Tradução de Condições Meteorológicas Multidomínio em Imagens para Condução Autónoma

Carolina Alexandre Vigário Ferreira,
Mestrado em Engenharia de Redes e Sistemas Informáticos
Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2025

Tradução de Condições Meteorológicas Multidomínio em Imagens para Condução Autónoma

Carolina Alexandre Vigário Ferreira

Dissertação realizada no âmbito do Mestrado em Engenharia de Redes e Sistemas Informáticos

Departamento de Ciência de Computadores

2025

Orientador
Inês Dutra, Professor Auxiliar, Faculdade de Ciências da Universidade do Porto

Agradecimentos

Gostaria de expressar a minha gratidão à professora Inês Dutra pela orientação, dedicação, paciência e incentivo constantes, que me permitiram evoluir e concluir esta dissertação.

Agradeço de coração à minha família pelo apoio incondicional, por me terem dado a oportunidade de estudar, pela motivação constante e por estarem sempre presentes, garantindo que nada me faltasse ao longo deste percurso.

Por fim, agradeço a todos os meus amigos pela motivação, pelos momentos de convívio e por terem sido um refúgio de descontração sempre que precisei.

Reconhece-se também a utilização de ferramentas de inteligência artificial, nomeadamente o DeepL e o ChatGPT, para apoio à revisão linguística e à formatação. O conteúdo científico, análise e conclusões são da exclusiva responsabilidade da autora.

Resumo

A condução autónoma depende de sistemas de perceção que sejam capazes de interpretar o ambiente circundante com precisão. Para que seja possível alcançar um desempenho robusto, é necessário treinar esses sistemas em conjuntos de dados que representem toda a diversidade de condições encontradas em cenários do mundo real. Contudo, os conjuntos de dados disponíveis apresentam um desequilíbrio significativo, com uma predominância de imagens capturadas em condições meteorológicas favoráveis, enquanto as condições adversas, tais como chuva, neve, nevoeiro e luminosidade reduzida, estão sub-representadas. Esta limitação tem um impacto negativo na generalização dos modelos de perceção. Para resolver esta questão, a presente dissertação propõe a transformação de imagens capturadas em condições meteorológicas favoráveis em versões sintéticas das mesmas cenas sob condições meteorológicas adversas, aumentando, deste modo, a diversidade do conjunto de dados e melhorando a robustez dos modelos de condução autónoma. Para tal, foram investigadas e comparadas duas abordagens generativas: CycleGAN, um modelo de referência para tradução imagem-imagem não emparelhada dentro do paradigma GAN, e Stable Diffusion, especificamente a versão Realistic Vision, combinado com ControlNet, um modelo de difusão latente que permite a geração de imagens de alta qualidade com condicionamento estrutural explícito. Os modelos foram treinados e testados num subconjunto do conjunto de dados BDD100K, a avaliação incluiu avaliação visual qualitativa, análise de segmentação semântica e métricas quantitativas. Os resultados obtidos demonstram que o Stable Diffusion com ControlNet superou consistentemente o CycleGAN, alcançando maior preservação estrutural e um realismo visual superior. O CycleGAN, apesar de apresentar uma eficácia inferior na modelação de fenómenos climáticos complexos, constituiu uma base para destacar os ganhos alcançados com a difusão condicionada.

Palavras-chave: Condução Autónoma, Condições Climáticas Adversas, Redes Generativas Adversariais, Modelos de Difusão Latente, Stable Diffusion, CycleGAN

Abstract

Autonomous driving relies on perception systems that are capable of accurately interpreting the surrounding environment. In order to achieve robust performance, these systems must be trained on datasets that represent the full diversity of conditions encountered in real-world scenarios. However, the available datasets are significantly imbalanced, with a predominance of images captured in favorable weather conditions, while adverse conditions such as rain, snow, fog, and low light are underrepresented. This limitation has a negative impact on the generalization of perception models. To address this issue, this dissertation proposes transforming images captured in favorable weather conditions into synthetic versions of the same scenes under adverse weather conditions, thereby increasing the diversity of the dataset and improving the robustness of autonomous driving models. To this end, two generative approaches were investigated and compared: CycleGAN, a reference model for unpaired image-to-image translation within the GAN paradigm, and Stable Diffusion, specifically the Realistic Vision version, combined with ControlNet, a latent diffusion model that allows the generation of high-quality images with explicit structural conditioning. The models were trained and tested on a subset of the BDD100K dataset, and the evaluation included qualitative visual assessment, semantic segmentation analysis, and quantitative metrics. The results obtained demonstrate that Stable Diffusion with ControlNet consistently outperformed CycleGAN, achieving greater structural preservation and superior visual realism. CycleGAN, despite being less effective in modeling complex climatic phenomena, provided a basis for highlighting the gains achieved with conditional diffusion.

Keywords: Autonomous Driving, Adverse Weather Conditions, Generative Adversarial Networks, Latent Diffusion Models, Stable Diffusion, CycleGAN

Índice

Lista de Tabelas vi
Lista de Figuras vii
Lista de Abreviaturas ix

1. Introdução 1
   1.1. Objetivos 2
   1.2. Estrutura da Dissertação 2
2. Fundamentos Teóricos 3
   2.1. *Artificial Neural Networks* 3
   2.2. *Convolutional Neural Network* 6
   2.3. *Deconvolutional Neural Network* 7
   2.4. *Generative Adversarial Network* 8
   2.5. Modelos de Difusão 9
3. Revisão do Estado da Arte 11
   3.1. Modelos de Geração baseados em GANs 11
      3.1.1. CycleGAN 11
      3.1.2. StarGAN v2 12
      3.1.3. Weather GAN 12
   3.2. Modelos de Geração baseados em Difusão 13
      3.2.1. *Stable Diffusion* 13
      3.2.2. InstructPix2Pix 14
      3.2.3. ControlNet 15
      3.2.4. WeatherDG 16
   3.3. Modelos de Segmentação Semântica 17
      3.3.1. DeepLabv3+ 17
      3.3.2. HRNet 18
   3.4. Conjunto de Dados 18
      3.4.1. *The Cityscapes Dataset* 19
      3.4.2. BDD100K 19
      3.4.3. DAWN 20
      3.4.4. ACDC 20
   3.5. Síntese da Revisão do Estado de Arte 21
4. Transformação de Imagens para Condições Meteorológicas Adversas 22
   4.1. Estruturação do Conjunto de Dados para a Geração Sintética 22

4.2. Transformação de Imagens com o Stable Diffusion 24
4.2.1. *Realistic Vision* 25
4.2.2. ControlNet 26
4.2.3. Implementação do pipeline 28
4.2.4. Procedimentos Experimentais do Stable Diffusion 31
4.2.4.1. *Prompt* 34
4.3. Transformação de Imagens com CycleGAN 36
4.3.1. Configuração e Treino do CycleGAN 37
4.3.2. Procedimentos Experimentais do CycleGan 39
4.4. Metodologia de Análise e Avaliação 40
4.4.1. Avaliação Qualitativa 40
4.4.2. Segmentação Semântica 40
4.4.2.1. Implementação da Segmentação Semântica 41
4.4.3. Avaliação Quantitativa das Imagens Geradas 42
4.4.3.1. Implementação das Métricas Quantitativas 43
5. Análise e Avaliação dos Resultados Obtidos 45
5.1. Análise Qualitativa 45
5.1.1. Chuva 45
5.1.2. Neve 46
5.1.3. Nevoeiro 47
5.1.4. Noite 49
5.1.5. Síntese da Análise Qualitativa 50
5.2. Segmentação Semântica 51
5.3. Análise Quantitativa 53
5.3.1. Síntese da Análise Quantitativa 56
6. Conclusão 57

Bibliografia 58
A. Anexo A: Exemplos adicionais das imagens geradas 70

Lista de Tabelas

4.1. Número de imagens utilizadas no treino do CycleGAN por condição atmosférica.....38

5.1. Resultados quantitativos por condição climática. 54

Lista de Figuras

2.1. Estrutura de uma Rede Neuronal Artificial (ANN) com camada de entrada (*Input Layer*), camadas ocultas (*Hidden Layers*) e camada de saída (*Output Layer*). [7] 4
2.2. Estrutura de uma Rede Adversarial Generativa (GAN), composta por um gerador, que cria imagens sintéticas a partir de ruído aleatório, e um discriminador, que avalia se as imagens são reais (do conjunto de treino) ou falsas (geradas). [14] 8

4.1. Representação do processo de difusão de imagens utilizado pelo Stable Diffusion. [66].....24
4.2. Exemplo dos mapas de condicionamento utilizados pelo ControlNet. (a) Imagem original; (b) Mapa de profundidade (Depth), que fornece informações sobre a distância dos objetos; (c) Mapa de contornos (Canny), que destaca as bordas e estruturas principais da cena; (d) Mapa de segmentação semântica (Seg), que classifica e localiza os diferentes elementos da imagem. 27
4.3. Influência do parâmetro *strength* na geração da imagem, variando de valores baixos (b) até valores elevados (e). 31
4.4. Influência do parâmetro do número de passos de inferência na geração da imagem, variando de valores baixos (b) até valores elevados (e). 32
4.5. Influência do parâmetro *Classifier-Free Guidance Scale*, variando de valores baixos (b) até valores elevados (e). 32
4.6. Influência do peso dos mapas do ControlNet na imagem gerada, variando de valores baixos (b) até valores elevados (e). 33
4.7. Diagrama da arquitetura do CycleGAN. [76] 37

5.1. Exemplos de transformação de imagens para a condição de chuva. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion. 46
5.2. Exemplos de transformação de imagens para a condição de neve. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion. 47
5.3. Exemplos de transformação de imagens para a condição de nevoeiro. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion. 48
5.4. Exemplos de transformação de imagens para a condição de noite. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion. 50
5.5. Exemplos de transformação de imagens pelo modelo Cyclegan e as respetivas segmentações. (a) Imagem original. (b–e) Imagens geradas pelo CycleGAN para chuva, neve, nevoeiro e noite, respetivamente. (f–j) Segmentações correspondentes a cada imagem acima. 52
5.6. Exemplos de transformação de imagens pelo modelo Stable Diffusion e as respetivas segmentações. (a) Imagem original. (b–e) Imagens geradas pelo Stable Diffusion para chuva, neve, nevoeiro e noite, respetivamente. (f–j) Segmentações correspondentes a cada imagem acima. 53
5.7. Comparação das métricas PSNR e SSIM para diferentes condições meteorológicas.54
5.8. Comparação da métrica LPIPS para diferentes condições meteorológicas. 55

1. Exemplos de imagens para a condição de chuva. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas. 70
2. Exemplos de imagens para a condição de neve. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas. 71
3. Exemplos de imagens para a condição de nevoeiro. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas. 72
4. Exemplos de imagens para a condição de noite. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas. 73

Lista de Abreviaturas

ACDC *Adverse Conditions Dataset with Correspondences*. iv, 20

Adam *Adaptive Moment Estimation*. 5

ANN *Artificial Neural Network*. 3

BDD100K *Berkeley DeepDrive 100K*. iv, 19

CFG *Classifier-Free Guidance*. 16

CNN *Convolutional Neural Network*. 6

DAWN *Detection in Adverse Weather Nature*. iv, 20

FID *Fréchet Inception Distance*. 43

GAN *Generative Adversarial Network*. 8

HRNet *High-Resolution Networks*. iv, 18

IA Inteligência Artificial. 3

KID *Kernel Inception Distance*. 43

LDM *Latent Diffusion Models*. 13

LLM *Large Language Model*. 16

LPIPS *Learned Perceptual Image Patch Similarity*. 42

ML *Machine Learning*. 3

PSNR *Peak Signal-to-Noise Ratio*. 42

ReLU *Rectified Linear Unit*. 4

SSIM *Structural Similarity Index Measure*. 42

1. Introdução

A condução autónoma tem vindo a emergir como um dos principais focos de investigação na área da visão computacional e da inteligência artificial, apresentando aplicações que prometem revolucionar a mobilidade e aumentar a segurança rodoviária. Para que os veículos autónomos operem de forma fiável e segura, é necessário que os sistemas de perceção sejam capazes de identificar e compreender, com elevada precisão, os elementos presentes no ambiente rodoviário.

No entanto, a fiabilidade destes sistemas depende fortemente da qualidade e diversidade dos dados utilizados no seu treino. A partir da análise dos principais conjuntos de dados públicos, tais como o BDD100K [1] e o Cityscapes [2], foi possível constatar uma distribuição desigual entre imagens captadas em condições meteorológicas favoráveis e desfavoráveis. Em particular, verifica-se uma sub-representação significativa de cenários com precipitação, neve, nevoeiro e baixa luminosidade. Esta limitação compromete a capacidade dos modelos de generalizar para situações adversas, conduzindo a uma degradação no desempenho de tarefas críticas, como a deteção de objetos e a segmentação semântica.

Dada a dificuldade prática de recolher dados equilibrados, devido à imprevisibilidade das condições meteorológicas, uma abordagem promissora consiste na geração de imagens sintéticas que complementem os conjuntos de dados existentes. Neste contexto, os modelos generativos assumem um papel central, ao permitirem a tradução de imagens capturadas em condições favoráveis para diferentes cenários meteorológicos de forma controlada e realista.

Nos últimos anos, os modelos de difusão latente, como o Stable Diffusion [3], têm demonstrado resultados de ponta na geração de imagens de alta fidelidade. A integração com arquiteturas como o ControlNet [4] possibilita o uso de condicionamentos adicionais, proporcionando um controlo preciso sobre os elementos que devem ser preservados na cena. Paralelamente, modelos como o CycleGAN [5] permitem a realização de traduções de imagem para imagem entre diferentes domínios meteorológicos, mesmo na ausência de pares correspondentes.

1.1. Objetivos

Esta dissertação propõe o desenvolvimento de uma pipeline de transformação climática aplicada a cenários de condução autónoma. O objetivo principal é testar e avaliar a eficácia de diferentes abordagens generativas na transformação de imagens em condições meteorológicas favoráveis em versões sintéticas das mesmas cenas sob condições meteorológicas adversas, visando o aumento da diversidade dos conjuntos de dados existentes e, consequentemente, a melhoria do desempenho de modelos de perceção em condições adversas.

Para tal, são explorados dois paradigmas principais: os modelos de difusão, nomeadamente o Stable Diffusion, especificamente a versão Realistic Vision, aliado ao ControlNet, e os modelos baseados em GANs, especificamente o CycleGAN. O conjunto de dados utilizado como base neste trabalho é o BDD10K, um subconjunto do BDD100K, escolhido pela sua diversidade visual e relevância para o contexto de condução autónoma.

Ao abordar o desafio da transformação climática, esta dissertação visa contribuir para o desenvolvimento de técnicas de aumento de dados aplicadas à condução autónoma, explorando a utilização de modelos generativos para transformar imagens de cenários rodoviários em condições meteorológicas adversas e avaliando se a qualidade das imagens geradas é adequada para o seu uso no treino e melhoria de modelos de perceção em sistemas de condução autónoma.

1.2. Estrutura da Dissertação

A dissertação está organizada em seis capítulos, incluindo este:

- Capítulo 2, **Fundamentos Teóricos**, são apresentados os conceitos fundamentais.
- Capítulo 3, **Revisão do Estado da Arte**, é apresentada uma descrição do estado da arte.
- Capítulo 4, **Transformação de Imagens para Condições Meteorológicas Adversas**, é detalhada a metodologia desenvolvida.
- Capítulo 5, **Análise e Avaliação dos Resultados Obtidos**, é apresentada a análise e a avaliação dos resultados.
- No Capítulo 6, **Conclusão**, são apresentadas as conclusões e as sugestões para trabalhos futuros.

2. Fundamentos Teóricos

Neste capítulo, são apresentados os conceitos fundamentais que servirão de base para a compreensão do tema abordado nesta dissertação. Estes conceitos fornecem uma visão geral dos principais assuntos relacionados com o trabalho.

2.1. *Artificial Neural Networks*

A Inteligência Artificial (IA) tem como objetivo desenvolver sistemas capazes de realizar tarefas típicas da inteligência humana, como o reconhecimento de padrões e a tomada de decisão. O aumento dos dados e do poder de processamento computacional impulsionou o surgimento da *Machine Learning* (ML), que permite que os sistemas aprendam diretamente a partir dos dados. No entanto, os métodos tradicionais, como as árvores de decisão e as máquinas de vetor de suporte, revelaram limitações face a dados de elevada dimensionalidade e tarefas complexas de visão computacional. Para superar estes desafios, surgiram as redes neuronais artificiais (*Artificial Neural Network* (ANN)).

Uma rede neuronal artificial (ANN) [6] é um modelo computacional inspirado no funcionamento do cérebro humano, projetado para reconhecer padrões, aprender com dados e tomar decisões com base em informações fornecidas.

As ANN são compostas por neurónios artificiais denominados unidades. Estas unidades estão dispostas numa série de camadas que, em conjunto, constituem toda a rede neuronal artificial de um sistema.

O modelo básico de uma ANN é composto pelas seguintes camadas:

- **Camada de entrada**: recebe os dados do mundo exterior que a rede neuronal precisa de analisar ou aprender;
- **Camadas ocultas**: transformam os dados recebidos pela camada de entrada em dados úteis para a camada de saída. Pode haver uma ou várias camadas ocultas, consoante a complexidade da rede;
- **Camada de saída**: produz o resultado final da rede sob a forma de uma resposta das redes neuronais artificiais aos dados de entrada fornecidos, que pode ser uma classificação, uma previsão ou outro tipo de resposta, consoante o problema.

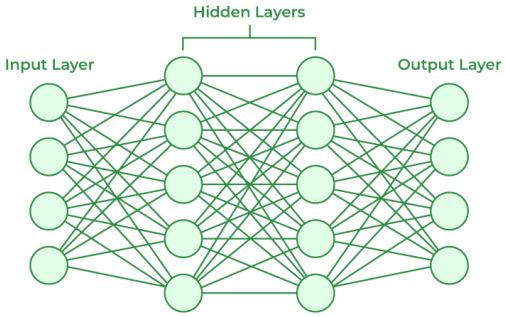


Figura 2.1: Estrutura de uma Rede Neuronal Artificial (ANN) com camada de entrada (*Input Layer*), camadas ocultas (*Hidden Layers*) e camada de saída (*Output Layer*). [7]

A Figura 2.1 demonstra a estrutura de uma rede neuronal artificial, onde cada unidade está conectada a várias unidades na camada seguinte através de conexões ponderadas. Cada conexão possui um peso, que representa a força ou importância dessa conexão na transmissão do sinal. Durante o treino, os pesos são ajustados de modo a permitir que a rede aprenda quais as conexões que se revelam mais relevantes para produzir a saída correta a partir das entradas fornecidas.

Com o intuito de capturar relações complexas e não lineares, cada neurónio aplica uma função de ativação ao sinal recebido. A *Rectified Linear Unit* (ReLU) [8], uma das mais comuns, transforma valores negativos em zero e mantém os positivos inalterados, facilitando o treino de redes profundas e prevenindo o fenómeno de desvanecimento do gradiente. Outras funções, tais como a sigmoid [9] ou a tanh, podem igualmente ser utilizadas, consoante a aplicação.

