Pontifícia Universidade Católica do Paraná
Programa de Pós-Graduação em Informática

Uma Abordagem de Redes Generativas Adversárias para Tratamento de Mudanças de Conceito em Classificação de Vagas de Estacionamento

Antonio Michel Ferreira dos Santos

Orientador

André Gustavo Hochuli

Co-orientador

Jean Paul Barddal

Curitiba
2024

Pontifícia Universidade Católica do Paraná
Programa de Pós-Graduação em Informática

Uma Abordagem de Redes Generativas Adversárias para Tratamento de Mudanças de Conceito em Classificação de Vagas de Estacionamento

Antonio Michel Ferreira dos Santos

Dissertação apresentada ao Programa de Pós-Graduação em Informática como requisito parcial para obtenção do título de Mestre em Informática.

Campo de Concentração: Ciência da Computação

Orientador: André Gustavo Hochuli

Co-orientador: Jean Paul Barddal

Curitiba
2024

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Edilene de Oliveira dos Santos CRB 9 / 1636

S237a
2024

Santos, Antonio Michel Ferreira dos
Uma abordagem de redes generativas adversárias para tratamento de mudanças de conceito em classificação de vagas de estacionamento / Antonio Michel Ferreira dos Santos ; orientador: André Gustavo Hochuli ; coorientador: Jean Paul Barddal. -- 2024
79 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2024
Bibliografia: f.76-79

1. Informática. 2. Processamento de imagem auxiliado por computador. 3. Inteligência artificial. 4. Aprendizado profundo. 5. Redes neurais (Computação). 6. Fluxo de dados (Computadores). I. Hochuli, André Gustavo. II. Barddal, Jean Paul. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título

CDD 20. ed. – 004

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

Curitiba, 07 de março de 2025.

36-2025

DECLARAÇÃO

Declaro para os devidos fins, que **ANTONIO MICHEL FERREIRA DOS SANTOS** defendeu a dissertação de Mestrado intitulada "**Uma Abordagem de Redes Generativas Adversárias para Tratamento de Mudanças de Conceito em Classificação de Vagas de Estacionamento"**, na área de concentração Ciência da Computação no dia 05 de dezembro de 2024, no qual foi aprovado.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

gov.br Documento assinado digitalmente
EMERSON CABRERA PARAISO
Data: 21/03/2025 15:14:58-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática

Dedico este trabalho à todos que de alguma maneira me acompanharam no processo de produção do mesmo, apoiando-me de alguma forma durante este tempo.
Em especial, dedico aos meus pais, pois sem eles eu nada seria. Além disso aos meus padrinhos Alcindo e Leonilda, os quais me incentivaram desde pequeno à ter minha vida voltada aos estudos e sempre acreditaram em mim.
Por fim, mas não menos especial, à minha finada avó Terezinha. Seu incentivo e alegria constantes nunca hão de ser esquecidos. Espero que, de onde quer que você esteja, sinta-se orgulhosa de todas as minhas conquistas.

A persistência é o menor caminho do êxito - Charles Chaplin

Resumo

Com a constante evolução na captura de dados em diversas áreas, o interesse em métodos de mineração de fluxos contínuos de dados têm aumentado. Devido à demanda de processamento dos dados em modo *on-line*, e também à volatilidade dos mesmos, o principal problema em tal área são as mudanças de conceito. Este fenômeno ocorre quando há uma alteração significativa na distribuição dos dados em uma *stream*, acarretando em perda de performance de modelos de aprendizado de máquina. Comumente tal problema é abordado por técnicas de janelamento de dados para a atualização do modelo, porém há ocasiões em que há pouca disponibilidade de dados rotulados para isso. Uma abordagem para mitigação desse tipo de problema é a aplicação de técnicas de aumento de dados. Redes DCGAN são conhecidas por suas aplicações em geração de dados sintéticos, porém ainda não são amplamente exploradas como técnica de aumento de dados em cenários de fluxos de dados. O uso de DCGANs para essa finalidade pode prover maior volume e variabilidade de dados rotulados para a adaptação do modelo de aprendizado de máquina à ocorrência de mudanças de conceito. A avaliação do impacto dessa abordagem é conduzida utilizando as bases de imagens PKLot e CNRPark-EXT, simulando um cenário de fluxo de imagens. A aplicação de DCGANs resulta em melhorias relevantes em CNNs com menor número de parâmetros, reduzindo a necessidade de atualização dos modelos ao longo da *stream* devido à maior generalização adquirida pelo modelo quando comparados à cenários onde a atualização do classificador é realizada apenas com dados reais.

Palavras-chave: Fluxos Contínuos de Dados, Mudanças de Conceito, Aprendizado Profundo

Abstract

With the constant evolution of data capture in a wide variety of areas, interest in methods for mining continuous data streams has increased. Due to the demand to process data in online mode, as well as its volatility, the main problem in this area is the concept drifts. This phenomenon occurs when there is a significant change in the data distribution in a stream, resulting in a performance loss of machine learning models. This problem is usually addressed by data windowing techniques for updating the model, but there are occasions when labeled data is not readily available for this purpose. One approach to mitigate this type of problem is the application of data augmentation techniques. GANs are known for their synthetic data generation applications but are not widely exploited as a data augmentation technique in a data stream scenario. Using GANs for this purpose can provide greater volume and variability of labeled data for adapting the machine learning model to the occurrence of concept changes. The assessment of this approach's impact is conducted using the PKLot and CNRPark-EXT datasets, simulating an image data stream. The application of DCGANs ends up resulting in significant improvements in CNNs with a smaller number of parameters, reducing the need to update the models throughout the stream due to the greater generalization acquired by the model when compared to scenarios where the classifier is updated only with real data.

Key-words: Data Streams, Concept Drifts, Deep Learning

Agradecimentos

Primeiramente, aos meus orientadores Prof. André Gustavo Hochuli e Prof. Jean Paul Barddal, os quais me apresentaram a pesquisa. Ambos promoveram diversas discussões e ideias, comentários e críticas, sempre que necessário. Além de suas carreiras acadêmicas relevantes, cabe aqui ressaltar diversas qualidades como seriedade, objetividade, presteza e disponibilidade, essenciais para a evolução deste trabalho e diversos outros. Espero herdar tantas qualidades para a continuidade de minha futura carreira. Agradeço profundamente a confiança depositada em mim para este presente trabalho.

