, "Single image deraining: From model-based to data-driven and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4059–4077, Nov. 2021.
- [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] 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.
- [8] 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.
- [9] 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.
- [10] A. K. Shetty, I. Saha, R. M. Sanghvi, S. A. Save, and Y. J. Patel, "A review: Object detection models," in *2021 6th International Conference for Convergence in Technology (I2CT)*. IEEE, 2021, pp. 1–8.
- [11] H. Porav, T. Bruls, and P. Newman, "I can see clearly now: Image restoration via de-raining," in *Proc. IEEE Int. Conf. Robotics Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 7087–7093.
- [12] 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.
- [13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [14] 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.
- [15] 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.
- [16] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 3943–3956, Nov. 2020.
- [17] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 695–704.
- [18] 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.
- [19] 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.
- [20] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design & Appl.*, vol. 5, no. 1, pp. 64–67, 2001.
- [21] 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.
- [22] 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.