As redes empregam ainda metodologias de normalização e regularização, com o propósito de otimizar a estabilidade e a robustez. A normalização por lotes (*Batch Normalization*) é um processo que permite ajustar as ativações intermediárias de cada camada, mantendo os seus valores dentro de uma faixa estável. Este processo acelera o treino e torna a rede menos sensível à inicialização dos pesos. O *Dropout* desativa aleatoriamente uma fração de neurónios em cada iteração, evitando que a rede dependa de neurónios específicos ou de combinações exatas, promovendo uma aprendizagem mais generalizada e prevenindo o *overfitting*.

Estas redes são treinadas com recurso a um conjunto de dados denominado conjunto de treino. Durante o processo, a rede recebe exemplos de entradas juntamente com as respetivas saídas desejadas, de modo a aprender a associar corretamente essas entradas às suas respostas.

O funcionamento desta rede pode ser dividido em duas fases principais: *Forward Propagation* e *Back Propagation*.

- ***Forward Propagation***: Durante este processo, a rede processa as entradas recebidas e gera uma saída, percorrendo sequencialmente as suas camadas. Em cada camada, cada neurónio executa uma operação matemática, que consiste na multiplicação das entradas pelos pesos, adição de um viés e aplicação de uma função de ativação, que permite introduzir não linearidade no modelo. Este processo é repetido em todas as camadas da rede, desde a camada de entrada até à camada de saída. A camada final produz a previsão do modelo.

  O objetivo desta fase é gerar uma previsão com base nos parâmetros atuais da rede, a qual será posteriormente comparada com o rótulo verdadeiro para calcular o erro através de uma função de custo (*loss function*). As funções de custo (ou funções de perda) medem a diferença entre a saída prevista pela rede e a saída correta desejada.

- ***Backpropagation***: Nesta etapa, os parâmetros da rede são atualizados com base no erro calculado durante a *Forward Propagation*, com o intuito de reduzi-lo. Este processo utiliza a regra da cadeia do cálculo diferencial para calcular os gradientes da função de custo em relação aos pesos e viés da rede.

  A correção dos parâmetros é geralmente efetuada por meio de algoritmos de otimização, como o gradiente descendente ou o *Adaptive Moment Estimation* (Adam) [10], ajustando a taxa de aprendizagem para cada parâmetro e acelerando a convergência do treino.

  Este ciclo é repetido sucessivamente ao longo de várias épocas (*epochs*), até que o modelo atinja a minimização da função de custo e aprenda os padrões presentes nos dados.

A capacidade das redes neuronais artificiais de identificar padrões complexos em dados é fundamental para a criação de redes mais específicas, como as redes neuronais convolucionais (CNN). Estas são amplamente utilizadas em visão computacional para tarefas como a segmentação e a deteção de objetos em imagens.

2.2. *Convolutional Neural Network*

Uma Rede Neuronal Convolucional (*Convolutional Neural Network* (CNN)) [11] é um tipo de rede neuronal artificial especialmente projetada para processar dados com uma estrutura de grade, como imagens ou sinais de áudio, em que os elementos, como os píxeis, têm uma relação espacial com os seus vizinhos.

As redes neuronais tradicionais tratam cada pixel de forma independente, ignorando essa relação espacial, ao contrário das CNN, que a preservam ao processar os dados em blocos locais.

As CNN [11] são constituídas por várias camadas que constituem os blocos de construção fundamentais:

- **Camada de Convolução**: A camada convolucional é o componente central das CNNs, realiza uma operação que implica a aplicação de filtros especializados, conhecidos como *kernels*, que são pequenas matrizes de números que percorrem a imagem de entrada, multiplicando os valores do *kernel* pelos valores originais dos píxeis da imagem e somando, em seguida, os resultados. Esta operação cria um mapa de características (*feature map*), que realça padrões visuais complexos e extrai características como arestas, texturas e formas.

  Os *kernels* são aplicados a toda a imagem e têm os mesmos pesos. Isto significa que a rede consegue reconhecer padrões em qualquer parte da imagem. Esta estratégia reduz o número de parâmetros que é necessário aprender, tornando a rede mais eficiente.

  Existem alguns parâmetros que controlam a operação de convolução:

  - **Tamanho do *Kernel***: o tamanho do *kernel* determina as dimensões da ”área”que será utilizada para processar a imagem de entrada, influencia a capacidade da rede de capturar diferentes tipos de padrões. *Kernels* maiores podem capturar características mais complexas, mas podem perder detalhes finos.
  - ***Stride***: o *stride* refere-se ao número de píxeis que o *kernel* avança à medida que percorre a imagem. Um *stride* maior resulta numa saída com dimensões mais pequenas, o que significa que a operação de convolução será mais rápida, mas pode resultar na perda de informações detalhadas da imagem.
  - ***Padding***: o *padding* é uma técnica utilizada para adicionar píxeis extras ao redor da imagem de entrada antes de aplicar a operação de convolução. Isto

é feito para controlar o tamanho da saída e garantir que as características nas bordas da imagem sejam preservadas.

- **Camadas de *Pooling***: Esta camada complementa as convolucionais, reduzindo a dimensionalidade dos mapas de características sem perder informações essenciais e descartando detalhes redundantes. Desta forma, minimiza o custo computacional e ajuda a evitar o *overfitting*, excesso de ajuste aos dados de treino. Isto é conseguido através da seleção dos píxeis do mapa de características e cálculo da média desses píxeis (*average pooling*) ou seleção do valor máximo (*max pooling*). Várias camadas convolucionais e de *pooling* podem ser empilhadas para aprender características cada vez mais complexas a partir da entrada. Isto ajuda a captar diferentes níveis de abstração e hierarquia nos dados.

As CNN também utilizam camadas totalmente conectadas, funções de ativação e técnicas de normalização e regularização, mas de forma integrada a uma arquitetura que prioriza a extração hierárquica de características espaciais das imagens. As camadas totalmente conectadas recebem os mapas de características gerados pelas camadas convolucionais e de *pooling*, condensando a informação para produzir a previsão final. As funções de ativação, como a ReLU, introduzem não linearidade, permitindo que a rede aprenda padrões complexos em diferentes níveis de abstração. Técnicas como *Batch Normalization* e *Dropout* são igualmente utilizadas para estabilizar e regularizar o treino, garantindo que a rede generalize bem para imagens novas e variadas.

2.3. *Deconvolutional Neural Network*

As redes neuronais deconvolucionais (DNN) [12] são redes neuronais convolucionais que funcionam num processo invertido. Estas esforçam-se por encontrar características ou sinais perdidos que, anteriormente, não tenham sido considerados importantes para a tarefa de uma CNN. Um sinal pode ficar perdido devido à convolução com outros sinais. A deconvolução de sinais pode ser utilizada tanto na síntese como na análise de imagens.

Também conhecidas como redes transpostas convolucionais, estas redes são projetadas para expandir as dimensões espaciais dos dados em vez de as reduzir, aplicando camadas de transposição convolucional que, em vez de compactar informações, geram saídas com maior resolução ou dimensão espacial, reconstruindo dados detalhados.

2.4. *Generative Adversarial Network*

As Redes Generativas Adversariais (*Generative Adversarial Network* (GAN)) [13] constituem um tipo de modelo projetado para gerar dados novos e realistas a partir de exemplos existentes, incluindo imagens, áudio ou texto. A inovação mais significativa reside na implementação de duas redes neuronais que competem entre si: o gerador e o discriminador, que interagem num processo de aprendizagem adversarial.

O gerador tem como objetivo criar amostras sintéticas que se assemelhem aos dados reais. À medida que produz novas amostras, tenta enganar o discriminador, fazendo com que este acredite que são reais.

O discriminador, por sua vez, aprende a distinguir entre amostras reais e geradas, fornecendo *feedback* ao gerador sobre o quão convincentes são as suas amostras.

A Figura 2.2 apresenta a estrutura de uma Rede Adversarial Generativa (GAN), composta por um gerador e um discriminador.

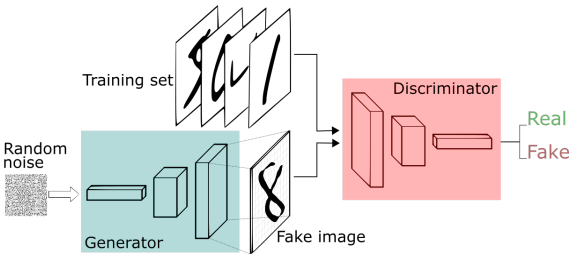


Figura 2.2: Estrutura de uma Rede Adversarial Generativa (GAN), composta por um gerador, que cria imagens sintéticas a partir de ruído aleatório, e um discriminador, que avalia se as imagens são reais (do conjunto de treino) ou falsas (geradas). [14]

Esta interação origina um ciclo de aprendizagem contínuo, em que ambos os componentes aperfeiçoam progressivamente as suas competências: o gerador adquire maior aptidão na criação de amostras realistas e o discriminador exibe uma maior precisão na sua avaliação.

O treino de uma GAN é fundamentado num processo alternado. Em cada iteração, os pesos do discriminador são atualizados primeiramente, com o propósito de maximizar a sua capacidade de classificar corretamente amostras reais e falsas. Posteriormente, procede-se ao ajustamento dos pesos do gerador, com vista a minimizar a capacidade do discriminador em distinguir as amostras deste das amostras reais. Este ciclo possibilita que o gerador aprenda progressivamente a produzir amostras cada vez mais realistas, enquanto o discriminador se adapta às novas estratégias do gerador.

As GANs empregam funções de perda específicas para orientar o treino. A função de perda do discriminador avalia a capacidade de distinguir entre amostras reais e amostras geradas, penalizando as classificações incorretas. A função de perda do gerador avalia a eficácia da criação de amostras mais realistas, ao incentivar a criação de amostras mais realistas. Este conceito é frequentemente descrito como um jogo minimax, no qual o gerador procura minimizar a capacidade do discriminador de diferenciar as amostras, enquanto o discriminador procura maximizar a sua precisão.

A capacidade das GANs de criar dados realistas a partir de exemplos existentes torna-as uma ferramenta poderosa para aplicações de visão computacional, especialmente em cenários onde a recolha de dados reais é dispendiosa ou limitada.

Entre as diversas variantes das GANs, destaca-se o CycleGAN, que será explorado neste trabalho. Este modelo introduz a ideia de tradução de imagens entre domínios sem necessidade de pares correspondentes.

2.5. Modelos de Difusão

Os modelos de difusão [15] são modelos generativos utilizados principalmente para gerar imagens e outras tarefas de visão computacional. Baseiam-se em redes neuronais treinadas por aprendizagem profunda, que adicionam progressivamente ruído aleatório aos dados e, posteriormente, invertem esse processo para gerar imagens realistas.

Durante o treino, um ponto de dados é gradualmente corrompido pela adição de ruído gaussiano, até se tornar praticamente irreconhecível. O modelo aprende então o processo inverso: remover esse ruído passo a passo, de modo a recuperar a distribuição original dos dados e gerar novas amostras realistas.

O processo de difusão pode ser descrito em duas fases principais:

- **Difusão direta (*forward diffusion*)**: consiste na adição gradual de ruído gaussiano às imagens, ao longo de vários passos, de modo a transformá-las em representações ruidosas.
- **Difusão inversa (*reverse diffusion*)**: envolve o treino de um modelo, geralmente chamado de preditor de ruído, que aprende a estimar e remover o ruído introduzido em cada passo. Este processo é realizado de forma iterativa, reconstruindo progressivamente a imagem inicial ou gerando uma nova amostra a partir de puro ruído.

Quando existe condicionamento textual, como nos modelos do tipo texto-para-imagem (por exemplo, Stable Diffusion), os *embeddings* do codificador de texto são transmitidos ao preditor de ruído por meio de atenção cruzada (*cross-attention*). Esta funcionalidade possibilita a integração de instruções textuais nas etapas do processo de difusão, proporcionando orientação à geração de acordo com a descrição fornecida.

Um dos modelos mais relevantes desta categoria é o Stable Diffusion, amplamente utilizado pela sua capacidade de gerar imagens de alta qualidade e diversidade. Neste trabalho, foi empregue uma variante deste modelo.

Embora os fundamentos discutidos até aqui sejam essenciais para compreender a área, é necessário examinar como essas ideias foram aplicadas e desenvolvidas na prática. O capítulo seguinte dedica-se a rever o estado da arte, evidenciando os avanços recentes.

3. Revisão do Estado da Arte

Na presente secção, é apresentada uma revisão da literatura relacionada com o tema deste trabalho, com o propósito de contextualizar os desenvolvimentos e identificar as abordagens mais relevantes na área. São explorados os principais modelos generativos utilizados na geração de imagens, com destaque para modelos baseados em redes generativas adversariais (GANs) e em modelos de difusão. A análise aborda o funcionamento, vantagens e limitações destes modelos. Adicionalmente, são discutidos modelos de segmentação de imagens e os conjuntos de dados mais utilizados neste domínio.

3.1. Modelos de Geração baseados em GANs

Uma das abordagens mais influentes para a geração de dados sintéticos é a das Redes Generativas Adversariais (GANs). Na presente secção, serão apresentados os modelos GAN relevantes para o presente trabalho.

3.1.1 CycleGAN

O CycleGAN [5] consiste num modelo de tradução imagem-a-imagem, proposto por Zhu et al. (2017), que possibilita o mapeamento de imagens entre domínios sem a necessidade de pares correspondentes. O modelo recorre a dois mapeamentos principais, $G : (X \rightarrow Y)$ e $F : (Y \rightarrow X)$, que correspondem aos geradores responsáveis por traduzir imagens entre os dois domínios, cada um acompanhado por um discriminador (DX e DY), cuja função é avaliar o realismo das imagens geradas.

Durante o treino, os geradores tentam enganar os discriminadores, enquanto estes aprendem a distinguir entre imagens reais e sintetizadas, formando um processo adversarial que melhora progressivamente a qualidade das traduções. No entanto, confiar exclusivamente nas perdas adversariais pode conduzir o modelo a gerar imagens plausíveis, mas não relacionadas com a entrada. Para evitar esse problema, o CycleGAN introduz a perda de consistência de ciclo, que garante que, ao traduzir uma imagem para um outro domínio e reverte-la, seja possível recuperar uma versão próxima da original. Este mecanismo força os geradores a preservar a estrutura e os detalhes fundamentais da cena, alterando unicamente o estilo visual.

Nos experimentos, o modelo foi avaliado em conjuntos como Cityscapes [16] e fotos aéreas do Google Maps, mostrando resultados comparáveis ao Pix2Pix [17] e superiores a

outros métodos não supervisionados (CoGAN [18], SimGAN [19], BiGAN [20], ALI [21]). Embora tenha demonstrado versatilidade em tarefas como transferência de estilo ou mudança de estação, apresenta limitações em transformações geométricas complexas (por exemplo, de cães para gatos).

Para este trabalho, o CycleGAN é relevante devido à sua vasta aplicação como padrão em tradução de imagens entre domínios não emparelhados, o que se revela crucial neste âmbito, uma vez que a maioria dos conjuntos de dados de condução autónoma não inclui pares de imagens correspondentes sob diferentes condições meteorológicas.

3.1.2 StarGAN v2

O StarGAN v2 [22], proposto por Choi et al. (2020), foi desenvolvido para tradução de imagens em múltiplos domínios utilizando apenas um gerador. O modelo em questão substitui as etiquetas de domínio por códigos de estilo, obtidos de uma rede de mapeamento ou extraídos de imagens de referência, incorporados no gerador via AdaIN. A avaliação da autenticidade e consistência de estilo é realizada através de um discriminador multitarefa, que opera de forma simultânea.

Nos experimentos, o StarGAN v2 demonstrou superioridade em comparação com métodos anteriores, tais como o MUNIT [23], o DRIT [24], o MSGAN [25] e o StarGAN original [26], em conjuntos de dados como o CelebA-HQ [27] e o AFHQ. Este modelo obteve os melhores valores de FID [28] e LPIPS [29], tendo recebido as avaliações humanas mais favoráveis em termos de qualidade e diversidade das imagens.

Apesar da eficácia demonstrada pelo modelo na tradução multimodal de rostos e animais, o mesmo foi concebido para cenários com ênfase estética, não abordando diretamente contextos estruturados, como ambientes rodoviários ou condições meteorológicas. Para este trabalho, este modelo foi descartado também por ser mais adequado à realização de múltiplas transformações condicionais em simultâneo, enquanto o presente estudo centra-se em alterações de uma condição climática de cada vez.

3.1.3 Weather GAN

O WeatherGAN [30], proposto por Li, Kou e Zhao (2021), foi concebido especificamente para a tradução de imagens entre diferentes condições meteorológicas, visa modificar exclusivamente os elementos meteorológicos de uma imagem, mantendo as regiões irrelevantes inalteradas.

O modelo em questão recorre a uma arquitetura GAN, apresentando três módulos principais: (i) tradução global inicial; (ii) módulo de atenção, que restringe as alterações às regiões relevantes; e (iii) módulo de segmentação de pistas meteorológicas, que guia a modificação seletiva dos elementos climáticos.

O treino combina perdas adversarial, de ciclo, inspirado no CycleGAN, e de classificação, garantindo assim realismo, preservação da estrutura da cena e correspondência correta da condição gerada. Avaliado no conjunto de dados Multitask Weather [31], o WeatherGAN demonstrou superioridade em relação a modelos como CycleGAN [5], UNIT [32] e MUNIT [23] em métricas como FID [28] e KID [33], produzindo imagens mais naturais e com maior controlo sobre a intensidade dos efeitos climáticos.

Embora ainda apresente falhas em traduções complexas, este modelo destaca-se por abordar diretamente o problema da manipulação de múltiplas condições atmosféricas em imagens reais, o que o torna relevante para este trabalho. No entanto, a indisponibilidade do código-fonte público inviabilizou a sua utilização prática neste trabalho.

3.2. Modelos de Geração baseados em Difusão

Recentemente, os modelos de difusão emergiram como uma alternativa poderosa para a geração de dados sintéticos. Nesta secção, serão discutidas as variantes mais relevantes para o presente trabalho, propostas na literatura.

3.2.1 *Stable Diffusion*

Os *Latent Diffusion Models* (LDM), proposto por Rombach et al. (2022) [3], como o Stable Diffusion, foram desenvolvidos com o propósito de contornar o elevado custo computacional das primeiras implementações de modelos de difusão, mantendo simultaneamente uma elevada fidelidade visual.

A arquitetura introduz uma separação entre a fase compressiva e a fase generativa. As imagens são primeiramente mapeadas para um espaço latente comprimido através de um *autoencoder* variacional [34], que preserva as características percetualmente relevantes enquanto reduz significativamente a dimensionalidade dos dados. Este espaço latente disponibiliza uma representação mais compacta, permitindo que o modelo de difusão se concentre nos elementos semânticos fundamentais da imagem, realizando o treino.

Durante o treino, aplica-se um processo de difusão direta, no qual as representações latentes são progressivamente corrompidas com ruído gaussiano ao longo de vários passos, até se tornarem praticamente indistinguíveis de puro ruído. O modelo é então otimizado para aprender o processo inverso.