Aos demais professores do Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná que de alguma forma contribuíram para o estudo de alguma forma, seja via sugestões diretas ou mesmo à ministração de disciplinas do curso de mestrado. Todos foram essenciais para o seguimento da pesquisa.

Agradeço ao meu irmão, amigos e colegas de trabalho, os quais sempre me acompanharam nessa jornada e prestaram auxílio em diversos âmbitos no tempo decorrido do curso.

A Fundação Araucária, pelo apoio financeiro cedido ao projeto.

Finalmente, e em especial, aos meus pais. Nenhuma conquista em minha vida seria possível sem eles, sem sua dedicação, esforços e paciência. Em vocês me espelho todos os dias, buscando a excelência cotidianamente.

Sumário

Lista de Figuras **viii**

Lista de Tabelas **x**

Lista de Acrônimos **xi**

1 Introdução **1**
1.1 Motivação e Hipótese 2
1.2 Objetivos 3
1.3 Organização 4

2 Mineração de Fluxos Contínuos de Dados **5**
2.1 Mudanças de Conceito 6
2.1.1 Mudança de Contexto Real 8
2.1.2 Mudança de Contexto Virtual 10
2.1.3 Mudança de Contexto Abrupta 11
2.1.4 Mudança de Contexto Gradual 11
2.1.5 Mudança de Contexto Incremental 12
2.1.6 Mudança de Contexto Recorrente 12
2.1.7 Mudança de Contexto por *Blips* 13
2.2 Detectores de Mudanças de Conceito 14
2.2.1 ADWIN 15

3 Aprendizado Profundo **17**
3.1 Redes Neurais Convolucionais 17
3.1.1 Operação de Convolução 19
3.1.2 Operação de *Pooling* 21
3.1.3 Camadas Totalmente Conectadas 21
3.1.4 Aplicações 23

3.2 Redes Generativas Adversárias 25
3.2.1 Estrutura e Funcionamento de uma GAN 26
3.2.2 *Deep Convolutional Generative Adversarial Network* 27

4 Bases de Dados **29**
4.1 *PKLot* 29
4.1.1 PUC 32
4.1.2 UFPR04 34
4.1.3 UFPR05 35
4.2 *CNRPark-Ext* 36

5 Estado da arte **39**

6 Metodologia **46**
6.1 Ambiente de Execução 47
6.2 Fluxo de Imagens 47
6.3 Pré-processamento das imagens 49
6.4 Arquitetura das CNNs 49
6.5 Arquitetura das DCGANs 50
6.6 Estratégia de Treinamento 51
6.7 Métodos de Detecção de Mudanças de Conceito 52
6.8 Cenários Avaliados 54
6.9 Métodos de Avaliação 55

7 Resultados **56**
7.1 Protocolo Experimental 56
7.2 Resultados Obtidos 58
7.2.1 Fluxo de Imagens CNRPark-Ext → PKLot 58
7.2.2 Fluxo de Imagens PKLot → CNRPark-Ext 63
7.2.3 Avaliação de Mudanças de Conceito 68

8 Conclusões **74**

Referências **76**

Lista de Figuras

2.1 Ilustração de Mudança de Conceito Real 9
2.2 Ilustração de Mudança de Conceito Virtual 10
2.3 Ilustração de Mudança de Conceito Abrupto 11
2.4 Ilustração de Mudança de Conceito Gradual 12
2.5 Ilustração de Mudança de Conceito Incremental 12
2.6 Ilustração de Mudança de Conceito Recorrente 13
2.7 Ilustração de Mudança de Conceito por *Blips* 13

3.1 Estrutura de uma rede CNN.[1] 18
3.2 Efeito de borda em uma imagem 5x5 com uma janela 3x3 - Fonte: (CHOLLET, 2017) 20
3.3 Reorganização dos dados em uma Camada Totalmente Conectada. 22
3.4 Estrutura completa de uma Rede Neural Convolucional. Fonte: NVidia 22
3.5 Exemplos de imagens e suas classes pertencentes à *ImageNet* - Fonte: (DENG et al., 2009) 24
3.6 Estrutura de uma Rede GAN. Fonte: (SILVA, 2017) 26
3.7 Estrutura do gerador de uma DCGAN. Fonte: (RADFORD; METZ; CHINTALA, 2016) 27
3.8 Comparação entre imagens reais, geradas por uma GAN e uma DCGAN. Fonte: (RADFORD; METZ; CHINTALA, 2016) 28

4.1 Exemplos de vagas segmentadas da base PKLot 30
4.2 Amostras da PKLot, seus estacionamentos e climas captados . . . 30
4.3 Exemplo de imagens dos subconjuntos da base *PKLot* ordenados temporalmente. 31
4.4 Distribuição das vagas de acordo com o clima - PUC. 33
4.5 Distribuição das vagas de acordo com o clima - UFPR04. 34

4.6 Distribuição das vagas de acordo com o clima - UFPR05. 35
4.7 Amostras da *CNRPark-Ext* 36
4.8 Amostras da *CNRPark-Ext*, suas câmeras e diferentes climas capturados. 37

5.1 Resultados em forma gráfica da IGI para os *datasets* testados. Fonte: (GUZY; WOźNIAK; KRAWCZYK, 2021). 40
5.2 R^2D^2 gerada utilizando a base *CelebA*. Fonte: (GUZY; WOźNIAK; KRAWCZYK, 2021). 41
5.3 Visão geral da arquitetura do modelo proposto. Fonte: (MALIALIS et al., 2022) 42
5.4 Modelo proposto por POUYANFAR et al. (2018) 44

6.1 Fluxos de imagens utilizados durante os experimentos. 47
6.2 Exemplos de mudanças de conceito por ângulo/câmera presentes nos fluxos de imagens utilizados. 48
6.3 Estrutura das redes convolucionais utilizadas nos experimentos. . 50
6.4 Estrutura da rede DCGAN utilizada nos experimentos. 51

7.1 Protocolo Experimental realizado. 57
7.2 Diagramas de diferenças obtidos em termos de acurácia dentre os diferentes cenários e redes para o fluxo de imagens CNRPark-Ext → PKLot. 60
7.3 Diagramas de diferenças obtidos em termos de detecção de mudanças de conceito dentre os diferentes cenários e redes para o fluxo de imagens CNRPark-Ext → PKLot. 62
7.4 Diagramas de diferenças obtidos em termos de detecção de acurácia dentre os diferentes cenários e redes para o fluxo de imagens PKLot → CNRPark-Ext. 64
7.5 Diagramas de diferenças obtidos em termos de detecção de mudanças de conceito dentre os diferentes cenários e redes para o fluxo de imagens PKLot → CNRPark-Ext. 67
7.6 Desempenho em termos de acurácia ao longo do fluxo de imagens CNRPark-EXT → PKLot. 68
7.7 Mudança de conceito relacionada a alteração da distribuição entre classes. 69
7.8 Ilustração da Mudança de Conceito encontrada na Figura 7.7. . . 69

7.9 Mudança de conceito relacionada a alterações climáticas. 70
7.10 Ilustração da Mudança de Conceito encontrada na Figura 7.9. . . 70
7.11 Desempenho em termos de acurácia ao longo do fluxo de imagens PKLot → CNRPark-EXT. 71
7.12 Mudança de conceito relacionada a alterações entre estacionamentos. 72
7.13 Ilustração da Mudança de Conceito encontrada na Figura 7.12. . . 72
7.14 Mudança de conceito relacionada a alterações entre bases de dados. 73
7.15 Ilustração da Mudança de Conceito encontrada na Figura 7.14. . . 73

Lista de Tabelas

2.1 Exemplo de Conceito. 7
2.2 Exemplo de possível modificação do conceito apresentado na Tabela 2.1. 8

4.1 Quantidade de imagens segmentadas coletadas e dias por estacionamento. 31
4.2 Distribuição entre as classes disponíveis na base PKLot. 32
4.3 Distribuição entre as classes disponíveis na base PKLot de acordo com o clima. 32
4.4 Distribuição de amostras entre as câmeras da base *CNRPark-Ext*. . 38
4.5 Distribuição entre as classes disponíveis na base *CNRPark-Ext* de acordo com o clima. 38

5.1 IGI médios e tempos de processamento para as redes DCGAN e WGAN. Fonte: (GUZY; WOźNIAK; KRAWCZYK, 2021). 40
5.2 Comparação dos trabalhos do estado da arte. 45

6.1 Resultados obtidos para seleção do detector de mudanças de conceito. 53

7.1 Acurácias obtidas (%) envolvendo o fluxo de imagens CNRPark-Ext → PKLot. 58
7.2 Valores-p relacionados às acurácias obtidas no cenário com o fluxo CNRPark-Ext → PKLot 59
7.3 Valores-p obtidos com o teste de Nemenyi em relação às acurácias obtidas para o fluxo de imagens CNRPark-Ext → PKLot. 60
7.4 Número de detecções de mudanças de conceito envolvendo o fluxo de imagens CNRPark-Ext → PKLot. 61

7.5 Valores-p relacionados ao número de detecções de mudanças de conceito no cenário com o fluxo CNRPark-Ext → PKLot 61
7.6 Valores-p obtidos com o teste de Nemenyi em relação ao número de mudanças de conceito detectadas para o fluxo de imagens CNRPark-Ext → PKLot. 62
7.7 Acurácias obtidas envolvendo o fluxo de imagens PKLot → CNRPark-Ext. 63
7.8 Valores-p relacionados ao número de detecções de mudanças de conceito no cenário com o fluxo PKLot → CNRPark-Ext. 64
7.9 Valores-p obtidos com o teste de Nemenyi em relação às acurácias obtidas para o fluxo de imagens PKLot → CNRPark-Ext. 64
7.10 Número de detecções de mudanças de conceito envolvendo o fluxo de imagens PKLot → CNRPark-Ext. 66
7.11 Valores-p relacionados ao número de detecções de mudanças de conceito no cenário com o fluxo PKLot → CNRPark-Ext. 66
7.12 Valores- p obtidos com o teste de Nemenyi em relação ao número de detecções de mudanças de conceito para o fluxo de imagens PKLot → CNRPark-Ext. 66

Lista de Acrônimos

CNN Convolutional Neural Network

GAN Generative Adversarial Network

DCGAN Deep Convolutional Generative Adversarial Network

WGAN Wasserstein Generative Adversarial Network

VGG Visual Geometric Group

ResNet Residual Net

1

Introdução

O avanço tecnológico e o aumento da acessibilidade digital impulsionaram uma produção massiva e diversificada de dados. Esse crescimento oferece vantagens competitivas para quem consegue extrair informações úteis. Dispositivos móveis, redes sociais e equipamentos de diagnóstico médico, entre outras fontes, geram fluxos contínuos de dados em taxas crescentes (EMC Education Services, 2015).

Nesse contexto, surge a necessidade de processar dados de forma instantânea, caracterizando o chamado fluxo contínuo de dados — uma sequência possivelmente infinita de informações, geralmente ordenada temporalmente e produzida em tempo real. Exemplos práticos incluem sistemas de monitoramento por vídeo, como a supervisão de vagas em estacionamentos. Esses fluxos de dados frequentemente possuem volatilidade e enfrentam restrições de processamento e memória, já que o volume de dados tende a crescer com o passar do tempo (GAMA, 2010).

A volatilidade dos dados, caracterizada por alterações ao longo do tempo, é conhecida como mudança de conceito. Utilizando o exemplo anterior, essas mudanças podem dizer respeito à variações no ângulo e posicionamento de câmeras de monitoramento, além de alterações de iluminação, condições climáticas e padrões de ocupação. Em cenários que demandam a previsão de informações com técnicas de aprendizado supervisionado, mudanças de conceito comprometem a eficácia do modelo, uma vez que ele é treinado em uma distribuição de dados específica e, com o tempo, pode perder acurácia à medida que a distribuição se altera.