No espaço latente, o modelo executa o processo de difusão recorrendo a uma U-Net [35] condicionada temporalmente, que investiga a estrutura espacial das imagens. A difusão é aplicada de forma sequencial, com a remoção do ruído e a reconstrução gradual da representação latente até se obter uma versão consistente da imagem. Adicionalmente, os LDMs podem ser influenciados por diversos tipos de informação, tais como *prompts* de texto, mapas semânticos ou imagens de referência, através de mecanismos de atenção cruzada, possibilitando o controle do processo de síntese e a realização de tarefas de geração condicionada.

Por fim, a representação latente obtida é decodificada de volta para o espaço de píxeis pelo *decoder* do *autoencoder*, produzindo imagens finais com alta fidelidade estrutural e visual, mas com custo computacional substancialmente inferior ao treino direto no espaço de píxeis.

Nos testes realizados, os LDM superaram as GAN e as difusões clássicas em termos de qualidade visual, custo computacional e capacidade de geração condicionada, obtendo baixos valores de FID em conjuntos de dados como o CelebA-HQ [27]. No entanto, continuam a ser mais lentos que os GAN [13] e podem não atingir a precisão pixel a pixel em determinadas tarefas.

Neste trabalho, o Stable Diffusion foi utilizado em virtude da sua capacidade de gerar imagens de elevada qualidade operando no espaço latente. Este modelo possibilitou a aplicação de condicionamentos adicionais via ControlNet, que se revelou crucial para garantir a preservação da estrutura das imagens.

3.2.2 InstructPix2Pix

O InstructPix2Pix [36], desenvolvido por Brooks et al. (2023), é um modelo de edição de imagens baseado em instruções textuais, treinado com um conjunto de dados sintéticos gerado a partir de dois modelos pré-treinados: GPT-3 [37], responsável pela criação das instruções, e Stable Diffusion [3], usado para gerar as imagens antes e depois das modificações.

A arquitetura do sistema é fundamentada na difusão condicional, adaptada para integrar de forma simultânea a imagem original e a instrução textual como referência. Foram introduzidos novos parâmetros para processar essas informações, mantendo-se os pesos pré-treinados inalterados. Adicionalmente, foram aplicadas técnicas de orientação sem classificador para reforçar a fidelidade da edição às instruções.

O modelo demonstrou capacidade para realizar múltiplas edições, tais como a substituição de objetos, mudanças de estilo e cenário, alterações sazonais e conversão de materiais, apresentando uma boa consistência entre texto e imagem, conforme medido por CLIP [38]. Contudo, apresenta limitações: a qualidade visual depende do difusor utilizado, a precisão do GPT-3 nas instruções não é perfeita e o método *Prompt-to-Prompt* [39] possui capacidade limitada para alterações consistentes.

Para este trabalho, o InstructPix2Pix é interessante como referência de técnicas de edição condicionada por texto, mas não é diretamente aplicável, pois o foco da dissertação está na manipulação de condições meteorológicas em imagens de estrada, onde a preservação estrutural e os elementos visuais são críticos.

3.2.3 ControlNet

Os modelos de geração de imagens, baseados em difusão, geram resultados visualmente impressionantes, contudo, apresentam limitações relativamente ao controlo da estrutura espacial. Para contornar esta limitação, os investigadores propuseram o recurso a entradas visuais adicionais, tais como mapas de bordas, segmentações, poses e profundidade, que servem de condições para orientar o processo de geração. Nesse sentido, Zhang, Rao e Agrawala (2023) propuseram a arquitetura ControlNet [4].

O método em questão baseia-se na incorporação de condições suplementares a um modelo pré-treinado, como o Stable Diffusion, preservando o seu desempenho original. Para tal, os parâmetros do modelo base são congelados e é criada uma cópia treinável dos blocos da U-Net [35] responsável pelo processamento das informações condicionais. Esta cópia é ligada ao modelo original através das convoluções zero, que se iniciam com pesos nulos com o intuito de evitar alterações iniciais na geração e, durante o processo de treino, passam a integrar gradualmente o novo condicionamento.

O modelo é treinado de forma a depender apenas das entradas condicionais (sem texto) em metade dos casos, reforçando a interpretação das condições visuais. As convoluções

zero possibilitam a preservação da qualidade do modelo original nos estágios iniciais, permitindo que o modelo manifeste um fenómeno designado convergência súbita, no qual, após um número reduzido de iterações, o modelo passa a seguir fielmente as condições fornecidas.

Adicionalmente, o impacto do condicionamento pode ser regulado através do ajuste da ponderação do *Classifier-Free Guidance* (CFG) [40], responsável pelo equilíbrio entre a interpretação textual e visual. O ControlNet também suporta a combinação de múltiplas fontes de condicionamento, ao integrar diretamente as suas saídas no modelo principal.

A aplicação do ControlNet ao Stable Diffusion demonstrou eficácia em diversos tipos de condicionamento, incluindo Canny Edge [41], mapas de profundidade [42], mapas normais [43], linhas M-LSD [44], HED soft edge [45], segmentações ADE20K [46], OpenPose [47] e esboços manuais. O modelo em questão gera imagens detalhadas e coerentes, mesmo na ausência de texto, mantendo a qualidade do modelo base. A validação da fidelidade e nitidez visual das imagens geradas foi efetuada através da aplicação de métricas quantitativas, tais como IoU (via OneFormer [48]), FID e CLIP.

No contexto deste trabalho, o ControlNet foi utilizado em conjunto com o Stable Diffusion, permitindo a utilização de vários mapas de condicionamento e assegurando um controlo sobre a composição e a estrutura da imagem, na transformação para condições meteorológicas adversas.

3.2.4 WeatherDG

O WeatherDG [49], proposto por Qian et al. (2024), é um modelo destinado à geração de imagens realistas de cenários rodoviários sob diferentes condições meteorológicas e de condução. Este modelo combina Stable Diffusion com agentes *Large Language Model* (LLM) para gerar *prompts* processuais que guiam a síntese das imagens.

A arquitetura baseia-se no *fine-tuning* do Stable Diffusion, utilizando imagens e *prompts* com identificadores únicos, garantindo a correspondência entre texto e imagem. São utilizados três agentes LLM responsáveis por organizar a cena: (i) Amostrador de Instâncias, equilibrando a presença de classes de objetos; (ii) Compositor de Cenas, definindo condições meteorológicas e horários; e (iii) Descritor de Cenas, adicionando elementos meteorológicos e iluminação. Métodos de adaptação de domínio não supervisionada, como DAFormer [50] e HRDA [51], ajudam a mitigar a ausência de etiquetas semânticas a nível de pixel.

Nos experimentos, o WeatherDG foi treinado com o conjunto de dados Cityscapes [2] como domínio de origem e os conjuntos de dados ACDC [52], BDD100k [1] e DarkZurich [53] como domínios-alvo, combinando imagens reais e sintéticas. Os resultados obtidos demonstram uma melhoria na generalização da segmentação em condições adversas, como nevoeiro, noite e precipitação, permitindo distinguir elementos como passeios, céu e peões de forma mais eficaz do que as abordagens anteriores.

Neste trabalho, o WeatherDG serve de referência para as técnicas de geração e adaptação de domínio. No entanto, não é aplicado diretamente, uma vez que se concentra na geração de imagens sintéticas a partir do zero. Em contraste, esta dissertação foca-se na transformação de imagens, preservando a estrutura.

3.3. Modelos de Segmentação Semântica

No âmbito das tarefas de visão computacional, a segmentação de imagens assume um papel de extrema importância, uma vez que permite a identificação e classificação de objetos ao nível de pixel. Nesta secção, serão apresentados os modelos de segmentação relevantes no contexto deste trabalho.

3.3.1 DeepLabv3+

O DeepLabv3+, proposto por Chen et al. (2018) [54], é um modelo avançado de segmentação semântica que combina o módulo de agrupamento piramidal espacial, responsável por capturar contexto em múltiplas escalas, com uma arquitetura codificador-decodificador, que melhora a recuperação de detalhes e a definição dos contornos dos objetos. A inovação mais significativa consiste na introdução de um decodificador simples e eficiente, que refina os mapas de segmentação ao combinar informações de alto e baixo nível, aprimorando notavelmente a precisão nas bordas.

O modelo recorre a convoluções atrous, que são convoluções ”dilatadas”que aumentam o campo de visão do kernel sem reduzir a resolução da imagem, permitindo capturar um contexto amplo sem perder detalhes, bem como convoluções separáveis em profundidade, que reduzem o custo computacional. A arquitetura é fundamentada na Aligned Xception [55], otimizada para segmentação, o que confere ao modelo maior rapidez e precisão.

Nos experimentos, o DeepLabv3+ alcançou resultados de ponta, atingindo 89% de mIoU no PASCAL VOC 2012 [56] e 82,1% no Cityscapes [2], superando métodos anteriores.

Não obstante, persistem desafios relacionados com objetos parcialmente ocultos, classes pouco representadas e categorias visualmente semelhantes.

No âmbito deste estudo, o modelo DeepLabv3+ foi utilizado para a segmentação de imagens de estrada, possibilitando a extração de mapas semânticos confiáveis. A opção por este modelo deve-se fundamentalmente à existência de uma versão pré-treinada no conjunto de dados BDD100K, utilizado para avaliar os modelos generativos, o que assegura uma maior conformidade com os dados.

3.3.2 *High-Resolution Networks* (HRNet)

A HRNet (*High-Resolution Network*), concebido por Wang et al. (2020) [57], apresenta uma abordagem inovadora para tarefas de visão computacional, tais como a segmentação semântica, a deteção de objetos e a estimativa de pose, ao preservar representações de alta resolução ao longo de toda a rede. Esta abordagem contrasta com as arquiteturas tradicionais, que reduzem a resolução e posteriormente tentam recuperá-la.

A rede recorre a diversos fluxos paralelos que operam em diferentes resoluções, procedendo à troca de informações através de módulos de fusão que executam operações de *downsampling* e *upsampling*. Esta abordagem permite a integração eficiente de características em várias escalas. No âmbito da segmentação semântica, o HRNetV2 gera representações detalhadas que são submetidas a um classificador para criar mapas de segmentação, expandidos por interpolação para o tamanho original da imagem.

Avaliada em conjunto de dados como Cityscapes [2], PASCAL-Context [58] e LIP [59], a HRNetV2 demonstrou superioridade em relação a métodos consagrados, tais como UNet++ [60], DeepLabv3 [61] e PSPNet [62], alcançando maior precisão com menor custo computacional e menos parâmetros, inclusive na segmentação de partes humanas.

No âmbito desta dissertação, o modelo em questão foi descartado, uma vez que não disponibiliza uma versão pré-treinada no conjunto de dados BDD100K. Deste modo, seria necessário um processo de treino a partir do zero ou uma adaptação a partir de outros conjuntos de dados, o que aumentaria os custos e o tempo de desenvolvimento.

3.4. Conjunto de Dados

Para o desenvolvimento de modelos de visão computacional em condução autónoma, é essencial dispor de conjuntos de dados diversificados. Na presente secção, são apresentados os principais conjuntos de dados utilizados neste domínio.

3.4.1 *The Cityscapes Dataset*

O Cityscapes [2] consiste num conjunto de dados e *benchmark* de grande escala, desenvolvido para a rotulagem semântica e de instância ao nível do pixel em cenas urbanas. Este conjunto de dados foi criado com o objetivo de colmatar a falta de conjuntos de dados realistas e complexos nesta área. O autor reuniu 5.000 imagens com anotações detalhadas e 20.000 imagens com anotações grosseiras, capturadas em 50 cidades europeias, principalmente na Alemanha, durante a primavera, o verão e o outono, evitando condições meteorológicas adversas. As imagens foram obtidas com uma câmara estéreo automotiva de alta qualidade e incluem informações adicionais, como GPS e odometria. O conjunto de dados inclui 30 classes organizadas em oito categorias, das quais 19 são utilizadas para avaliação.

As imagens densamente anotadas foram divididas em 2.975 para treino, 500 para validação e 1.525 para teste, mantendo o equilíbrio geográfico e diversidade urbana.

Este conjunto de dados não foi utilizado neste trabalho. A sua referência, nesta revisão, justifica-se pelo reconhecimento generalizado no domínio da condução autónoma e pelo facto de evidenciar o desequilíbrio entre imagens em condições favoráveis e desfavoráveis, problema central que se pretende mitigar.

3.4.2 *Berkeley DeepDrive 100K* (BDD100K)

Os conjuntos de dados tradicionais empregues na condução autónoma evidenciam limitações em termos de diversidade visual, distribuição geográfica e abrangência das anotações, o que dificulta o treino de modelos robustos. Para superar esses desafios, foi concebido o BDD100K pelos investigadores das universidades da Califórnia em Berkeley, Cornell e San Diego [1]. Este conjunto contém vídeos de condução, com 100 mil gravações de alta resolução, com uma duração aproximada de 40 segundos, acompanhadas de dados GPS/IMU e anotações detalhadas. O estudo em questão foi realizado através de *crowdsourcing*, com o apoio da Nexar, abrangendo um conjunto de mais de 50 mil viagens em diferentes regiões dos EUA. As vias analisadas foram de diversos tipos, assim como as condições meteorológicas e os horários.

Organizado em conjuntos de treino, validação e teste, o BDD100K é notável pela sua diversidade, abrangendo imagens rotuladas segundo seis condições meteorológicas, seis tipos de cena e três períodos do dia, incluindo cenários como neve e chuva. As anotações em questão abrangem uma série de tarefas, nomeadamente a deteção de objetos,

a identificação de marcações de faixas, a categorização de áreas transitáveis, a segmentação semântica com 40 classes, o rastreamento de objetos com mais de 130 mil trajetórias e 3,3 milhões de caixas delimitadoras.

Este conjunto de dados, dispõe de um sub-conjunto reduzido, o BDD10K. Na presente dissertação, foi utilizada essa versão para testar os modelos generativos. A seleção deste conjunto deve-se à sua heterogeneidade e capacidade para ilustrar o problema central desta dissertação, o desequilíbrio das condições meteorológicas.

3.4.3 *Detection in Adverse Weather Nature* (DAWN)

O DAWN [63] é um conjunto de dados desenvolvido para otimizar a deteção de veículos em condições meteorológicas desfavoráveis, tais como nevoeiro, chuva, neve e tempestades de areia. Em contraste com os conjuntos de dados amplamente utilizadas, tais como o KITTI [64] e o MS-COCO [65], que se restringem a imagens em condições normais, o DAWN disponibiliza 1000 imagens reais capturadas em autoestradas, vias urbanas e cruzamentos em diferentes países, totalizando 7.845 caixas delimitadoras anotadas manualmente para cinco categorias de veículos e pessoas. As imagens foram obtidas por meio de buscas online e apresentam grande diversidade em termos de iluminação, oclusão, densidade de tráfego e ângulos de captura, tornando um conjunto representativo de cenários desafiadores.

No presente trabalho, o conjunto de dados DAWN revelou-se particularmente útil para o treino do CycleGAN, fornecendo imagens reais capturadas em condições adversas.

3.4.4 *Adverse Conditions Dataset with Correspondences* (ACDC)

O ACDC [52] é um conjunto de dados com 4006 imagens, distribuídas equitativamente por quatro condições adversas — nevoeiro, noite, chuva e neve —, cada uma acompanhada de anotações semânticas detalhadas, imagens correspondentes em condições normais e máscaras binárias que indicam regiões com conteúdo claro ou incerto.

As imagens foram captadas na Suíça, maioritariamente em áreas urbanas, mas também em autoestradas e zonas rurais, recorrendo a uma GoPro Hero 5 com posicionamento GPS para alinhar trajetos normais e adversos, garantindo correspondência precisa entre as cenas. O conjunto de dados foi meticulosamente selecionado para assegurar a diversidade e foi dividido em treino, validação e teste. As anotações são compatíveis com o

Cityscapes e seguem um protocolo bifásico que combina imagens normais e adversas, permitindo a segmentação tradicional e sensível à incerteza.

Tal como o DAWN, o conjunto de dados ACDC forneceu imagens reais em condições adversas, contribuindo, neste estudo, para o treino do CycleGAN.

3.5. Síntese da Revisão do Estado de Arte

A metodologia deste estudo centrou-se no desenvolvimento de uma pipeline robusta para gerar imagens sintéticas de estradas sob diferentes condições meteorológicas, garantindo realismo e preservação estrutural. Para tal, com base no estudo da literatura, foram selecionadas duas abordagens complementares: o CycleGAN, eficiente na tradução de imagens entre domínios não emparelhados, e o Stable Diffusion com ControlNet, capaz de gerar imagens de alta qualidade com controlo preciso sobre a composição e a estrutura.

Para os conjuntos de dados, foi utilizada a versão reduzida do BDD100K, o BDD10K, representativo do desequilíbrio entre condições favoráveis e desfavoráveis. Os modelos ACDC e DAWN forneceram imagens reais em condições desfavoráveis, as quais se revelaram úteis para treinar o CycleGAN a gerar imagens consistentes nestas condições.

No que diz respeito à segmentação semântica, a opção recaiu sobre o DeepLabV3+, que alia eficiência e precisão, usufruindo de uma versão pré-treinada no BDD100K, o que evita treinos iniciais e reduz o tempo e o custo de desenvolvimento.

4. Transformação de Imagens para Condições Meteorológicas Adversas

O presente estudo propõe a utilização de modelos de geração de imagens como solução para o desequilíbrio dos conjuntos de dados de condução autónoma, criando versões realistas de vias rodoviárias sob diversas condições meteorológicas.

Para o efeito, procedeu-se à avaliação e comparação de dois modelos de geração de imagens: um modelo de difusão condicional, o Stable Diffusion [3], e um modelo baseado em redes generativas adversariais, o CycleGAN [5], previamente mencionados no capítulo anterior. Ambos são adequados para a transformação de imagens, funcionando como uma ferramenta prática para aumentar a diversidade de cenários nos conjuntos de dados.

Nas próximas secções do presente capítulo, descrevem-se os procedimentos metodológicos aplicados para a geração das imagens, incluindo a implementação e utilização do modelo Stable Diffusion, a preparação dos dados e o treino do modelo CycleGAN. Adicionalmente, serão apresentados os procedimentos experimentais realizados para cada modelo, bem como a metodologia adotada para a avaliação das imagens geradas.

4.1. Estruturação do Conjunto de Dados para a Geração Sintética

Com o intuito de demonstrar a utilidade da transformação de imagens como ferramenta para mitigar o desequilíbrio meteorológico em conjuntos de dados de condução autónoma, foi selecionado um conjunto de dados que apresenta essa desproporção, o BDD10K, um subconjunto do BDD100K. As imagens deste conjunto serão utilizadas nos modelos de geração para aplicar diferentes transformações, visando equilibrar o conjunto de dados.

O BDD100K (*Berkeley DeepDrive 100K*) é um dos maiores conjuntos de dados públicos para tarefas de visão computacional em condução autónoma, composto por 100.000 imagens anotadas de cenários urbanos e suburbanos. Este conjunto disponibiliza um subconjunto denominado BDD10K, composto por 10.000 imagens, que mantém a diversidade e a estrutura do conjunto de dados completo, mas com um volume de dados mais reduzido.

No âmbito deste trabalho, e em virtude das limitações de armazenamento e processamento, decidiu-se empregar o subconjunto BDD10K. Neste subconjunto, foi possível verificar que, das 10.000 imagens analisadas, 8.287 foram classificadas como apresentando boas condições meteorológicas, enquanto apenas 1.713 imagens foram consideradas como apresentando condições meteorológicas adversas:

- 414 imagens retratam chuva;
- 528 imagens com a presença de neve;
- 209 imagens incluem nevoeiro;
- 562 imagens noturnas.