Para mitigar essa queda de desempenho, é fundamental detectar mudanças de conceito e adaptar o modelo preditivo à nova distribuição de dados. Em fluxos contínuos de dados, essa adaptação é facilitada por detectores de mudança de conceito, que sinalizam a necessidade de atualização do modelo e sugerem novos conjuntos de dados para treinamento.

Por outro lado, as redes generativas, em particular as Redes Generativas Adversárias (GANs) (GOODFELLOW et al., 2014), vêm se destacando pela geração de dados sintéticos. As DCGANs são compostas por duas redes neurais que competem entre si: um gerador, que sintetiza dados a partir de ruído aleatório, e um discriminador, que classifica as imagens como reais ou sintéticas, fornecendo *feedback* ao gerador. Apesar do sucesso das DCGANs na geração de imagens sintéticas, seu uso em fluxos contínuos de dados é ainda pouco explorado, com poucos estudos aplicando ambas as abordagens conjuntamente, sendo um raro exemplo disso o trabalho de GUZY; WOźNIAK; KRAWCZYK (2021).

Este estudo investigará a aplicação de redes DCGAN em um cenário de monitoramento de vagas de estacionamento, avaliando como as imagens sintetizadas impactam o desempenho de um modelo preditivo em um fluxo contínuo de dados. Para isso, serão utilizadas as bases públicas PKLot (DE ALMEIDA et al., 2015) e CNRPark-Ext (AMATO et al., 2017). A base PKLot contém cerca de 695.000 imagens segmentadas divididas em três subconjuntos, captadas em dois estacionamentos com diferentes posicionamentos de câmera, enquanto a CNRPark-Ext possui aproximadamente 145.000 imagens de vagas segmentadas, capturadas simultaneamente por nove câmeras em um único estacionamento, variando em ângulos e posicionamentos.

Essas bases possuem atributos temporais (data e hora da captação das imagens), o que permite simular um contexto de fluxo contínuo de imagens com mudanças de conceito, podendo estas ser caracterizadas como mudanças de ângulo e de câmera, bem como mudanças climáticas, de iluminação e de taxas de ocupação do estacionamento.

1.1 Motivação e Hipótese

O crescimento das aplicações de mineração de fluxos contínuos de dados, impulsionado pelo avanço computacional, aumentou o interesse na pesquisa sobre mudanças de conceito. Essas mudanças afetam modelos preditivos, e entender

como mitigá-las é essencial para adaptar os modelos a novas distribuições de dados

Apesar desses avanços, ainda são escassos os estudos que investigam o uso de redes generativas em cenários de fluxos contínuos de dados, visando entender seus potenciais benefícios e limitações nesse contexto. Um ponto de motivação para este trabalho é o estudo conduzido por GUZY; WOźNIAK; KRAWCZYK (2021), no qual DCGANs são aplicadas a fluxos de dados de imagens, mas seu foco está na proposição de métricas para avaliar e monitorar o impacto das mudanças de conceito em redes generativas, sem explorar diretamente seu uso na atualização de modelos preditivos.

Este trabalho propõe investigar o uso de redes DCGAN como método de suporte para atualização de modelos preditivos em resposta a mudanças de conceito detectadas, especificamente em bases de dados de imagens de estacionamento. A escolha se justifica pela disponibilidade de bases robustas como as bases PKLot e CNRPark-Ext. Apesar de tais bases serem amplamente utilizadas na literatura em contextos de aprendizado em lote, detecção de objetos e adaptação de contexto, ambas fornecem informações temporais, o que permite ordenar os dados para simular um problema de mineração de fluxos contínuos de dados com possíveis mudanças de conceito devido a mudanças de câmera, clima e iluminação, por exemplo .

A pergunta de pesquisa deste trabalho questiona se o uso de redes DCGAN como método de aumento de dados para a atualização do modelo preditivo ao longo de um fluxo de imagens melhora a performance em termos de acurácia quando comparado à outras estratégias, como aumento de dados convencional e apenas a utilização de dados reais nesse processo.

A hipótese deste trabalho é que a aplicação de redes DCGAN como método auxiliar durante a ocorrência de mudanças de conceito pode reduzir os efeitos dessas mudanças, proporcionando maior acurácia ao modelo preditivo e diminuindo o número de ocorrências de mudanças de conceito (quedas de performance) ao longo do fluxo.

1.2 Objetivos

O objetivo principal desta pesquisa é investigar a aplicação de redes DCGAN para sintetização de dados e avaliar seu impacto na adaptação de modelos de

aprendizado de máquina a mudanças de conceito em bases de imagens de estacionamento. Os objetivos específicos incluem:

- Implementar DCGANs em um contexto com ocorrência de mudanças de conceito;
- Avaliar o impacto no número de mudanças de conceito detectadas ao longo de um fluxo de dados com o uso de DCGANs;
- Analisar o impacto do aumento de dados provido por redes DCGAN no fluxo de dados diante da presença de mudanças de conceito;
- Realizar uma avaliação crítica dos resultados obtidos em todos os experimentos.

1.3 Organização

Este trabalho está organizado da seguinte forma: os Capítulos 2 e 3, abordam Fluxos Contínuos de Dados, Redes Convolucionais (CNNs) e Redes Generativas Adversárias (GANs), respectivamente, cobrindo toda a fundamentação teórica necessária para a pesquisa. O Capítulo 4 apresenta sobre as bases de dados utilizadas, com uma análise detalhada se suas características. No Capítulo 5 , são discutidos trabalhos relacionados ao uso de redes DCGAN e técnicas de aumento de dados em cenários de fluxos de dados contínuos. O Capítulo 6 descreve a metodologia adotada nos experimentos realizados, enquanto o Capítulo 7 discute os resultados obtidos. Por fim, o Capítulo 8 apresenta as conclusões do estudo e sugere direções para trabalhos futuros.

2

Mineração de Fluxos Contínuos de Dados

Com o avanço e crescente interesse em inteligência artificial, áreas como indústria, comércio, educação e saúde têm explorado os dados gerados em volumes cada vez maiores para obter vantagens competitivas. Esse cenário representa um desafio crescente para a pesquisa, dado o aumento contínuo na produção de dados em contextos variados.

Consequentemente, a demanda por processamento e extração de conhecimento em fluxos de dados contínuos, definidos como sequências potencialmente infinitas de dados, tem crescido (GAMA, 2010). Um fluxo de dados $\mathcal{S}$ é uma sequência ordenada e potencialmente infinita de o qual produz instâncias de dados $\vec{x_i}$ a um determinado tempo i, podendo ser representada por $\mathcal{S} = \{\vec{x_1}, \vec{x_2}, ..., \vec{x_\infty}\}$. Para acessar um valor qualquer v em uma instância $\vec{x_i}$ utiliza-se a notação $\vec{x_{i,v}}$.

Devido à sua natureza dinâmica, esses fluxos impõem desafios que dificultam a aplicação de técnicas convencionais de aprendizado de máquina, projetadas para operar com dados em lote (GAMA, 2010). Além disso, há restrições de memória e processamento, pois os dados exigem resposta em tempo real e monitoramento contínuo do modelo (GAMA, 2010).

A restrição de memória decorre do volume potencialmente infinito dos dados, o que torna inviável o armazenamento completo, independentemente da complexidade do ambiente de execução do modelo. Já a restrição de processamento exige que o modelo responda rapidamente a cada nova amostra,

processando-a uma única vez antes de descartá-la, devido à alta taxa de geração de dados.

Além dessas restrições, fluxos contínuos de dados são dinâmicos e não podem ser considerados estacionários, ou seja, suas distribuições de classes e características podem mudar ao longo do tempo. Modelos convencionais tendem a perder desempenho frente a essas mudanças, demandando treinamentos frequentes, o que não é viável em cenários que exigem resposta imediata.

Essas questões levantam um problema central: como o modelo deve selecionar os atributos mais relevantes do vetor de características, descartando aqueles redundantes, dentro das restrições impostas pelos fluxos de dados? Para endereçar essa questão, surgem os Algoritmos de Aprendizagem Adaptativa (*Adaptive Learning Algorithms*), que oferecem soluções específicas para classificação em fluxos contínuos de dados. Esses algoritmos utilizam técnicas de amostragem, aleatoriedade e operadores de exclusão e atualização para descartar dados desatualizados (AGGARWAL, 2007; CAUWENBERGHS; POGGIO, 2000).

Assim, as principais características de um modelo para fluxos contínuos de dados incluem incrementabilidade, aprendizado online, tempo constante de processamento por instância e a capacidade de realizar uma única varredura sobre o conjunto de treinamento. Essas características permitem que o modelo responda a mudanças de distribuição ao longo do tempo (DOMINGOS; HULTEN, 2001).

2.1 Mudanças de Conceito

A maioria dos modelos de aprendizado de máquina opera em modo lote, utilizando um lote de instâncias para treinamento uma única vez e aplicando-o a novos dados com base na distribuição vigente no momento do treino. No entanto, em aplicações dinâmicas, as distribuições de probabilidade dos dados frequentemente são não-estacionárias, sofrendo alterações ao longo do tempo devido a fatores temporais ou externos, explícitos ou implícitos no problema.

Essas mudanças de distribuição de probabilidade são conhecidas como **mudanças de conceito**. Em fluxos contínuos de dados, um conceito pode ser entendido como uma função lógica que, ao utilizar um vetor de características $\vec{x}$ disponível, produz um atributo meta y, conforme a equação abaixo (WIDMER; KUBAT, 1996).

$$y = f(\vec{x})$$

De acordo com WEBB et al. (2016), um conceito pode ser definido como um conjunto de vetores onde qualquer objeto no conjunto pertence ao conceito, sendo representado como $\vec{X} \rightarrow Y$. De forma mais prática, a definição probabilística proposta por KUNCHEVA, (2004) sugere que um conceito é a combinação da probabilidade de classes $P(Y)$ e a probabilidade condicional $P(X|Y)$, onde $P(Y)$ e $P(X|Y)$ determinam a distribuição conjunta $P(X, Y)$ (GAMA et al., 2014).

$$\text{Conceito} = P(X, Y)$$

Baseada no trabalho de BARDDAL, (2015), a Tabela 2.1 apresenta um exemplo prático de conceito, onde o indivíduo decide comprar ações de empresas do setor "Elétrico" com índice P/L (preço sobre lucro) abaixo de 10. .

Atributo	**Domínio**
Setor	[Elétrico, Bancário, Varejo, ...]
P/L (Preço sobre Lucro)	$\mathbb{R}^+$
Compra (classe)	[Sim, Não]
Conceito	Setor = Elétrico $\wedge$ P/L $\leq$ 10 $\Rightarrow$ Compra = Sim

Tabela 2.1: Exemplo de Conceito.

Um dos principais desafios em mineração de fluxos de dados é que os conceitos podem variar ao longo do tempo, frequentemente devido a variáveis não observáveis nos dados (*hidden context*). Usando a definição probabilística, uma mudança de conceito entre tempos t e $t + 1$ é representada como: :

$$P_t(X, Y) \neq P_{t+1}(X, Y)$$

Por exemplo, uma modificação do conceito da Tabela 2.1 pode ocorrer após a privatização de empresas no setor elétrico, causando alta no índice P/L. Assim, o conceito ajustado pode envolver o setor bancário com um limite $P/L \leq 7.5$, como na Tabela 2.2.

Algoritmos convencionais de aprendizado de máquina não detectam nem se adaptam automaticamente a mudanças de conceito. Para cenários de fluxos contínuos, são necessárias abordagens que detectem essas mudanças e adaptem

Atributo	**Domínio**
Setor	[Elétrico, Bancário, Varejo, ...]
P/L (Preço sobre Lucro)	$\mathbb{R}^+$
Compra (classe)	[Sim, Não]
Conceito	Setor = Bancário $\wedge$ P/L $\leq 7.5 \Rightarrow$ Compra = Sim

Tabela 2.2: Exemplo de possível modificação do conceito apresentado na Tabela 2.1.

o modelo ao novo conceito de forma rápida e autônoma, distinguindo mudanças de conceito de ruídos ou anomalias para evitar falsos positivos (GAMA, 2010).