Esta desproporção evidencia um desequilíbrio significativo, o que justifica a aplicação de técnicas de geração sintética para aumentar a diversidade e representar melhor os cenários menos comuns.

Com o propósito de alcançar um equilíbrio no conjunto de dados, foi estabelecida uma divisão equitativa: 5.000 imagens em condições normais e 5.000 em condições adversas. Estas últimas foram distribuídas em quatro categorias principais: precipitação, neve, nevoeiro e noite, com 1.250 imagens em cada, totalizando as 5.000.

Para tal, foi realizada uma análise manual do conjunto de dados, com vista a identificar e separar as imagens que já apresentavam naturalmente condições meteorológicas adversas. Após a contabilização das imagens reais, as restantes necessárias para completar as 1.250 em cada categoria foram selecionadas entre aquelas com boas condições, para serem posteriormente convertidas sinteticamente nas condições adversas em falta. Resultando na seguinte divisão:

- 836 imagens convertidas para a condição chuva;
- 722 imagens convertidas para a condição neve;
- 1041 imagens convertidas para a condição nevoeiro;
- 688 imagens convertidas para a condição noite.

Adicionalmente, foram selecionadas quatro imagens que foram transformadas em todas as condições meteorológicas, de modo a permitir a comparação direta da mesma cena em diferentes contextos.

Durante o processo de triagem, foram eliminadas manualmente as imagens com dimensões incompatíveis ou com baixa qualidade visual, garantindo a consistência e a legibilidade do conjunto de dados final.

No final, obteve-se um conjunto de dados equilibrado, que serviu de base para testar e comparar o desempenho dos modelos de geração na transformação de imagens em diferentes condições meteorológicas.

4.2. Transformação de Imagens com o Stable Diffusion

O Stable Diffusion, previamente descrito no Capítulo 3, constitui um modelo avançado de geração de imagens que se fundamenta em modelos de difusão latente. Este modelo recorre a um processo estocástico reverso para gerar imagens detalhadas e coerentes a partir de ruído aleatório. A Figura 4.1 apresenta uma representação esquemática do processo de difusão latente, evidenciando as fases de codificação, difusão direta, remoção de ruído através de uma rede U-Net condicionada, e decodificação final da imagem.

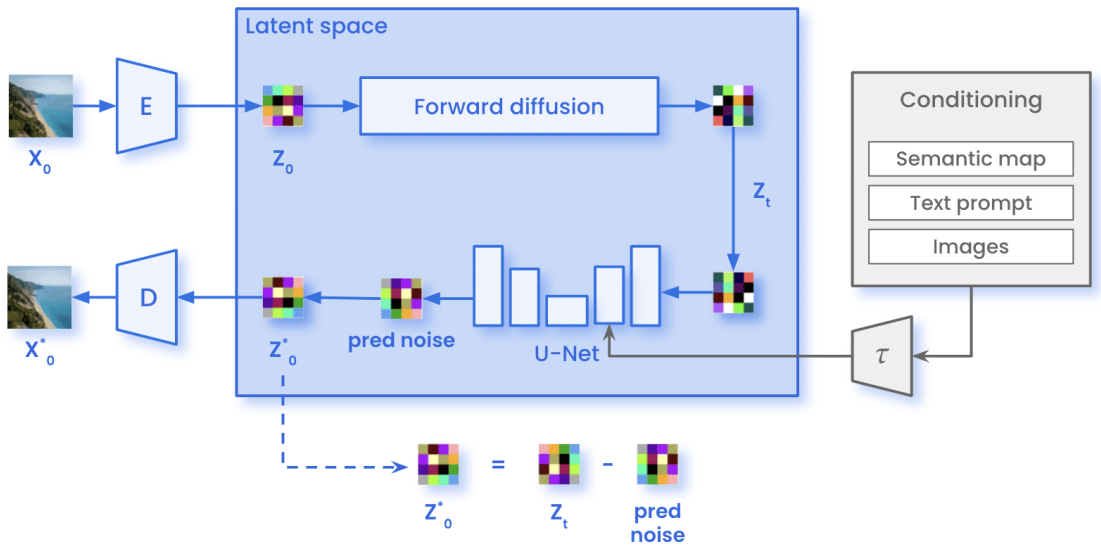


Figura 4.1: Representação do processo de difusão de imagens utilizado pelo Stable Diffusion. [66]

Com o objetivo de transformar imagens reais de estradas em diferentes condições meteorológicas, mantendo a estrutura e os elementos originais da cena, foi desenvolvida uma pipeline baseada no Stable Diffusion.

Neste contexto, o modelo utiliza a imagem de entrada como referência, aplicando alterações controladas apenas no aspeto meteorológico, com base numa descrição textual

(*prompt*) e em mapas de controlo. Para isso, a entrada condicional da pipeline é composta por:

- Imagem Base: uma imagem real de estrada e em boas condições meteorológicas;
- *Prompt* textual: uma descrição curta da condição climática desejada, por exemplo, “a rainy road” ou “snowy day on a street”, interpretada pelo modelo para guiar a transformação;
- Mapas de controlo do ControlNet: mapas estruturais que orientam a geração para que a estrutura da imagem base não sofra alterações.

A imagem base será, portanto, transformada pelo modelo Stable Diffusion de forma a refletir a condição climática especificada no *prompt*, o objetivo é que ocorra apenas alteração nos elementos visuais necessários e preservação da estrutura e dos objetos presentes na imagem original. Contudo, para garantir a manutenção total da estrutura da imagem, é necessário recorrer a controladores adicionais, como o ControlNet, que auxiliam o modelo a respeitar a geometria e os elementos da cena durante o processo de transformação.

Neste trabalho, foram utilizados modelos pré-treinados, o Realistic Vision v5.1 [67], e os módulos ControlNet [4], para construir uma pipeline eficaz e reprodutível de geração de imagens sintéticas.

4.2.1 *Realistic Vision*

Inicialmente, foram realizados testes com os modelos Stable Diffusion v1.5 e v2.1 [68], bem como com algumas variantes pré-treinadas destes modelos. No entanto, os resultados obtidos revelaram limitações ao nível da qualidade visual e da fidelidade estrutural das imagens geradas. Também foi considerado o modelo Stable Diffusion XL (SDXL) [69], devido à sua superior capacidade de geração de detalhes, mas a elevada exigência de recursos computacionais e a instabilidade associada ao ControlNet tornaram a sua utilização inviável neste trabalho.

Por conseguinte, foi realizada uma pesquisa sobre modelos alternativos na plataforma Hugging Face [70], tendo sido identificado o Realistic Vision v5.1 [67] como uma opção consolidada na comunidade. Deste modo, optou-se por utilizar este modelo, que consiste

numa variante do Stable Diffusion, especificamente treinada para produzir imagens com um elevado grau de realismo visual.

O modelo Stable Diffusion padrão já obtém resultados expressivos na geração de imagens a partir de descrições textuais. No entanto, o modelo Realistic Vision melhora essa capacidade ao focar-se em aspetos como a textura, a iluminação e a profundidade, tornando as imagens geradas mais semelhantes a fotografias reais.

Este modelo foi treinado com grandes conjuntos de dados contendo imagens de elevada qualidade, abrangendo diversos estilos, objetos e cenários, o que permite ao Realistic Vision generalizar bem para uma ampla gama de contextos visuais.

Neste projeto, o Realistic Vision foi utilizado para garantir que as imagens geradas apresentassem um elevado realismo e qualidade visual.

Apesar do satisfatório grau de realismo alcançado com o Realistic Vision, verificou-se que a fidelidade estrutural da imagem base poderia ser comprometida em determinadas transformações. Para resolver o problema em questão, foi integrada a arquitetura ControlNet.

4.2.2 ControlNet

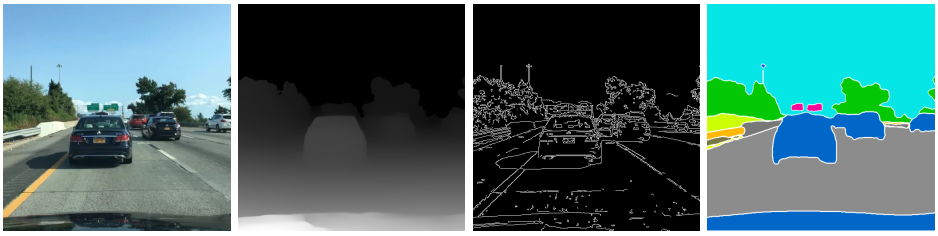
Para garantir a preservação da estrutura e dos elementos essenciais da imagem base durante a geração de imagens, mesmo quando é submetida a alterações significativas, foi utilizado a arquitetura ControlNet [4].

Nos modelos de difusão tradicionais, embora seja possível condicionar a geração com base em descrições textuais ou imagens de referência, o controlo sobre a estrutura e o conteúdo semântico da imagem base ainda é limitado. Esta situação pode resultar na produção de imagens com boa qualidade visual, contudo, com a distorção de elementos significativos, tais como a disposição de veículos ou faixas de rodagem.

O ControlNet resolve este problema ao facultar ao modelo de difusão informações suplementares sobre a estrutura da imagem de entrada, através de mapas condicionantes. Estes mapas atuam como guias que direcionam o processo de difusão, garantindo que a imagem gerada mantenha a geometria e o conteúdo semântico dos objetos presentes no cenário original.

Neste trabalho, foram empregues três tipos de condicionamento em simultâneo: Depth, Canny e Seg. Cada um fornece um tipo complementar de informação extraída da imagem original, e em conjunto permitem que o modelo preserve a estrutura durante a tradução para novas condições meteorológicas:

- O condicionamento Depth utiliza um mapa de profundidade gerado a partir da imagem de entrada. Este mapa representa a distância entre cada pixel e a câmara, fornecendo ao modelo uma noção tridimensional da cena;
- O condicionamento Canny baseia-se na aplicação do algoritmo de deteção de arestas de Canny [71], que extrai os contornos principais da imagem de entrada. Este mapa disponibiliza uma representação simplificada dos limites dos objetos e das estruturas visuais, garantindo que as formas básicas dos elementos da cena sejam preservadas na imagem final;
- O condicionamento Seg utiliza um mapa de segmentação semântica, no qual cada pixel da imagem de entrada é associado a uma classe de objeto (por exemplo, estrada, veículo, peão, entre outros). A preservação da identidade dos elementos da cena e da sua localização é fundamental para garantir que, por exemplo, um veículo identificado na imagem original mantenha a sua identidade após a tradução climática.



(a) (b) (c) (d)

Figura 4.2: Exemplo dos mapas de condicionamento utilizados pelo ControlNet. (a) Imagem original; (b) Mapa de profundidade (Depth), que fornece informações sobre a distância dos objetos; (c) Mapa de contornos (Canny), que destaca as bordas e estruturas principais da cena; (d) Mapa de segmentação semântica (Seg), que classifica e localiza os diferentes elementos da imagem.

Conforme ilustrado na Figura 4.2, os três mapas de condicionamento fornecem informações complementares sobre a cena original, permitindo que o ControlNet preserve simultaneamente a geometria, a estrutura e o conteúdo semântico das cenas.

Os três mapas são fornecidos em simultâneo ao Stable Diffusion durante o processo de difusão, na fase de difusão reversa, influenciando diretamente a remoção de ruído de forma a respeitar a estrutura da imagem base. Em cada geração, o modelo utiliza as informações referidas para efetuar os ajustes necessários à adição e remoção de ruído, conduzindo a imagem a um estado final.

4.2.3 Implementação do pipeline

Para implementar a pipeline descrita, foi desenvolvido um código em Python para cada condição climática, adaptando os *prompts* textuais e as funções de pós-processamento de acordo com as características visuais desejadas. Cada código implementa o fluxo completo de geração, garantindo que as particularidades de cada cenário sejam fielmente representadas. Esta abordagem permitiu controlar individualmente os elementos visuais de cada clima.

A implementação do pipeline foi desenvolvida com a biblioteca diffusers [72], concebida pela plataforma Hugging Face [70], que disponibiliza uma interface modular para a utilização de modelos como o Stable Diffusion. Esta biblioteca permitiu integrar o modelo Realistic Vision v5.1 e os três modelos auxiliares da arquitetura ControlNet, responsáveis por condicionar a geração. Cada ControlNet foi carregado separadamente, com precisão reduzida em float16 para otimizar o uso de GPU, e posteriormente combinados por meio do componente MultiControlNetModel, permitindo que múltiplos mapas de controlo fossem utilizados de forma simultânea pela pipeline.

Para gerar os mapas de controlo necessários à operação do ControlNet, foram implementadas funções específicas:

- A função `get_canny`, aplica o detetor de bordas de Canny [71] à imagem original;
- A função `get_segmentation_mask` aplica a segmentação semântica, utilizando o modelo UperNet [73], disponibilizado pela biblioteca Transformers [70], também da Hugging Face, que gera um mapa de classes para os diferentes elementos da imagem.

Em relação ao mapa de profundidade (Depth), não foi realizado nenhuma função específica. No pipeline, a imagem original é utilizada diretamente como entrada para o ControlNet de Depth. O modelo é responsável pela interpretação da imagem e pela geração das informações de profundidade.

Após o carregamento destes modelos, foram implementadas funções auxiliares de pós-processamento visual, executados através das bibliotecas Pillow e NumPy, para realçar visualmente a condição climática simulada na imagem gerada.

Apesar da capacidade do modelo de incorporar diversas características visuais relacionadas com condições meteorológicas, foi observado que a necessidade de um controlo rígido, necessário para que a estrutura da imagem permanecesse inalterada, impôs um compromisso. Por um lado, se o modelo tiver uma maior liberdade criativa, conseguirá inserir a condição climática de forma muito evidente, mas acabará por deformar a estrutura da imagem. Por outro lado, se tiver um maior controlo, conseguirá preservar a estrutura original, mas limitará a criatividade e, consequentemente, os efeitos meteorológicos serão adicionados de forma discreta.

O equilíbrio encontrado privilegiou a preservação da estrutura da cena, resultando em efeitos climáticos mais subtis, dessa forma foi necessário aplicar funções de pós-processamento, para evidenciar melhor as características das condições meteorológicas. Entre essas funções, estão:

- Escurecimento gradual da zona superior da imagem, aplicada no contexto da condição chuva, simulando a presença de nuvens densas e reduzindo a luminosidade do céu.

  Esta função opera diretamente sobre os valores dos pixéis RGB, aplicando uma máscara de intensidade variável que escurece progressivamente a imagem desde o topo até aproximadamente 40% da sua altura total. A máscara é construída com base na altura da imagem, atribuindo maior intensidade de escurecimento às linhas superiores e diminuindo gradualmente esse efeito até desaparecer;

- Adição de efeitos visuais de chuva, através da sobreposição de linhas que imitam gotas de água em queda, com direção e brilho controlados para parecerem naturais.

  Para esse efeito, foi construída uma camada adicional do mesmo tamanho da imagem original, onde são delineadas linhas com comprimento, espessura, inclinação e posição definidas, simulando gotas a cair com diferentes intensidades. O número de gotas é proporcional à dimensão da imagem e ao parâmetro de densidade definido.

Posteriormente, é aplicado um filtro de convolução, que suaviza ligeiramente as linhas, criando um efeito mais natural. Por fim, a camada de chuva é fundida com a imagem original através de uma média ponderada dos seus valores de pixel.

- Adição de um tom frio que modifica o canal azul da imagem, na implementação da condição neve, de modo a enfatizar uma tonalidade fria, alinhada com o cenário invernal.

  Esta função atua diretamente sobre os canais de cor da imagem, amplificando ligeiramente a intensidade do canal azul (canal B do espaço RGB), responsável por transmitir sensações visuais de frio.

No caso das condições de nevoeiro e noite, não se revelou necessária a aplicação de funções adicionais de pós-processamento visual. O modelo com os condicionamentos estruturais mostrou-se adequado para representar de forma convincente a atmosfera enevoada e escurecida.

O fluxo principal de execução do pipeline (`process_image`) consiste nas seguintes etapas:

1. Abertura da imagem de entrada e criação dos mapas de controlo correspondentes;
2. Execução da pipeline de difusão utilizando a imagem inicial e os mapas do ControlNet, juntamente com um *prompt* textual, que descreve a cena desejada, e com parâmetros que influenciam a geração, tais como o número de passos, o peso do *prompt* e a intensidade do efeito dos mapas do ControlNet. Neste processo a imagem é gerada;
3. Aplicação das funções de pós-processamento na imagem gerada.

Para permitir a execução eficiente em GPU, recorreu-se à biblioteca torch, PyTorch [74], que fornece o suporte necessário para carregar os modelos com precisão reduzida, float16, e gerar resultados com maior desempenho computacional.

Todas as bibliotecas citadas foram obtidas através do gestor de pacotes pip, encontrando-se disponíveis para utilização.

Este processo possibilita uma tradução climática realista e estruturada da imagem, preservando a composição e a semântica da cena original, ao mesmo tempo que adiciona elementos visuais coerentes com a condição climática indicada.

4.2.4 Procedimentos Experimentais do Stable Diffusion

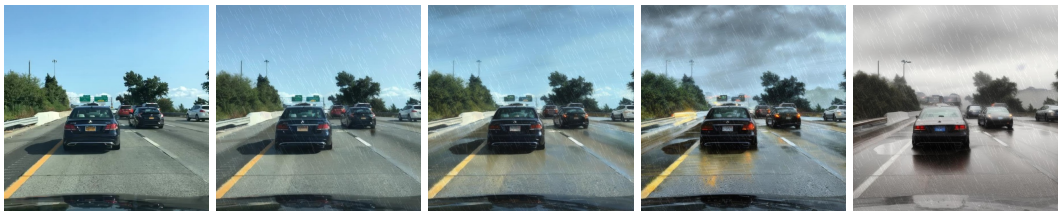
Durante a execução da pipeline, foram definidos os seguintes parâmetros, que controlam a intensidade da transformação e a qualidade da imagem:

- `strength = 0.60`

  Este parâmetro regula o grau de transformação da imagem original. Um valor de 0 indica que a imagem será preservada integralmente, enquanto um valor próximo de 1 permite uma transformação quase total.

  O valor de 0,60 foi selecionado de forma a assegurar um equilíbrio entre a preservação da estrutura da imagem original e a aplicação dos efeitos meteorológicos.

  A Figura 4.3 ilustra a influência da variação do parâmetro *strength* na imagem. É possível observar que a imagem (b), com um *strength* reduzido, praticamente não apresenta alterações, ao passo que a imagem (e), com um *strength* elevado, exibe uma diferença significativa.



(a) Imagem original | (b) `strength = 0.10` | (c) `strength = 0.40` | (d) `strength = 0.60` | (e) `strength = 1`

Figura 4.3: Influência do parâmetro *strength* na geração da imagem, variando de valores baixos (b) até valores elevados (e).

- `num_inference_steps = 60`

  O presente parâmetro define o número de passos inerente ao processo de difusão reversa. Em geral, valores mais elevados tendem a produzir imagens de maior qualidade e detalhe, mas implicam um maior custo computacional.

  O valor 60 mostrou-se apropriado para alcançar um equilíbrio entre a qualidade visual e a eficiência computacional.