As mudanças de conceito podem ser classificadas de acordo com a relação entre as variáveis de entrada e o atributo meta. A classificação mais comum na área é a divisão entre mudanças **reais** e **virtuais**. Além disso, mudanças de conceito podem variar quanto ao modo e velocidade de ocorrência, sendo categorizadas em: :

- Mudança Abrupta;
- Mudança Gradual;
- Mudança Incremental;
- Mudança Recorrente;
- Mudança por *Blips*.

A seguir, cada uma dessas classificações será abordada, com início nas mudanças reais e virtuais e, em seguida, nas variações quanto à velocidade e ao modo como ocorrem.

2.1.1 Mudança de Contexto Real

Também conhecida como mudança de classe ou alteração de probabilidade a priori, uma mudança de contexto real caracteriza-se pela alteração da probabilidade conjunta entre o atributo meta y e o vetor de atributos de entrada $\vec{x}$ ao longo do tempo. Matematicamente, isso pode ser expresso como:

$$P_t(Y|X) \neq P_{t+1}(Y|X)$$

Um exemplo desse tipo de mudança ocorre quando alterações em leis relacionadas ao imposto de renda modificam as faixas salariais para determinação das alíquotas, o que, por sua vez, altera o conjunto de características relevantes para o cálculo de imposto de um contribuinte.

Resumidamente, uma mudança de contexto real ocorre quando a relação funcional entre o vetor de características $\vec{x}$ e o atributo meta y é alterada de forma permanente, ou seja, $f(\vec{x})$ muda. Isso implica que o modelo, treinado com base no conceito anterior, torna-se inadequado para o novo contexto, exigindo uma adaptação. Esse tipo de mudança geralmente resulta de modificações reais no ambiente ou nos processos que geram os dados.

Quando ocorre uma mudança de conceito real, o limite de decisão entre as classes do modelo é alterado. Na Figura 2.1, cada instância do problema é representada por círculos, enquanto as cores representam as classes. Observa-se que uma instância pertencente a uma determinada classe em um tempo t_1 passa a pertencer a outra classe em t_2, após a mudança de conceito.

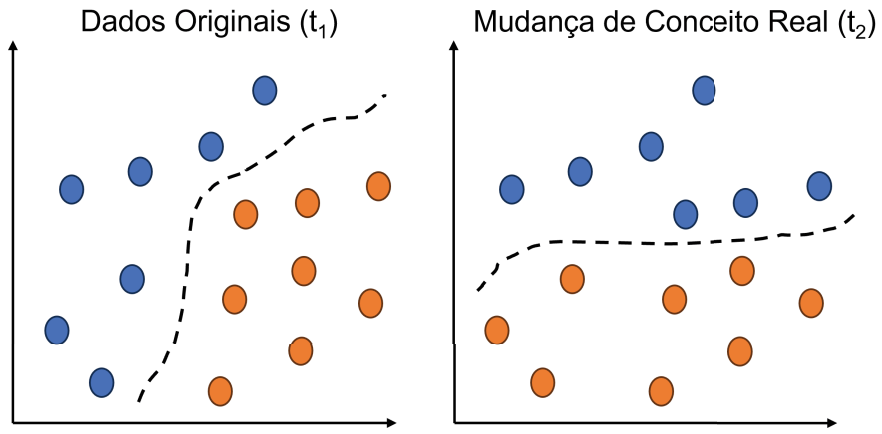


Figura 2.1: Ilustração de Mudança de Conceito Real

Segundo WEBB et al., (2016), uma mudança de conceito real pode ser subclassificada com base na severidade da alteração. Existem dois tipos principais:

- **Mudança Cruzada** (sub-conceito): a alteração se limita a um subespaço do domínio de X. Matematicamente, isso é representado por:

$$P_t(Y|X) \neq P_{t+1}(Y|X) \wedge \exists_{x \in Dom(X)} \forall_{y \in Dom(Y)} P_t(Y = y|X = x) = P_{t+1}(Y = y|X = x)$$

  Em uma mudança cruzada, a probabilidade de Y dado X é alterada, mas há ao menos um subconjunto de X cuja saída permanece a mesma após a mudança de conceito.

- **Mudança Severa** (conceito completo): toda a distribuição de classes para todos os tipos de entrada é alterada. Formalmente, essa mudança é descrita como:

$$\forall_{x \in Dom(X)} \exists_{y \in Dom(Y)} P_t(Y = y | X = x) \neq P_{t+1}(Y = y | X = x)$$

2.1.2 Mudança de Contexto Virtual

Também conhecida como desvio de covariáveis ou *data drift*, a mudança de conceito virtual ocorre quando a distribuição dos dados de entrada X se altera ao longo do tempo, ou seja, $P(X)$ muda. Matematicamente, isso é expresso como:

$$P_t(X) \neq P_{t+1}(X)$$

Um exemplo desse tipo de mudança pode ser observado em uma empresa que utiliza dados socioeconômicos para prever o comportamento de clientes e o volume de vendas. Com o tempo, mudanças nas condições demográficas dos clientes podem afetar esses fatores e, consequentemente, o desempenho do modelo preditivo.

Na Figura 2.2, é ilustrada uma mudança de conceito virtual. Nesse caso, a fronteira de decisão entre as classes permanece inalterada; o que muda é a distribuição dos dados de entrada. Diferentemente da mudança de conceito real, a probabilidade condicional $P(Y|X)$ permanece constante, enquanto apenas $P(X)$ é alterada.