  A Figura 4.4 mostra como a variação do número de passos de inferência afeta a geração de imagens. Apesar de as diferenças serem subtis, observa-se que a imagem (b), produzida com menos passos, apresenta menor nível de detalhe em comparação com a imagem (e), obtida com um número maior de passos.



(a) Imagem original | (b) `inference_steps = 10` | (c) `inference_steps = 60` | (d) `inference_steps = 110` | (e) `inference_steps = 210`

Figura 4.4: Influência do parâmetro do número de passos de inferência na geração da imagem, variando de valores baixos (b) até valores elevados (e).

- `guidance_scale = 9`

  Este parâmetro refera-se ao *Classifier-Free Guidance Scale* e é responsável pelo controlo do peso do *prompt* textual durante o processo de geração da imagem. Valores mais baixos aumentam a criatividade e uma consequente diminuição da fidelidade ao *prompt*, enquanto valores altos promovem maior aderência ao texto.

  O valor 9 foi selecionado para garantir que os elementos descritos no *prompt* fossem incorporados na imagem, mantendo simultaneamente uma certa liberdade criativa para evitar rigidez artificial.

  A Figura 4.5 ilustra o impacto da variação deste parâmetro na geração de imagens. Observa-se que a imagem (b), com um valor reduzido, apresenta menos características meteorológicas por não seguir de forma rigorosa o *prompt*. Em contraste, a imagem (e), com um valor elevado, exibe características meteorológicas exageradas, resultando num aspeto pouco realista.



(a) Imagem original | (b) `guidance_scale = 1` | (c) `guidance_scale = 9` | (d) `guidance_scale = 20` | (e) `guidance_scale = 60`

Figura 4.5: Influência do parâmetro *Classifier-Free Guidance Scale*, variando de valores baixos (b) até valores elevados (e).

- `seed = 3214`

  A utilização de uma seed fixa permite, fundamentalmente, assegurar a reprodutibilidade dos resultados. A seleção do valor foi efetuada de forma arbitrária. A

prioridade foi assegurar que as entradas permanecessem constantes, facilitando a comparação dos resultados obtidos.

- `controlnet_conditioning_scale = [0.7, 0.7, 0.7]`

  Este parâmetro viabiliza o ajuste do peso ou da influência de cada um dos mapas de controlo (Depth, Canny, Seg) na geração final.

  O valor 0,7 foi selecionado para os três mapas, proporcionando um nível considerável de controlo, sem suprimir totalmente a criatividade do modelo. A aplicação do mesmo valor a todos os elementos garante um condicionamento equilibrado.

  A Figura 4.6 demonstra o impacto do peso do ControlNet na geração de imagens. Na imagem (b), com um peso igual a 0, a estrutura da imagem original não é preservada. À medida que o valor aumenta, a geração passa a respeitar progressivamente a estrutura original.



(a) Imagem original | (b) `controlnet = 0.0` | (c) `controlnet = 0.5` | (d) `controlnet = 0.7` | (e) `controlnet = 1.0`

Figura 4.6: Influência do peso dos mapas do ControlNet na imagem gerada, variando de valores baixos (b) até valores elevados (e).

Apesar das diferenças visuais entre os cenários de chuva, neve, nevoeiro e noite, os parâmetros utilizados no pipeline de geração foram mantidos constantes em todas as condições meteorológicas. A uniformidade justifica-se, dado que os valores inicialmente definidos revelaram ser suficientemente robustos para produzir resultados satisfatórios em todos os contextos.

Estes parâmetros foram ajustados com base em apenas algumas imagens de teste e, posteriormente, aplicados uniformemente a todas as imagens do conjunto de dados que foram transformadas. Embora esta abordagem tenha apresentado resultados satisfatórios de forma geral, como será demonstrado no Capítulo 5, a adaptação dos parâmetros a cada imagem individualmente poderia ter gerado resultados ainda mais precisos. No entanto, este procedimento não foi exequível devido à complexidade e ao volume de imagens.

Para realizar testes exploratórios e obter uma perceção visual rápida da performance do modelo, recorreu-se ao AUTOMATIC1111 Stable Diffusion Web UI [75], uma interface gráfica que permite interagir com modelos de Stable Diffusion de forma prática e acessível.

Esta ferramenta possibilitou a execução de múltiplas experiências preliminares com diversos modelos base (SD v1.5, SD v2.1, SDXL, Realistic Vision), a geração por *prompt*, o controlo de parâmetros (*guidance scale*, *steps*, *seeds*, entre outros) e a integração com o ControlNet. Esta interface foi crucial durante a fase experimental para compreender o comportamento dos modelos sob diferentes condições de entrada.

4.2.4.1 *Prompt*

Os *prompts* textuais desempenham um papel fundamental, uma vez que guiam o modelo na criação de imagens com determinadas características visuais e semânticas. Para este projeto, foram utilizados dois tipos de *prompts*: o *prompt* positivo e o *prompt* negativo.

O *prompt* positivo descreve, com termos detalhados e descritivos, os elementos desejados na imagem gerada. Já o negative *prompt* especifica características indesejadas ou artefactos comuns em modelos de geração, tais como distorções anatómicas, aparência artificial ou baixa qualidade visual.

- No contexto da chuva, foram incluídas expressões como ”wet pavement reflecting”, ”realistic raindrops on windshield”e ”overcast”, a fim de induzir elementos associados ao clima chuvoso e à atmosfera cinzenta típica dos dias nublados.

```
prompt = (photorealistic:1.4), (detailed textures:1.3),
(film grain:0.3), Rainy daytime urban highway, wet pavement
reflecting, realistic raindrops on windshield, water streaks, overcast


negative_prompt = cartoon, 3d, doll, bad anatomy, blurry, unrealistic,
plastic, deformed, overexposed, underexposed, low quality
```

- No que se refere à neve, o *prompt* inclui descrições específicas de neve em diferentes elementos da cena, como “snow falling heavily” e “snow accumulation on car roofs”, e também enfatiza o clima frio e húmido, de modo a orientar a geração de texturas e iluminação compatíveis com ambientes de inverno.

Além disso, o ”negative *prompt*”para a neve inclui termos como ”folhas verdes”e ”tempo ensolarado”para evitar a presença de elementos incompatíveis com o cenário de inverno.

```
prompt = (photorealistic:1.4), (detailed textures:1.3),
(film grain:0.3), A highway in winter, snow falling heavily,
snow covering the roadside, trees heavily covered with snow, cars
driving with headlights on, snow accumulation on car roofs and
road edges, cloudy sky, cold atmosphere, realistic lighting, wet
and icy road

negative_prompt = cartoon, 3d, doll, bad anatomy, blurry,
unrealistic, plastic, deformed, overexposed, underexposed,
low quality, green leaves, autumn leaves, foliage,
sunny weather, lush vegetation
```

- No caso do nevoeiro, destacam-se termos como ”thin atmospheric fog”, ”trees fading softly”e ”muted colors”, que incentivam a criação de efeitos visuais característicos do nevoeiro, como a diminuição da nitidez e o desfoque suave dos objetos distantes, bem como a atmosfera fria e húmida típica.

```
prompt = (photorealistic:1.4), (detailed textures:1.3),
(film grain:0.3), a realistic highway on an early foggy morning,
thin atmospheric fog, trees fading softly in the distance,
headlights reflecting on the humid road, muted colors, low contrast
background, natural foggy haze, misty air, overcast sky, cool and
damp ambiance

negative_prompt = cartoon, 3d, doll, bad anatomy, blurry, unrealistic,
plastic, deformed, overexposed, underexposed, low quality
```

- Para a condição noite , foram usado termos como ”urban street at night”que estabelece um ambiente urbano e o período noturno; ”dark sky”, que enfatiza a escuridão

característica do céu noturno e ”streetlights on”, que indica a presença de iluminação artificial nas ruas.

```
prompt = (photorealistic:1.4), (detailed textures:1.3),
(film grain:0.3), nighttime, realistic road scene at night, dark sky,
black sky, no visible sun

negative_prompt = cartoon, 3d, doll, bad anatomy, blurry, unrealistic,
plastic, deformed, overexposed, underexposed, low quality, sun,
sunlight, daytime, bright sky, white sky, blue sky, clouds,
overexposed, high exposure, glare, lens flare, artifacts,
overexposed highlights, white streaks, overexposed sky,
daylight lighting
```

Nos quatro casos, foram adicionados modificadores ponderados ao *prompt* positivo, como ”(photorealistic:1.4)”e ”(detailed textures:1.3)”, que aumentam o peso dos atributos, incentivando o modelo a priorizar o realismo visual e a riqueza de detalhes na textura.

No *prompt* negativo, foram utilizadas, para todas as condições meteorológicas, palavras como ”cartoon”, ”plastic”, ”blurry”ou ”low quality”, com o intuito de afastar uma estética irrealista ou com pouca fidelidade fotográfica.

Após a utilização do Stable Diffusion para a geração de imagens sintéticas condicionadas, foi também explorado o CycleGAN, com vista a avaliar um método alternativo baseado na tradução de domínios.

4.3. Transformação de Imagens com CycleGAN

Para a tarefa de transformação de imagens em diferentes condições meteorológicas, foi testada a utilização do modelo CycleGAN, já descrito no Capítulo 3, uma arquitetura de redes generativas adversariais (GAN) desenvolvida especificamente para a tradução de imagens entre domínios sem utilização de pares correspondentes. A Figura 4.7 ilustra a arquitetura do CycleGAN.

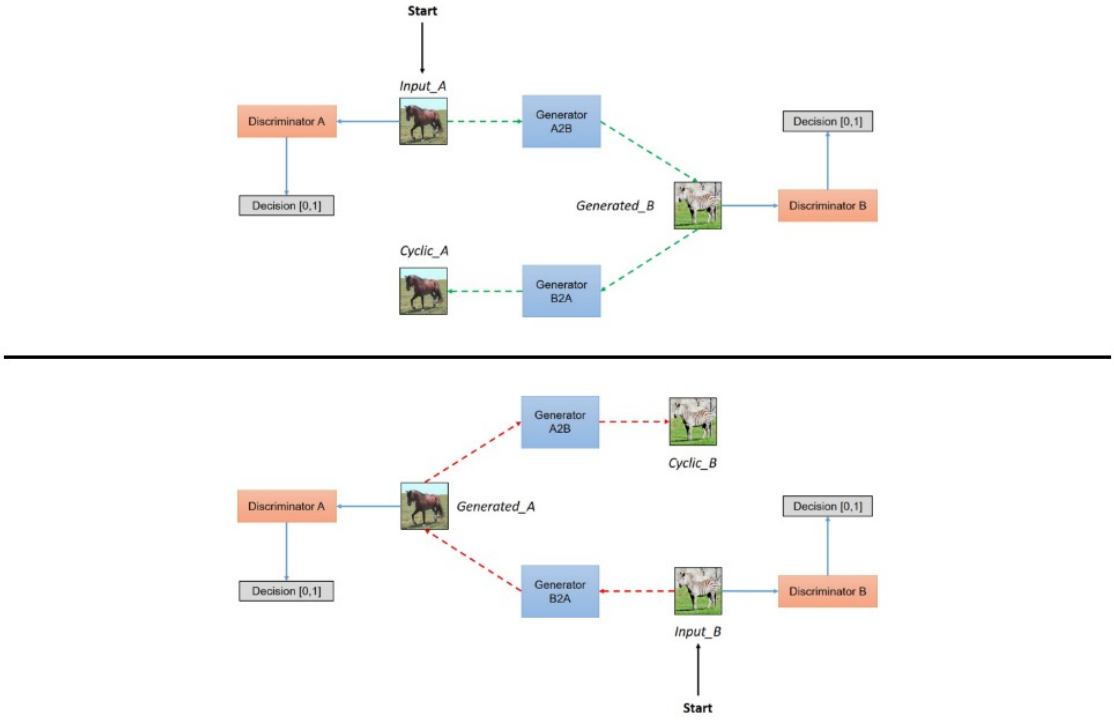


Figura 4.7: Diagrama da arquitetura do CycleGAN. [76]

O modelo CycleGAN utilizado foi obtido a partir do código disponibilizado pelos autores. O código em questão disponibiliza a arquitetura do modelo, os pesos pré-treinados e os procedimentos de treino recomendados, possibilitando a utilização direta.

4.3.1 Configuração e Treino do CycleGAN

O treino do modelo envolveu a criação de dois domínios visuais distintos:

- Domínio A: imagens extraídas do conjunto de dados BDD100K, composto por fotografias reais de condução urbana em boas condições de visibilidade.

  É importante salientar que as imagens de treino foram obtidas exclusivamente do BDD100K e não do BDD10K, de modo a evitar sobreposição com imagens que, posteriormente, pudessem ser transformadas.

- Domínio B: imagens retiradas do conjunto de dados DAWN, que contém exemplos reais de estradas sob chuva, nevoeiro e neve. No entanto, devido ao reduzido número de imagens, este domínio foi complementado com imagens do ACDC, um conjunto de dados realista que inclui diversas condições atmosféricas adversas.

A utilização de várias fontes para este domínio permitiu aumentar a diversidade visual, ao apresentar diferentes intensidades de precipitação, nevoeiro e variações de iluminação.

As imagens foram redimensionadas para 512×512 píxeis e organizadas em pastas separadas, de acordo com a estrutura de diretórios apontada pelo repositório oficial. A opção por uma resolução superior à padrão (256×256) visou melhorar a qualidade visual das imagens geradas e permitir ao modelo aprender pormenores espaciais mais ricos. Esta decisão implicou um custo computacional mais elevado, mas manteve-se viável no contexto dos recursos disponíveis.

Foram realizados quatro treinos independentes, correspondentes às transformações entre o domínio de boas condições de visibilidade e os seguintes domínios-alvo: chuva, neve, nevoeiro e noite. Em todos os cenários, o domínio A foi constituído por 1022 imagens de treino. O número de imagens no domínio B variou consoante a condição meteorológica, conforme resumido na Tabela 4.1. Esta variação deve-se à quantidade de imagens disponíveis em cada condição.

Uma vez que este modelo não utiliza pares correspondentes de imagens, é necessária uma quantidade considerável de imagens para que a rede consiga aprender as distribuições de ambos os domínios, sendo necessárias cerca de 900 a 1000 imagens para gerar transformações mais consistentes.

Tabela 4.1: Número de imagens utilizadas no treino do CycleGAN por condição atmosférica.

Condição Atmosférica	**Domínio A (imagens)**	**Domínio B (imagens)**
Chuva	1022	941
Neve	1022	986
Nevoeiro	1022	987
Noite	1022	900

O modelo foi treinado com PyTorch, utilizando o código oficial disponibilizado pelos autores do CycleGAN sem alterações à arquitetura nem aos hiperparâmetros, dado que os valores definidos estão amplamente validados em estudos anteriores e são adequados a esta tarefa.

Durante a fase de teste e avaliação, foram utilizadas as imagens do conjunto BDD10K, conforme explicado na secção 4.1. As imagens foram transformadas para os diferentes domínios atmosféricos (chuva, neve, nevoeiro e noite), de modo a gerar versões sintéticas da mesma cena sob uma condição diferente.

4.3.2 Procedimentos Experimentais do CycleGan

Durante o treino do modelo CycleGAN foram utilizados os seguintes parâmetros:

- Número de épocas: 200

  Uma época corresponde a uma passagem completa por todo o conjunto de dados de treino.

  Nos primeiros 100 ciclos, a taxa de aprendizagem (*learning rate*) manteve-se constante, permitindo ao modelo ajustar os seus pesos de maneira estável e consistente. Nos 100 ciclos seguintes, a taxa de aprendizagem foi reduzida linearmente até atingir o valor zero, num processo conhecido como decaimento da taxa de aprendizagem (*learning rate decay*).

- Taxa de aprendizagem: 0.0002

  A taxa de aprendizagem define o tamanho dos passos que o otimizador dá para atualizar os pesos da rede neural em cada iteração. O valor utilizado é considerado baixo o suficiente para garantir uma aprendizagem estável, sem que o modelo salte para regiões de pesos inadequados, nem tão pequeno que o treino se torne muito lento.

- Otimizador: Adam (β1=0.5, β2=0.999)

  O otimizador é o algoritmo responsável por atualizar os pesos do modelo durante o treino com base no gradiente da função de perda.

  O valor β1 = 0,5 reduz a influência dos gradientes anteriores, promovendo atualizações mais estáveis.

  O valor β2 = 0,999 controla a média móvel dos quadrados dos gradientes, de modo a ajustar a magnitude das atualizações.

- Função de perda

  A função de perda é o critério que o modelo tenta minimizar para aprender a gerar imagens coerentes e realistas.

A perda adversarial (*Adversarial Loss*) obriga os geradores a produzir imagens indistinguíveis das reais, promovendo o realismo.

A perda de consistência cíclica garante que a estrutura da imagem original é preservada durante a transformação e a reconversão.

A perda de identidade incentiva o gerador a preservar características importantes quando a imagem já se encontra no domínio-alvo, ajudando a manter a cor e a textura naturais.

O treino do modelo, para cada condição atmosférica, teve uma duração aproximada de 33 horas.

4.4. Metodologia de Análise e Avaliação

Após a geração das imagens sintéticas por cada abordagem, estas foram submetidas a uma avaliação com vista a verificar a sua qualidade, fidelidade à cena original e a presença dos efeitos meteorológicos desejados.

No presente capítulo, são apresentados os métodos utilizados para avaliar as imagens. A metodologia de análise é composta por três abordagens complementares: a avaliação visual qualitativa, a segmentação semântica e as métricas quantitativas de avaliação da similaridade.

Cada método é detalhado ao longo do capítulo, com a respetiva fundamentação teórica e descrição do processo de implementação adotado para a sua utilização neste trabalho.

4.4.1 Avaliação Qualitativa

A análise qualitativa consistiu na avaliação visual das imagens geradas, observando diretamente as características percebidas, como a fidelidade à cena original, a intensidade e o realismo dos efeitos meteorológicos. Esta avaliação permitiu identificar padrões, pontos fortes e limitações dos modelos a partir da perceção humana, sem recorrer a métricas quantitativas.

4.4.2 Segmentação Semântica

A segmentação semântica foi utilizada para verificar se os elementos presentes nas imagens geradas permanecem corretamente identificados, com base na análise visual das

máscaras. A hipótese é que imagens sintéticas de alta qualidade devem preservar a estrutura da cena (veículos, estrada, edifícios, vegetação, entre outros) de forma suficientemente coerente para que um modelo de segmentação pré-treinado consiga identificar corretamente os elementos da imagem. Num contexto prático, nomeadamente no que diz respeito a sistemas de visão computacional aplicados à condução autónoma, a qualidade da segmentação é um fator crucial para assegurar uma interpretação precisa do ambiente circundante.

Para proceder à segmentação semântica das imagens, inicialmente, recorreu-se ao modelo HRNet (*High-Resolution Network*), previamente referido na secção 3. Contudo, durante o processo de seleção de modelos pré-treinados, foi detetada a indisponibilidade de um modelo HRNet pré-treinado com o conjunto de dados BDD100K, utilizado para a tradução das imagens neste estudo. O único modelo HRNet pré-treinado disponível, no contexto da condução autónoma, é o treinado no conjunto de dados Cityscapes.

A aplicação direta do modelo HRNet pré-treinado no conjunto Cityscapes sobre as imagens do conjunto BDD100K resultou num desempenho insatisfatório, possivelmente devido às diferenças significativas entre os domínios dos dois conjuntos de dados.

Diante das limitações mencionadas, optou-se pela utilização do modelo Deeplabv3+, no qual se encontrou disponível uma versão pré-treinada diretamente no conjunto de dados BDD100K, o que garantiu uma melhor adequação ao domínio de interesse e, consequentemente, resultados superiores em relação ao modelo HRNet pré-treinado num domínio distinto.

O modelo foi aplicado tanto a imagens reais como a imagens sintéticas, permitindo a comparação dos mapas de segmentação obtidos.

4.4.2.1 Implementação da Segmentação Semântica

Para realizar a segmentação semântica das imagens, foi implementado um código em python utilizando o framework MMSegmentation [77], baseado em PyTorch e amplamente utilizado para tarefas de segmentação. Inicialmente, o modelo Deeplabv3+ pré-treinado no conjunto de dados BDD100K foi carregado, juntamente com os metadados do conjunto de dados, como a paleta de cores e os nomes das classes, fundamentais para a interpretação correta das máscaras geradas.

O processo de segmentação consistiu na leitura das imagens de entrada, seguida da execução da inferência pelo modelo, que produziu máscaras com a classificação semântica pixel a pixel. Com o intuito de facilitar a visualização, as máscaras numéricas foram convertidas em imagens coloridas, utilizando a paleta correspondente ao conjunto de dados BDD100K. Estas máscaras foram posteriormente redimensionadas para se alinharem à resolução original das imagens e combinadas com as imagens originais.

A combinação das bibliotecas MMSegmentation [77], mmcv [78], numpy [79] e OpenCV [80] permitiu um fluxo de trabalho adaptável, desde o carregamento do modelo até à geração e arquivo das imagens segmentadas.

4.4.3 Avaliação Quantitativa das Imagens Geradas

A avaliação quantitativa da qualidade das imagens geradas foi realizada através da aplicação de três métricas: SSIM, PSNR e LPIPS. Estas métricas possibilitam a comparação entre imagens sintéticas e imagens reais, mediante a avaliação de diversos aspetos, tais como a similaridade estrutural, a qualidade de reconstrução e a fidelidade percetual. A seguir, descreve-se brevemente cada uma:

- *Structural Similarity Index Measure* (SSIM) [81]:

  Mede a similaridade estrutural entre duas imagens. Em contraste com as métricas baseadas exclusivamente em píxeis, o SSIM incorpora fatores como a luminância, o contraste e a estrutura local. É crucial para verificar se a geometria da estrada e a disposição de objetos como veículos, árvores e edifícios se mantêm após a transformação para uma condição climática diferente. Resultados próximos de 1 indicam uma alta similaridade estrutural.

- *Peak Signal-to-Noise Ratio* (PSNR) [82]:

  Avalia a qualidade de reconstrução das imagens comparando a intensidade dos píxeis. Embora se trate de uma métrica tradicional, é sensível a pequenas variações e menos alinhada à perceção visual humana. É útil para quantificar diferenças objetivas e distorções introduzidas pela transformação, servindo como referência técnica complementar às métricas percetuais. Resultados mais altos indicam menor distorção.

- *Learned Perceptual Image Patch Similarity* (LPIPS) [29]:

Avalia a similaridade percetual entre duas imagens, recorrendo a redes neuronais treinadas para identificar diferenças visuais percebidas pelo olho humano. É especialmente relevante para avaliar transformações meteorológicas, nas quais alterações de iluminação, cor ou textura podem afetar significativamente a perceção visual sem modificar a estrutura da cena. Os valores mais reduzidos indicam uma maior similaridade percetual.

Adicionalmente, ponderou-se a aplicação das métricas *Fréchet Inception Distance* (FID) [28] e *Kernel Inception Distance* (KID) [33] com vista a uma avaliação quantitativa da proximidade das imagens geradas em relação ao domínio real. Contudo, não foi viável dispor de um conjunto suficiente de imagens reais referentes ao domínio específico das condições adversas (chuva, neve, nevoeiro e noite) em estradas urbanas. A ausência destas imagens comprometeu a utilização destas métricas.

4.4.3.1 Implementação das Métricas Quantitativas

Para o cálculo das métricas de avaliação das imagens geradas, foi implementado um código em Python, capaz de processar os conjuntos de imagens e de calcular indicadores quantitativos. A seguir, proceder-se-á à descrição das principais etapas das medições.

PSNR e SSIM O procedimento consistiu na realização de uma comparação direta entre cada imagem gerada e a respetiva imagem original correspondente. As imagens foram normalizadas para valores na faixa [0,1], condição necessária para o cálculo correto das métricas. O PSNR foi calculado com o propósito de quantificar a relação entre o sinal original e o ruído introduzido, expressando a qualidade da imagem reconstruída em decibéis (dB), em que valores superiores indicam menor distorção. Já o SSIM avalia a similaridade percetual entre as imagens, fornecendo um valor entre 0 e 1, sendo 1 a correspondência perfeita. A implementação recorreu a funções da biblioteca scikit-image [83], reconhecida pela sua confiabilidade e eficiência na computação de métricas de qualidade de imagem.

LPIPS O procedimento consistiu no cálculo da distância LPIPS entre pares correspondentes de imagens originais e sintéticas. As imagens foram normalizadas e transformadas para um formato compatível com o modelo pré-treinado LPIPS, baseado na arquitetura VGG, utilizando a biblioteca PyTorch para uma manipulação eficiente e um uso acelerado por GPU. Para cada par de imagens, a distância percetual foi calculada sem

atualização de gradientes, e os valores obtidos foram acumulados para posterior análise estatística. A média e o desvio padrão do LPIPS foram calculados, proporcionando uma métrica quantitativa da qualidade percetual das imagens geradas em relação às originais.

Os valores obtidos através dos cálculos das métricas fornecem a base para a avaliação dos resultados, que será objeto de discussão no capítulo seguinte.

5. Análise e Avaliação dos Resultados Obtidos

No presente capítulo, os resultados obtidos são objeto de análise, de modo a avaliar o desempenho dos modelos utilizados, nomeadamente o CycleGAN e o Stable Diffusion (particularmente a versão Realistic Vision), seguindo a metodologia descrita no capítulo anterior. Inicialmente, é realizada uma análise qualitativa. Em seguida, é avaliada a segmentação semântica. Por fim, é feita uma análise quantitativa.

No Anexo A são apresentados exemplos adicionais das imagens geradas, com o objetivo de complementar e ilustrar os resultados obtidos.

5.1. Análise Qualitativa

A análise qualitativa visa avaliar a qualidade visual e a fidelidade estrutural das imagens geradas, considerando a presença dos elementos meteorológicos simulados e a preservação da geometria da imagem original. Nesta secção, a avaliação é realizada exclusivamente de forma visual, sem recurso a métricas automáticas.

5.1.1 Chuva

A Figura 5.1 expõe exemplos das imagens geradas pelos modelos para a condição de chuva, servindo como ilustração do desempenho geral obtido.

Através da visualização das imagens obtidas, observou-se que o modelo CycleGAN foi capaz de introduzir elementos visuais que sugerem a presença de chuva, como o escurecimento do céu e o efeito de chão molhado, de forma notavelmente realista, conforme evidenciado nas imagens (e) e (h). Contudo, o modelo demonstrou limitações significativas na preservação da estrutura da cena. Em diversas instâncias, como nas imagens (f) e (g), é possível identificar distorções significativas em árvores, veículos e outros elementos da paisagem, conferindo um aspeto excessivamente ruidoso.

O Stable Diffusion, por outro lado, demonstrou um desempenho superior na preservação da estrutura das imagens, mantendo os elementos urbanos com maior consistência. A aplicação dos efeitos de chuva foi realizada de forma mais uniforme, resultando em imagens visualmente mais estáveis e próximas da cena original. Contudo, em determinadas amostras, como em (k) e (l), o modelo adicionou focos de luz não presentes na cena original, introduziu uma claridade excessiva que não corresponde à iluminação típica de

condições chuvosas e, algumas imagens apresentaram um efeito de aprimoramento estético artificial, conferindo uma aparência menos realista à cena.



Figura 5.1: Exemplos de transformação de imagens para a condição de chuva. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion.

5.1.2 Neve

A Figura 5.2, exemplifica algumas das imagens geradas pelos modelos para a condição de neve.

O modelo CycleGAN demonstrou limitações significativas na geração de neve. Apesar da adição de padrões claros e texturas esbranquiçadas que remetem para esta condição climática, a aplicação foi excessiva e descontrolada, resultando em imagens altamente saturadas e com perda de informação estrutural, como nas imagens (f) e (h). Em diversas cenas, elementos como veículos e vegetação apresentaram uma perda significativa de detalhe, indicando uma incapacidade do modelo em aprender adequadamente a transformação necessária para manter a estrutura original.

O Stable Diffusion demonstrou uma superior eficácia na preservação da estrutura e na introdução de elementos característicos da neve. As imagens apresentam tonalidades frias e uma coerência visual que transmitem a sensação de inverno, mantendo os elementos principais visíveis. A imagem (j), por exemplo, ilustra bem o equilíbrio entre a aplicação de neve sobre as árvores e a preservação dos detalhes da cena. No entanto, quando comparadas com imagens de chuva, verifica-se que a geração de neve apresenta um nível de realismo ligeiramente inferior. O efeito de neve acumulada sobre alguns elementos, visível em várias cenas, cria um aspeto mais artificial, resultando em imagens com um estilo mais próximo de uma ilustração digital do que de uma fotografia real.



Figura 5.2: Exemplos de transformação de imagens para a condição de neve. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion.

5.1.3 Nevoeiro

A Figura 5.3 apresenta os resultados obtidos para a condição de nevoeiro.

As imagens geradas pelo CycleGAN, apresentam um efeito esbranquiçado que, apesar de transmitir de forma relativamente realista a sensação de nevoeiro, esse efeito foi pouco controlado, resultando na perda quase total de informação estrutural em várias imagens, como na imagem (g).

Em contraste, o Stable Diffusion demonstrou uma capacidade superior na preservação da estrutura das imagens. Contudo, a camada de nevoeiro manifesta-se de forma excessivamente homogénea, resultando numa aparência artificial e menos realista em comparação com o CycleGAN. Adicionalmente, em determinadas regiões, observa-se a introdução de iluminação e contrastes pouco realistas para este tipo de meteorologia, como na imagem (l).



Figura 5.3: Exemplos de transformação de imagens para a condição de nevoeiro. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion.

5.1.4 Noite

A Figura 4 apresenta exemplos de imagens produzidas pelos modelos para a condição de noite, ilustrando o desempenho alcançado.

O efeito noturno aplicado pelo CycleGAN mostrou-se globalmente realista, apresentando um maior grau de convicção em comparação com o produzido pelo Stable Diffusion. Contudo, a geometria da imagem apresentou distorções em áreas de alto contraste, resultando numa redução da naturalidade.

No que diz respeito ao Stable Diffusion, as imagens geradas apresentam uma iluminação introduzida naturalmente pelo modelo, através de postes de luz e faróis de veículos, que confere um aspeto mais convincente, como é notório nas imagens (j) e (k). O modelo demonstrou uma maior fidelidade na preservação dos elementos da cena, mantendo a sua reconhecibilidade. Contudo, algumas amostras evidenciam detalhes artificiais que comprometem o realismo pretendido. Por exemplo, na imagem (l), as nuvens que se encontravam presentes nas imagens originais, foram mantidas na mesma posição e formato. Estes detalhes devem-se à utilização de mapas de controlo fornecidos pelo ControlNet, que forçam o modelo a preservar os contornos e a geometria exata da cena.



Figura 5.4: Exemplos de transformação de imagens para a condição de noite. As imagens (a-d) correspondem às imagens originais. As imagens (e-h) foram geradas pelo modelo CycleGAN. As imagens (i-l) foram geradas pelo modelo Stable Diffusion.

5.1.5 Síntese da Análise Qualitativa

CycleGAN Os resultados obtidos pelo modelo CycleGAN, nomeadamente o realismo das características meteorológicas e a degradação estrutural, devem-se ao facto de o modelo ter sido treinado para aprender mapeamentos entre domínios sem pares de imagens correspondentes. Esta situação permite que o modelo consiga adicionar as características das condições desejadas. No entanto, o modelo não possui mecanismos explícitos de preservação estrutural, o que leva a distorções da estrutura.

Stable Diffusion Os resultados obtidos refletem as características intrínsecas do modelo, devido ao treino em grandes conjuntos de dados, o modelo consegue gerar imagens com boa qualidade visual, contudo tem a propensão de incluir efeitos estéticos artificiais, que comprometem o realismo. Em situações em que a imagem original apresenta alta luminosidade, o modelo demonstrou dificuldade em controlar essas regiões, resultando

num excesso de brilho que não corresponde às condições meteorológicas adversas. Em termos visuais, o nevoeiro e a neve foram as condições menos convincentes, apresentaram um aspeto artificial, semelhante a pinturas digitais. A utilização do ControlNet, garantiu uma maior fidelidade estrutural, mas deu origem a alguns artefactos, como texturas indesejadas ou a manutenção de nuvens em cenas noturnas. Embora fosse possível reduzir o controlo do ControlNet para evitar esses artefactos, tal ajuste comprometeria a preservação da estrutura original, tornando o controlo extra imprescindível.

Em suma, o CycleGAN apresenta uma melhor reprodução dos detalhes das condições meteorológicas, contudo, gera imagens com distorções estruturais. Já o Stable Diffusion preserva melhor a estrutura da cena e o realismo, embora represente os efeitos climáticos de forma menos detalhada.

5.2. Segmentação Semântica

A segmentação semântica foi utilizada para realizar uma avaliação visual das máscaras, com o objetivo de verificar se os principais objetos permanecem coerentemente identificados na imagem gerada.

O resultado da segmentação semântica das imagens geradas pelo Cyclegan é apresentado na Figura 5.5. A análise das imagens obtidas demonstrou que, apesar das condições meteorológicas variadas introduzirem alterações visuais, a estrutura semântica das cenas permanece, em grande parte, inalterada. Os elementos predominantes da cena, designadamente veículos, pavimentos e vegetação, apresentam cores consistentes com as respetivas categorias.

Contudo, algumas regiões apresentam discrepâncias. Em algumas imagens, observou-se que o ruído visual presente, que deformou a estrutura da cena, acabou por se refletir na segmentação semântica. Esta situação resultou em regiões classificadas com categorias incorretas, apresentando uma distribuição irregular.

Em determinadas áreas de transição entre classes, observa-se uma perda de nitidez, resultando em contornos menos precisos para veículos e vegetação, particularmente entre a vegetação e a estrada, como na imagem (j), na qual as regiões de vegetação próximas ao pavimento apresentam contornos ligeiramente irregulares; observa-se ainda que, em certos casos, a vegetação irregular identificada é inexistente na realidade. Adicionalmente, objetos de pequena escala, tais como placas, postes e sinais de trânsito, foram sujeitos a distorções ou até desapareceram.

Em condições de visibilidade reduzida, tais como nevoeiro ou noite, estas degradações intensificam-se. Em cenas noturnas, observa-se uma diminuição na definição das bordas entre veículos e estrada, indicando que a baixa iluminação compromete a precisão da segmentação, especialmente em regiões de sombra intensa ou com reflexos. De maneira semelhante, em determinadas circunstâncias de nevoeiro, parte da vegetação exibe perda de definição ou desaparecimento parcial, indicando que a interferência atmosférica compromete a deteção precisa das classes semânticas.

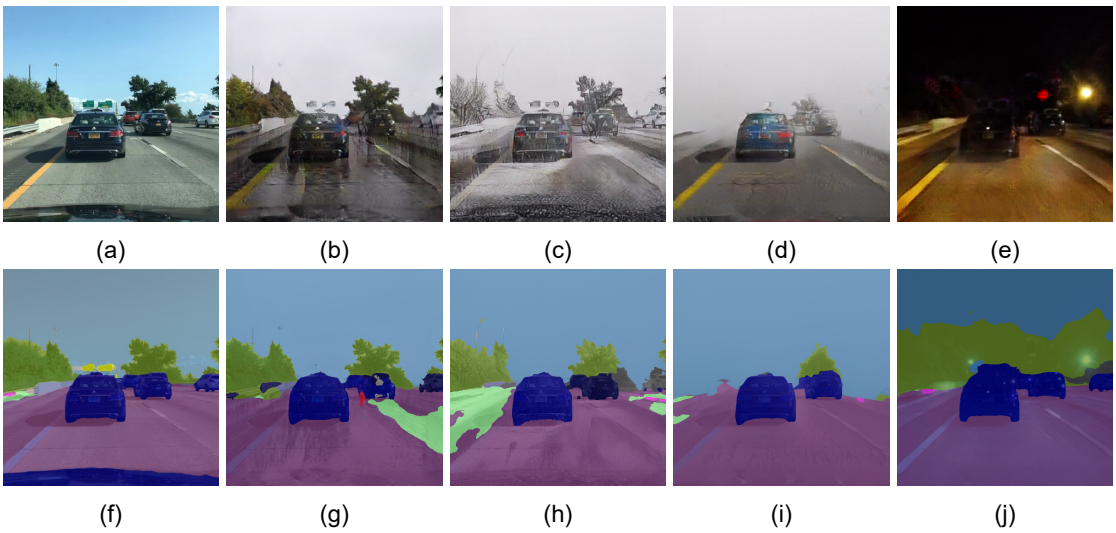


Figura 5.5: Exemplos de transformação de imagens pelo modelo Cyclegan e as respetivas segmentações. (a) Imagem original. (b–e) Imagens geradas pelo CycleGAN para chuva, neve, nevoeiro e noite, respetivamente. (f–j) Segmentações correspondentes a cada imagem acima.

Na segmentação semântica das imagens geradas pelo Stable Diffusion, apresentada na Figura 5.6 por alguns exemplos, observa-se uma maior preservação estrutural e semântica. As fronteiras entre classes, como estrada e passeio ou veículos, apresentam uma maior consistência, e a separação entre elementos da cena mantém-se mais fiel à referência original. As classes principais, nomeadamente estrada, céu, veículos e vegetação, mantêm-se claramente definidas mesmo em condições meteorológicas desfavoráveis.

Contudo, é importante salientar as limitações inerentes a este tipo de representação. Determinadas categorias exibem inconsistências subtis: na imagem de chuva (g), as regiões da estrada e do céu evidenciam uma pequena falha de segmentação, possivelmente atribuíveis aos artefatos introduzidos pela água. Na imagem noturna (j), observa-se que a transição entre estrada e vegetação foi menos precisa, indicando uma sensibilidade maior do modelo a condições de baixa luminosidade.