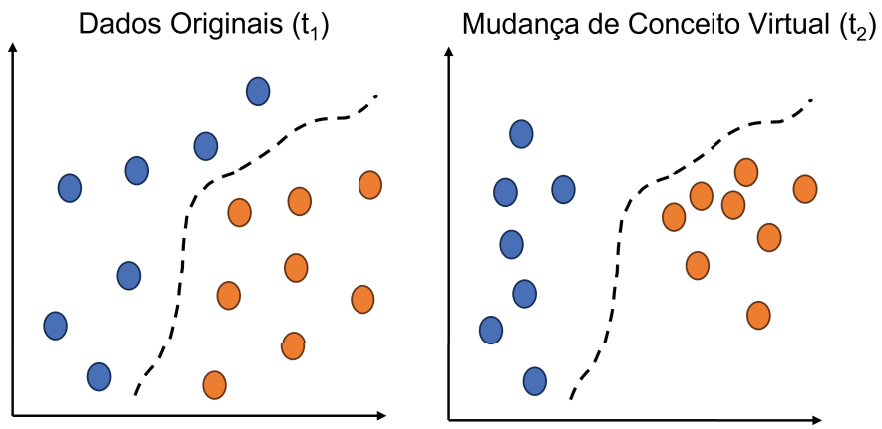


Figura 2.2: Ilustração de Mudança de Conceito Virtual

A seguir, serão discutidos os tipos de mudança de conceito de acordo com o modo e a velocidade com que ocorrem.

2.1.3 Mudança de Contexto Abrupta

Quando um novo conceito substitui o conceito anterior de forma imediata, ocorre uma mudança de conceito abrupta ou repentina, geralmente identificada no momento t em que ocorre (AGRAHARI; SINGH, 2022). Esse tipo de mudança é caracterizado por uma degradação súbita nas métricas de monitoramento do modelo.

De acordo com WEBB et al. (2016), uma mudança abrupta se classifica como uma alteração que leva em conta a duração da transição entre conceitos. A Figura 2.3 ilustra esse tipo de mudança, onde as formas e cores representam conceitos específicos. Observa-se que, em um determinado instante t, o conceito é alterado de forma imediata, sem a presença de transições intermediárias entre o conceito antigo e o novo.

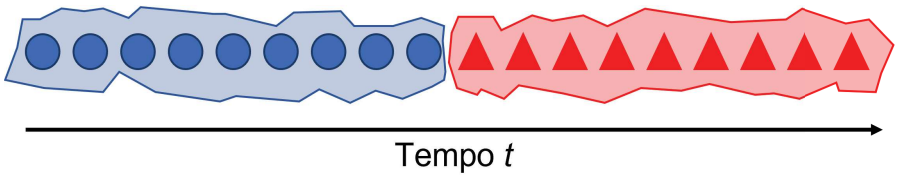


Figura 2.3: Ilustração de Mudança de Conceito Abrupto

Além disso, segundo WEBB et al. (2016), uma variação da mudança abrupta é a mudança por *blips*. Esse é um caso específico, e será abordado separadamente em uma seção dedicada.

2.1.4 Mudança de Contexto Gradual

A mudança de conceito gradual ocorre quando há um estado intermediário entre os dois conceitos envolvidos onde existem ocorrências tanto do novo conceito quanto do antigo, durante todo o período que a mudança leva para ser completada.

Segundo WEBB et al. (2016), a mudança gradual é caracterizada pelo modo de transição entre conceitos. A Figura 2.4 ilustra essa dinâmica, mostrando como ambos os conceitos podem estar presentes simultaneamente até que o novo conceito se estabeleça completamente.

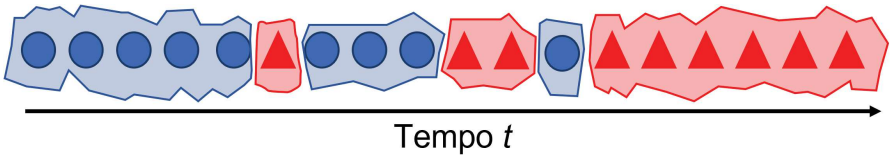


Figura 2.4: Ilustração de Mudança de Conceito Gradual

2.1.5 Mudança de Contexto Incremental

A mudança de conceito é classificada como incremental quando ocorre uma transição gradual, na qual o conceito evolui progressivamente do antigo para o novo. Durante essa transição, o conceito apresenta características que não são exclusivas de um ou outro estado, mas representam uma adaptação gradual da distribuição dos dados, até que o novo conceito seja totalmente estabelecido.

Segundo WEBB et al. (2016), a mudança incremental é categorizada pelo modo de transição entre conceitos. A Figura 2.5 ilustra essa dinâmica.

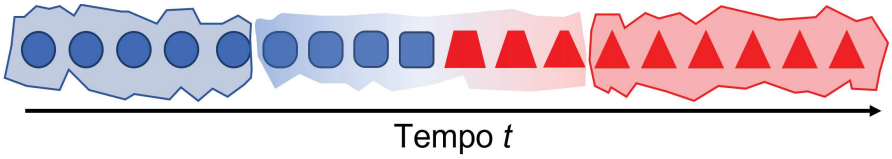


Figura 2.5: Ilustração de Mudança de Conceito Incremental

Um exemplo prático é a área de detecção de fraudes em cartões de crédito, onde a introdução de cartões com tecnologia de pagamento por aproximação resulta em novas formas de fraude. Com a adoção crescente dessa tecnologia, o cenário de fraudes evolui até que o novo conceito, baseado nessa nova realidade tecnológica, seja completamente estabelecido.

2.1.6 Mudança de Contexto Recorrente

A mudança de contexto recorrente ocorre quando um conceito que existiu no passado volta a se manifestar no futuro, alternando-se com outros conceitos ao longo do tempo. Esse tipo de mudança é particularmente interessante e desafiador, pois exige o armazenamento dos estados do modelo para que ele possa se readaptar rapidamente ao conceito quando este reaparecer.

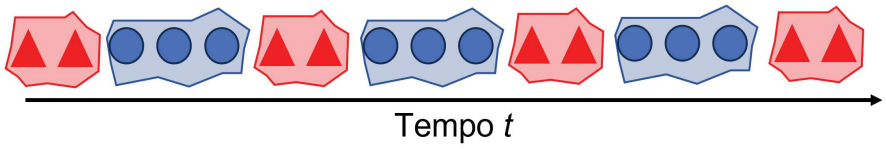


Figura 2.6: Ilustração de Mudança de Conceito Recorrente

A Figura 2.6 ilustra uma mudança de contexto recorrente envolvendo dois conceitos alternados. WEBB et al. (2016) classificam a mudança recorrente em duas categorias: cíclica e não cíclica.