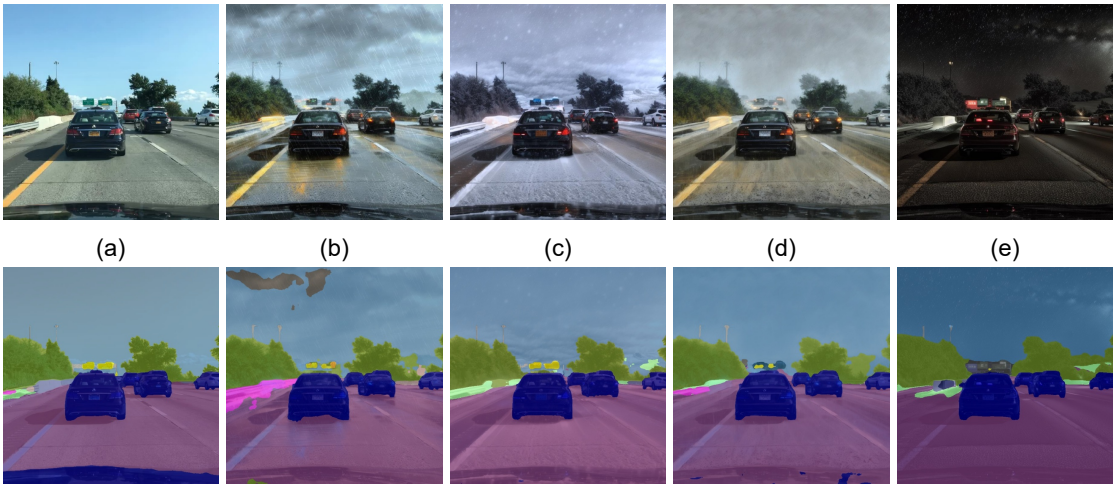


Figura 5.6: Exemplos de transformação de imagens pelo modelo Stable Diffusion e as respetivas segmentações. (a) Imagem original. (b–e) Imagens geradas pelo Stable Diffusion para chuva, neve, nevoeiro e noite, respetivamente. (f–j) Segmentações correspondentes a cada imagem acima.

Em geral, o Stable Diffusion revela uma maior robustez à degradação introduzida pelo processo de tradução de imagens, mantendo uma melhor definição e separação das classes. Em contraste, o CycleGAN, apesar de preservar grande parte da estrutura semântica, exibe uma maior propensão para a perda de detalhes finos, o que pode comprometer aplicações que dependem de segmentações precisas. O Anexo A contém mais exemplos das imagens geradas segmentadas.

5.3. Análise Quantitativa

A avaliação quantitativa teve como objetivo comparar o desempenho dos modelos de geração de imagens nas diferentes condições meteorológicas geradas. Esta análise vem complementar a avaliação qualitativa, fornecendo uma base objetiva para identificar os pontos fortes e as limitações de cada abordagem, bem como para fundamentar a comparação entre o Stable Diffusion e o CycleGAN.

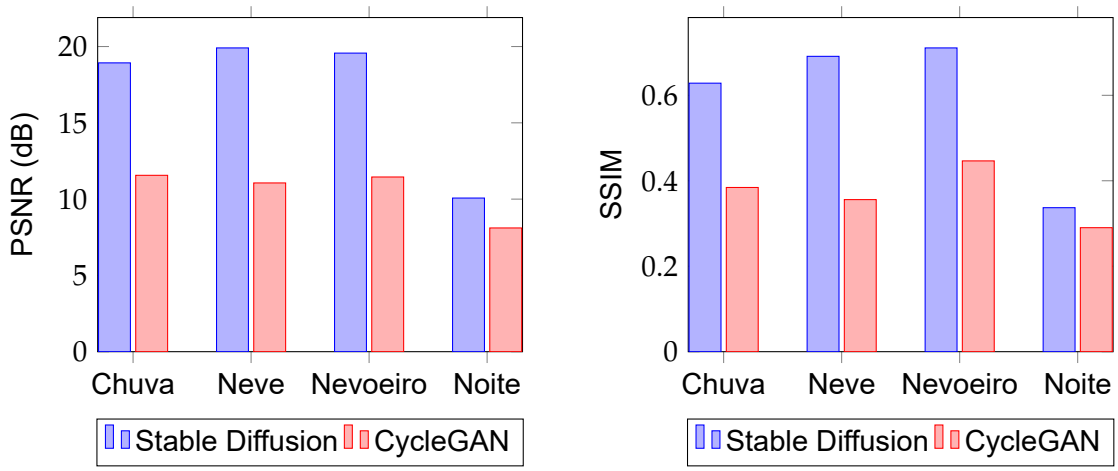
Os valores das métricas foram calculados de forma independente para cada condição climática. Esta metodologia possibilita a identificação de variações no desempenho dos modelos em diversos contextos visuais.

Os resultados obtidos são apresentados na Tabela 5.1.

Condição	Método	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Chuva	Stable Diffusion	18.93	0.6284	0.4431
	CycleGAN	11.56	0.3843	0.6616
Neve	Stable Diffusion	19.91	0.6912	0.4644
	CycleGAN	11.06	0.3559	0.6791
Nevoeiro	Stable Diffusion	19.57	0.7109	0.4016
	CycleGAN	11.45	0.4465	0.6843
Noite	Stable Diffusion	10.07	0.3370	0.5626
	CycleGAN	8.11	0.2902	0.7702

Tabela 5.1: Resultados quantitativos por condição climática.

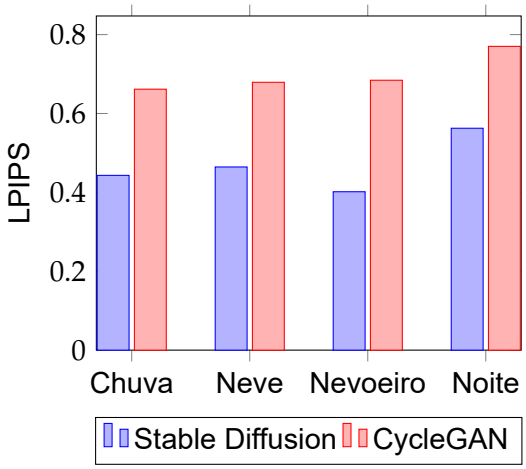
Para uma melhor compreensão dos resultados, as métricas foram também representadas por meio de gráficos, permitindo visualizar tendências e diferenças de forma mais clara e imediata.



(a) Comparação da métrica PSNR entre Stable Diffusion e CycleGAN.

(b) Comparação da métrica SSIM entre Stable Diffusion e CycleGAN.

Figura 5.7: Comparação das métricas PSNR e SSIM para diferentes condições meteorológicas.



(a) Comparação da métrica LPIPS entre Stable Diffusion e CycleGAN.

Figura 5.8: Comparação da métrica LPIPS para diferentes condições meteorológicas.

Relativamente ao PSNR e ao SSIM, o Stable Diffusion obteve valores cerca de 8 dB de PSNR superiores e quase o dobro de similaridade estrutural (SSIM) em comparação com o CycleGAN, conforme observado na Tabela 5.1 e nos Gráficos 5.7 e 5.8, o que sugere uma maior fidelidade aos detalhes da cena original e manutenção da disposição espacial dos objetos. Esta tendência foi consistente em todas as condições meteorológicas, evidenciando a robustez do modelo na preservação da geometria da cena.

A análise de LPIPS corrobora estas conclusões. Os valores mais reduzidos indicam uma proximidade percetual superior das imagens geradas em relação às imagens de referência. O Stable Diffusion exibiu um LPIPS entre 0,4016 (nevoeiro) e 0,5626 (noite), enquanto o CycleGAN oscilou entre 0,6616 (chuva) e 0,7702 (noite), corroborando a hipótese de que o Stable Diffusion gera imagens visualmente mais coerentes e com menor presença de artefatos.

A análise por condição climática demonstrou que o nevoeiro foi a situação mais bem reproduzida pelo Stable Diffusion, como é possivel observar nos Gráficos 5.7 e 5.8, apresentando a maior similaridade estrutural (SSIM = 0.7109), conforme indicado na Tabela 5.1, e o valor mais baixo de LPIPS (0.4016), revelando que a percepção visual da imagem gerada está mais próxima da referência. Este desempenho pode ser justificado pela natureza do fenómeno atmosférico, o qual se manifesta sobretudo por meio de alterações de contraste e dispersão de luz, exigindo menos modificações.

Relativamente à chuva, esta foi a condição que o CycleGAN reproduziu com maior precisão, apresentando valores de PSNR (11,56 dB) e SSIM (0,3843) ligeiramente superiores

aos dos outros cenários. Adicionalmente, obteve o valor de LPIPS (0,6616) mais baixo, indicando uma menor discrepância percetual em relação à imagem de referência. No entanto, o desempenho global do CycleGAN foi inferior ao do Stable Diffusion, que apresentou valores satisfatórios.

No caso da neve, o Stable Diffusion atingiu o valor mais elevado de PSNR (19,91 dB), o que indica que a imagem gerada não se desviou muito da imagem de referência.

Por fim, a condição de noite apresentou os piores resultados para ambos os modelos. O Stable Diffusion demonstrou um desempenho superior ao CycleGAN, embora tenha apresentado uma redução significativa na PSNR (10,07 dB) e na SSIM (0,3370). Tal comportamento era esperado, dado que a conversão para um cenário noturno exige modificações globais de iluminação e alterações de tonalidade, representando uma tarefa mais complexa para ambos os modelos.

5.3.1 Síntese da Análise Quantitativa

Em termos gerais, os valores numéricos obtidos corroboram o comportamento observado dos modelos.

No caso do Stable Diffusion, os valores de PSNR e SSIM indicam que o modelo é capaz de preservar a geometria da cena durante a tradução climática.

Relativamente às métricas percetuais, para condições mais extremas, como a presença de neve e a ocorrência de noite, os valores de LPIPS elevados refletem as alterações drásticas de iluminação.

Em suma, os resultados demonstram que o Stable Diffusion superou o CycleGAN em todas as métricas avaliadas, revelando-se mais robusto e generalizável para a tradução de domínios meteorológicos. Destaca-se particularmente em condições de nevoeiro e chuva, enquanto a neve e a noite se apresentam como cenários mais desafiadores. Estas conclusões são corroboradas tanto pelas métricas objetivas como pela análise visual das imagens geradas.

6. Conclusão

O presente trabalho demonstrou a relevância da geração de imagens sintéticas para mitigar o desequilíbrio meteorológico em conjuntos de dados de condução autónoma. Através da comparação entre duas abordagens representativas, o CycleGAN e o Stable Diffusion, especificamente a versão Realistic Vision, com ControlNet.

Os resultados obtidos evidenciaram que o CycleGAN é eficaz na aplicação de estilos meteorológicos, contudo, apresenta limitações na preservação da estrutura da cena. Por sua vez, o Stable Diffusion, em associação com o ControlNet, demonstrou uma superioridade na produção de imagens realistas e estruturalmente coerentes. A avaliação qualitativa, quantitativa e de segmentação semântica confirmou a superioridade dos modelos de difusão neste contexto.

Apesar de o Stable Diffusion demonstrar superioridade em relação ao CycleGAN, existiram diferenças metodológicas que influenciaram diretamente os resultados. Como, a utilização do ControlNet que garantiu uma maior fidelidade estrutural e a aplicação de funções de pós-processamento que ”editaram”as imagens, por exemplo, escurecendo o céu no caso de chuva ou adicionando um tom frio no caso de neve. Estas funções ajudaram o modelo a caracterizar melhor as condições meteorológicas, mas foram necessárias devido à restrição extrema de manter a estrutura original da imagem.

A contribuição mais significativa deste trabalho reside na demonstração de que a geração sintética de imagens, por meio de modelos generativos modernos, destacando-se o Stable Diffusion, tem o potencial de mitigar o desequilíbrio nos conjuntos de dados de condução autónoma.

Limitações Apesar dos resultados promissores do Stable Diffusion, este estudo apresenta várias limitações. Em primeiro lugar, o elevado custo computacional que limita a sua escalabilidade em contextos de grande dimensão. Além disso, foram observados alguns efeitos estéticos indesejados, como reflexos excessivos no asfalto ou efeitos artificiais, resultantes do facto de o modelo base ter sido treinado em múltiplos domínios visuais e não especificamente para cenários rodoviários. Acresce ainda a dificuldade em equilibrar a preservação estrutural da cena com a intensidade do efeito atmosférico:

quando o modelo tem mais liberdade, insere de forma expressiva as condições meteorológicas, mas altera a estrutura original; quando é rigidamente controlado para manter a estrutura, perde criatividade e reduz a intensidade desses efeitos.

Trabalho Futuro No âmbito dos trabalhos futuros, propõe-se a afinação (*fine-tuning*) de modelos de difusão especificamente para o domínio rodoviário, de modo a reduzir artefactos estéticos e a melhorar a coerência semântica das transformações meteorológicas. Além disso, seria pertinente avaliar diretamente o impacto das imagens sintéticas no desempenho dos sistemas de perceção, integrando-as no treino de segmentadores ou detetores e comparando os resultados com os obtidos por modelos treinados apenas com dados reais. Por fim, é necessário estender esta abordagem às sequências de vídeo, garantindo a consistência temporal dos fenómenos adversos, o que representa um passo crucial para as aplicações práticas da condução autónoma.

Cordts

Bibliografia

[1] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. [Cited on pages 1, 17, and 19.]

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [Cited on pages 1, 17, 18, and 19.]

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 674–10 685. [Cited on pages 1, 13, 14, and 22.]

[4] L. Zhang, A. Rao, and M. Agrawala, " Adding Conditional Control to Text-to-Image Diffusion Models ," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2023, pp. 3813–3824. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00355 [Cited on pages 1, 15, 25, and 26.]

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251. [Cited on pages 1, 11, 13, and 22.]

[6] S.-C. Wang and S.-C. Wang, "Artificial neural network," *Interdisciplinary computing in java programming*, pp. 81–100, 2003. [Cited on page 3.]

[7] "Artificial Neural Networks and its Applications - GeeksforGeeks — geeksforgeeks.org," https://www.geeksforgeeks.org/artificial-intelligence/artificial-neural-networks-and-its-applications/, [Accessed 02-09-2025]. [Cited on pages vii and 4.]

[8] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2019. [Online]. Available: https://arxiv.org/abs/1803.08375 [Cited on page 4.]

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986. [Online]. Available: https://api.semanticscholar.org/CorpusID:205001834 [Cited on page 4.]

[10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106 [Cited on page 5.]

[11] K. O’Shea, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015. [Cited on page 6.]

[12] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2528–2535, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:9893011 [Cited on page 7.]

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. [Cited on pages 8 and 14.]

[14] T. S. Silva, “A Short Introduction to Generative Adversarial Networks - Thalles' blog — sthalles.github.io,” https://sthalles.github.io/intro-to-gans/, [Accessed 02-09-2025]. [Cited on pages vii and 8.]

[15] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., 2020. [Cited on page 9.]

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223. [Cited on page 11.]

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976. [Cited on page 11.]

[18] M.-Y. Liu and O. Tuzel, “Coupled generative adversarial networks,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser.

NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 469–477. [Cited on page 12.]

[19] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2242–2251. [Cited on page 12.]

[20] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2017. [Online]. Available: https://arxiv.org/abs/1605.09782 [Cited on page 12.]

[21] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," 2017. [Online]. Available: https://arxiv.org/abs/1606.00704 [Cited on page 12.]

[22] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8185–8194. [Cited on page 12.]

[23] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018. [Cited on pages 12 and 13.]

[24] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2018, p. 36–52. [Online]. Available: https://doi.org/10.1007/978-3-030-01246-5_3 [Cited on page 12.]

[25] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1429–1437. [Cited on page 12.]

[26] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797. [Cited on page 12.]

[27] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *International Conference on Learning Representations*, 2018. [Cited on pages 12 and 14.]

[28] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf [Cited on pages 12, 13, and 43.]

[29] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595. [Cited on pages 12 and 42.]

[30] X. Li, K. chang Kou, and B. Zhao, “Weather gan: Multi-domain weather translation using generative adversarial networks,” *ArXiv*, vol. abs/2103.05422, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232168941 [Cited on page 12.]

[31] B. Zhao, L. Hua, X. Li, X. Lu, and Z. Wang, “Weather recognition via classification labels and weather-cue maps,” *Pattern Recognition*, vol. 95, pp. 272–284, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320319302481 [Cited on page 13.]

[32] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf [Cited on page 13.]

[33] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, “Demystifying MMD GANs,” in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1lUOzWCW [Cited on pages 13 and 43.]

[34] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:216078090 [Cited on page 13.]

[35] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation”, booktitle=”medical image computing and computer-assisted intervention – miccai 2015,” N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241. [Cited on pages 14 and 15.]

[36] T. Brooks, A. Holynski, and A. A. Efros, “Instructpix2pix: Learning to follow image editing instructions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 18 392–18 402. [Cited on page 14.]

[37] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” ser. NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., 2020. [Cited on page 14.]

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231591445 [Cited on page 15.]

[39] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, “Prompt-to-prompt image editing with cross attention control,” 2022. [Cited on page 15.]

[40] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” 2022. [Online]. Available: https://arxiv.org/abs/2207.12598 [Cited on page 16.]

[41] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986. [Cited on page 16.]

[42] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,”

IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 3, pp. 1623–1637, 2022. [Cited on page 16.]

[43] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, and G. Shakhnarovich, "DIODE: A Dense Indoor and Outdoor DEpth Dataset," *CoRR*, vol. abs/1908.00463, 2019. [Online]. Available: http://arxiv.org/abs/1908.00463 [Cited on page 16.]

[44] G. Gu, B. Ko, S. Go, S.-H. Lee, J. Lee, and M. Shin, "Towards light-weight and real-time line segment detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 726–734, 06 2022. [Cited on page 16.]

[45] S. Xie and Z. Tu, "Holistically-nested edge detection," *Int. J. Comput. Vision*, vol. 125, no. 1–3, p. 3–18, Dec. 2017. [Cited on page 16.]

[46] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [Cited on page 16.]

[47] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021. [Cited on page 16.]

[48] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 2989–2998. [Cited on page 16.]

[49] C. Qian, Y. Guo, Y. Mo, and W. Li, "Weatherdg: Llm-assisted procedural weather generation for domain-generalized semantic segmentation," *IEEE Robotics and Automation Letters*, vol. 10, no. 6, pp. 5919–5926, 2025. [Cited on page 16.]

[50] L. Hoyer, D. Dai, and L. Van Gool, "Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 9924–9935. [Cited on page 16.]

[51] H. Lukas, D. Dengxin, and V. G. Luc, “Domain adaptive and generalizable network architectures and training strategies for semantic image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 1, pp. 220–235, 2024. [Cited on page 16.]

[52] C. Sakaridis, D. Dai, and L. Van Gool, “ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. [Cited on pages 17 and 20.]

[53] S. Christos, D. Dengxin, and V. G. Luc, “Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2019. [Cited on page 17.]

[54] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, e. V. Adam, Hartwig”, M. Hebert, C. Sminchisescu, and Y. Weiss, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018, pp. 833–851. [Cited on page 17.]

[55] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [Cited on page 17.]

[56] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, 06 2010. [Cited on page 17.]

[57] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, “Deep high-resolution representation learning for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2021. [Cited on page 18.]

[58] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, “The role of context for object detection and semantic segmentation in the wild,” 06 2013. [Cited on page 18.]

[59] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin, “ Look into Person: Self-Supervised Structure-Sensitive Learning and a New Benchmark for Human Parsing ,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Los Alamitos, CA, USA: IEEE Computer Society, Jul 2017, pp. 6757–6765. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.715 [Cited on page 18.]

[60] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," 07 2018. [Cited on page 18.]

[61] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017. [Online]. Available: https://arxiv.org/abs/1706.05587 [Cited on page 18.]

[62] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [Cited on page 18.]

[63] M. KENK, "Dawn," 2020. [Online]. Available: https://data.mendeley.com/datasets/766ygrbt8y/3 [Cited on page 20.]

[64] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, pp. 1231 – 1237, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:9455111 [Cited on page 20.]

[65] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft coco," 2025. [Online]. Available: https://dx.doi.org/10.21227/x1ng-4d22 [Cited on page 20.]

[66] "An Introduction to Diffusion Models and Stable Diffusion - Marvik — blog.marvik.ai," https://blog.marvik.ai/2023/11/28/an-introduction-to-diffusion-models-and-stable-diffusion/, [Accessed 05-08-2025]. [Cited on pages vii and 24.]

[67] Evgeny, "SG161222/Realistic_Vision_V5.1_noVAE · Hugging Face — huggingface.co," https://huggingface.co/SG161222/Realistic_Vision_V5.1_noVAE, [Accessed 27-08-2025]. [Cited on page 25.]

[68] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "Stable diffusion v1.5," https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5, 2022, [Accessed 27-08-2025]. [Cited on page 25.]

[69] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image

synthesis," 2023. [Online]. Available: https://arxiv.org/abs/2307.01952 [Cited on page 25.]

[70] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," 2020. [Online]. Available: https://arxiv.org/abs/1910.03771 [Cited on pages 25 and 28.]

[71] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986. [Cited on pages 27 and 28.]

[72] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf, "Diffusers: State-of-the-art diffusion models," https://github.com/huggingface/diffusers, 2022. [Cited on page 28.]

[73] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *European Conference on Computer Vision*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:50781105 [Cited on page 28.]

[74] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. [Online]. Available: https://arxiv.org/abs/1912.01703 [Cited on page 30.]

[75] "GitHub - AUTOMATIC1111/stable-diffusion-webui: Stable Diffusion web UI — github.com," https://github.com/AUTOMATIC1111/stable-diffusion-webui, [Accessed 12-09-2025]. [Cited on page 34.]

[76] H. Bansal, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks — junyanz.github.io," https://junyanz.github.io/CycleGAN/, [Accessed 24-09-2025]. [Cited on pages vii and 37.]

[77] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020. [Cited on pages 41 and 42.]

[78] MMCV Contributors, “MMCV: OpenMMLab computer vision foundation,” https://github.com/open-mmlab/mmcv, 2018. [Cited on page 42.]

[79] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with numpy,” *Nature*, vol. 585, no. 7825, p. 357–362, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1038/s41586-020-2649-2 [Cited on page 42.]

[80] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000. [Cited on page 42.]

[81] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. [Cited on page 42.]

[82] W. M. Floyd, “A unified theory of signal-to-noise ratio in linear modulation systems,” *Bell System Technical Journal*, vol. 35, no. 6, 1956. [Cited on page 42.]

[83] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, “scikit-image: image processing in python,” *PeerJ*, vol. 2, p. e453, Jun. 2014. [Online]. Available: http://dx.doi.org/10.7717/peerj.453 [Cited on page 43.]

Anexos

A.Anexo A: Exemplos adicionais das imagens geradas

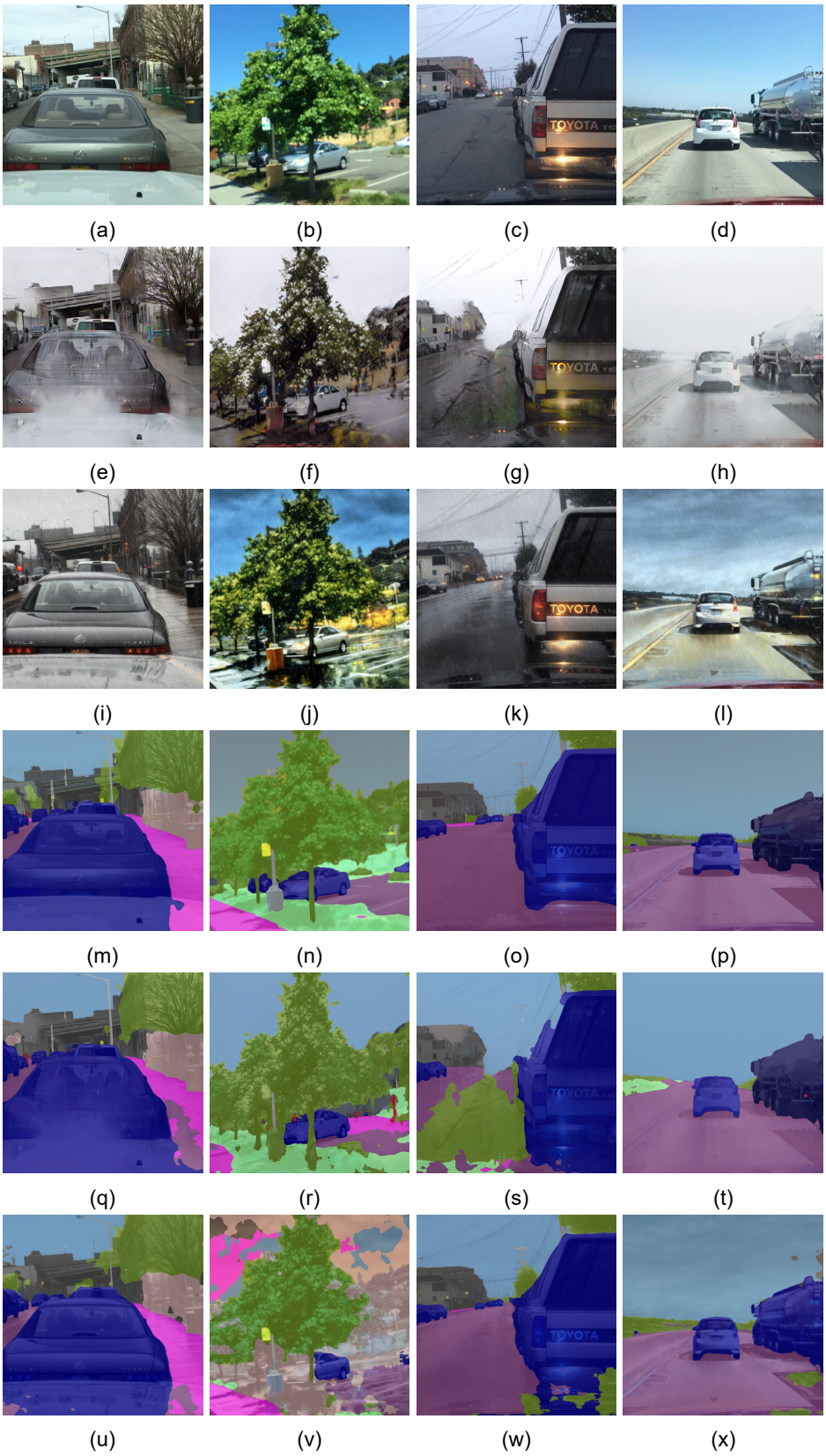


Figura 1: Exemplos de imagens para a condição de chuva. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas.

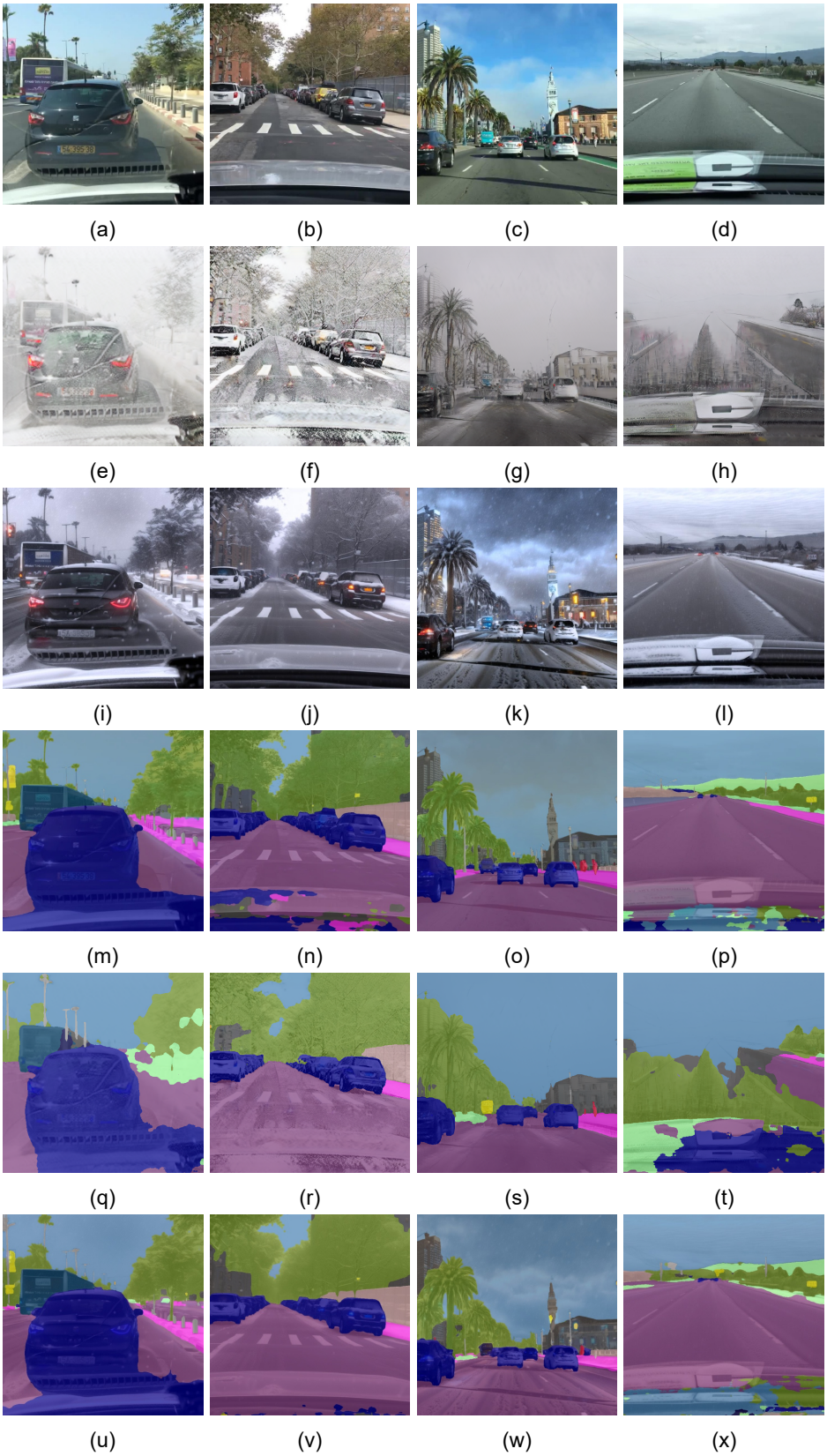


Figura 2: Exemplos de imagens para a condição de neve. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas.

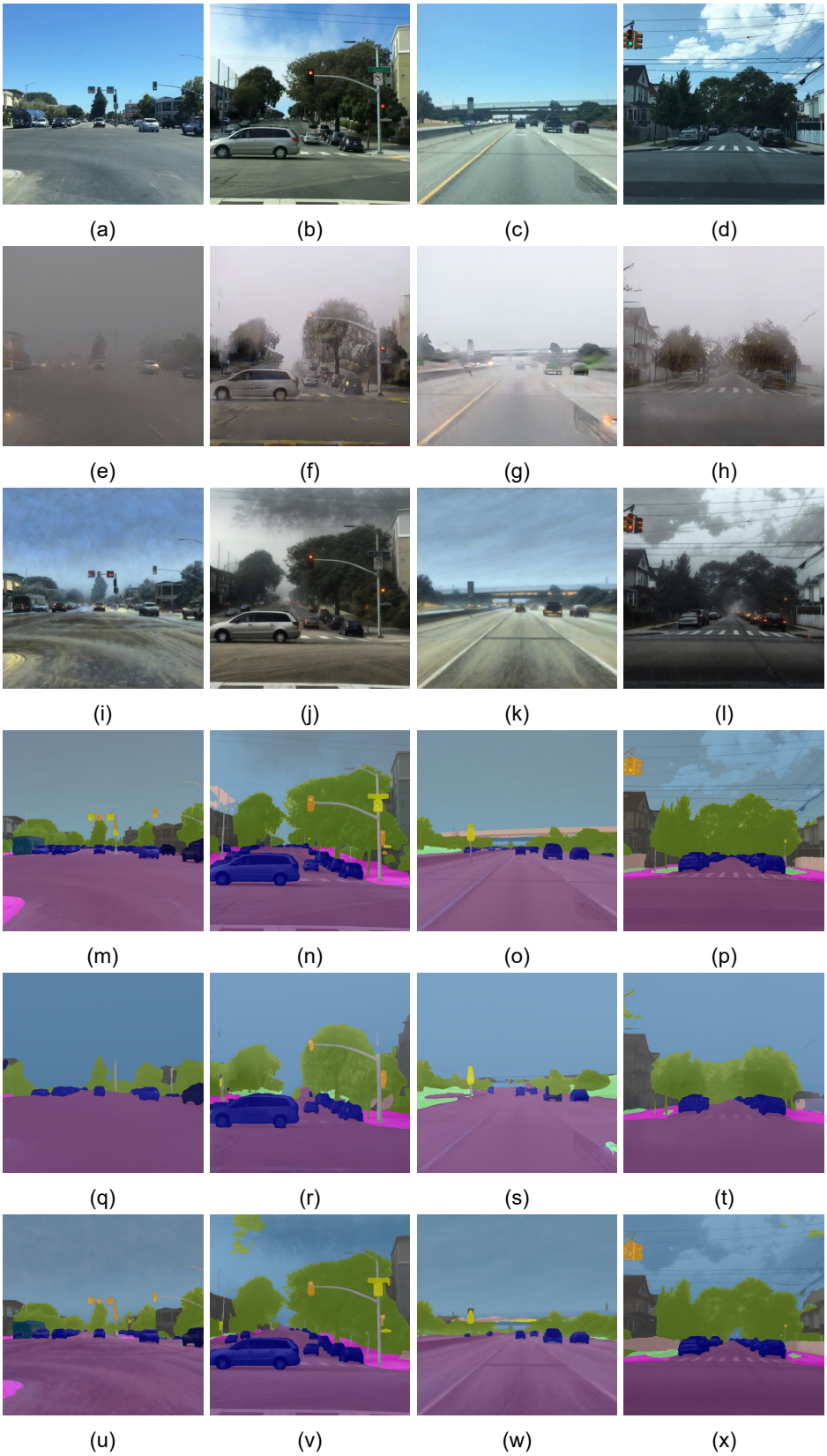


Figura 3: Exemplos de imagens para a condição de nevoeiro. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas.

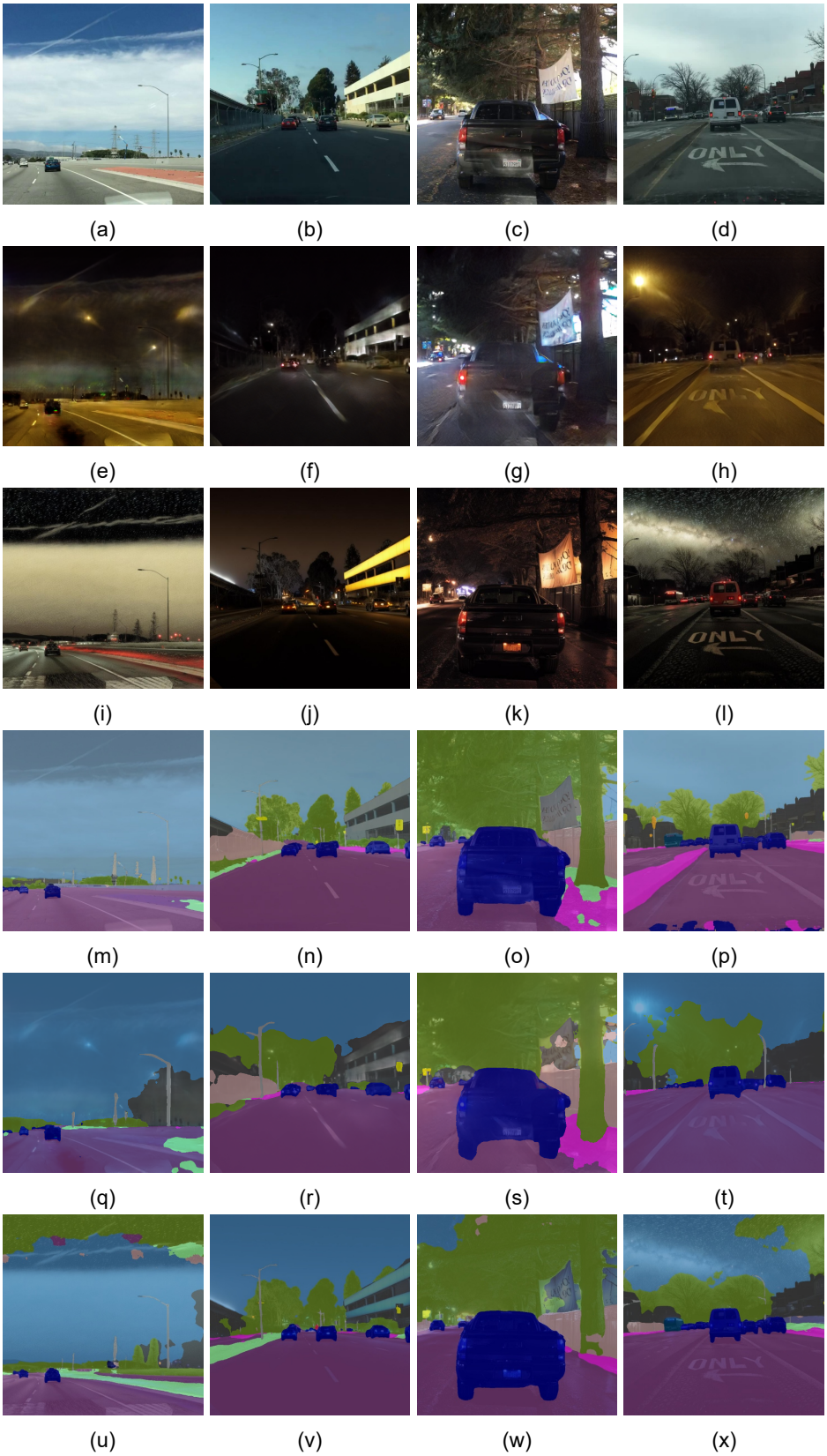


Figura 4: Exemplos de imagens para a condição de noite. (a–d) imagens originais; (e–h) imagens geradas pelo CycleGAN; (i–l) imagens geradas pelo Stable Diffusion; (m–p) imagens originais segmentadas; (q–t) imagens geradas pelo CycleGAN segmentadas; (u–x) imagens geradas pelo Stable Diffusion segmentadas.

Todo o material complementar do trabalho prático, incluindo imagens geradas e códigos implementados, pode ser consultado em:

https://drive.google.com/drive/folders/1zSSmP2D_Nrxmzx6dJV78nzZ7iuDAHCmD?usp=drive_link