Mudança Recorrente Cíclica: ocorre em ciclos regulares, alternando-se entre conceitos em uma ordem específica. Um exemplo cotidiano são os padrões climáticos, como as quatro estações do ano, onde cada estação (conceito) se repete anualmente com características específicas, como maior umidade e temperatura no verão e menor temperatura no inverno.

WEBB et al. (2016) também subdividem a mudança cíclica em duas variações, de acordo com a duração do ciclo de cada conceito:

- **Frequência Fixa**: ocorre em intervalos regulares e conhecidos, o que facilita a adaptabilidade do modelo, pois é possível prever a transição entre ciclos e ajustar o modelo para os períodos de estabilidade de cada conceito.
- **Frequência Variável**: os ciclos não têm intervalos fixos, tornando a detecção e adaptação às mudanças mais complexas, pois o modelo deve lidar com variações imprevisíveis nos períodos em que os conceitos ocorrem.

Mudança Recorrente Não Cíclica: como o próprio nome já diz, mudanças de conceito recorrentes que não obedecem a um ciclo de tempo determinado, o que dificulta sua detecção e adaptação do classificador.

2.1.7 Mudança de Contexto por *Blips*

A mudança de contexto por *blips* é uma forma específica de mudança abrupta de conceito, caracterizada pela curta duração do novo conceito, que substitui o anterior por um breve período. A Figura 2.7 ilustra esse tipo de mudança.

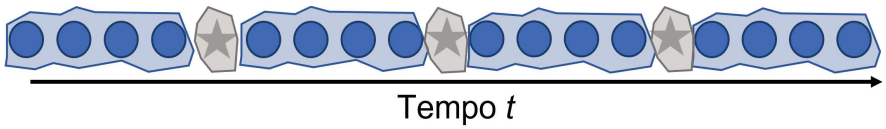


Figura 2.7: Ilustração de Mudança de Conceito por *Blips*

Atualmente, há debate sobre a necessidade de adaptação dos modelos a mudanças por *blips*, dada sua curta duração. Alguns defendem que essas mudanças deveriam ser ignoradas ou tratadas como valores anômalos (*outliers*), enquanto outros apontam que, se o modelo se adapta a um *blip*, sua acurácia pode diminuir caso ele não consiga retornar ao conceito original rapidamente após o evento.

Segundo WEBB et al. (2016), embora muitos sistemas sejam projetados para ignorar mudanças por *blips*, elas não devem ser confundidas com *outliers*. Enquanto um *outlier* é uma única amostra anômala em relação ao conceito atual, um *blip* representa uma curta sequência de instâncias que formam um conceito temporário.

Um exemplo comum de *blip* é a "*Black Friday*", quando lojas de varejo oferecem grandes descontos e realizam campanhas publicitárias intensivas. Esse evento anual pode ser considerado um *blip* em fluxos de dados de vendas, uma vez que as vendas aumentam significativamente apenas por um breve período.

Por fim, vale a pena salientar que a duração de uma mudança para esta ser considerada um *blip* depende do contexto aplicado, e, geralmente, difere de caso para caso.

2.2 Detectores de Mudanças de Conceito

A maioria dos modelos de aprendizado de máquina é desenvolvida com o pressuposto de que as amostras processadas são geradas aleatoriamente sob uma mesma distribuição estacionária (GAMA, 2010). Entretanto, como discutido anteriormente neste Capítulo muitas aplicações de fluxos contínuos de dados apresentam distribuições que mudam ao longo do tempo, refletindo a natureza dinâmica do problema.

Quando essas mudanças de distribuição ocorrem o desempenho do modelo tende a diminuir, exigindo estratégias adequadas para detectar e tratar as mudanças de conceito ao longo do fluxo de dados.

Em problemas não estacionários, onde a distribuição dos dados varia temporalmente, podem ocorrer fases de transição entre sequências que representam diferentes conceitos (GAMA, 2010). O principal desafio para algoritmos de detecção de mudanças de conceito é identificar essas mudanças no momento exato em que ocorrem, pois as transições incluem amostras que podem pertencer tanto

ao conceito anterior quanto ao novo conceito. Assim, amostras de um conceito em $t + 1$ podem ser percebidas como ruído para o conceito em t .

Para lidar com esse problema, algoritmos de detecção precisam distinguir entre ruído e mudanças reais, baseando-se na persistência: uma mudança é estabelecida apenas quando há um conjunto consistente de amostras da nova distribuição. Esses algoritmos devem combinar robustez contra ruídos com sensibilidade às mudanças de conceito, evitando a detecção de falsos drifts, mas também identificando-os rapidamente para minimizar o processamento de amostras sob um modelo treinado em uma distribuição desatualizada.

De acordo com GAMA (2010), métodos de detecção de mudanças de conceito geralmente assumem que as amostras mais recentes são mais relevantes do que as anteriores.

Nesta pesquisa será utilizado o método de detecção de mudanças de conceito ADWIN, que será abordado a seguir.

2.2.1 ADWIN

O ADWIN (ADaptive sliding WINdow, ou Janela Deslizante Adaptativa) é um algoritmo de detecção de drifts e estimador de média proposto por BIFET; GAVALDà (2007). Ele utiliza uma janela deslizante com tamanho adaptativo, de modo que o comprimento da janela de dados recentes seja estatisticamente consistente com a hipótese de que "não ocorreram mudanças no valor médio dos dados na janela".

Um valor mais antigo é removido da janela apenas quando há evidências de que sua média diverge da média dos valores mais recentes. Dessa forma, o ADWIN permite duas funcionalidades: detectar mudanças de conceito na janela de dados e estimar a média atual do fluxo de dados a qualquer instante.

O ADWIN possui como entrada um valor de confiança $\delta \in (0, 1)$ e uma sequência de valores reais possivelmente infinita $(x_1, x_2, x_3, ..., x_t, ...)$ onde cada valor de x_t é acessível somente em um instante t. O algoritmo assume que os valores estão no intervalo [0, 1].

A ideia do método ADWIN é a seguinte: sempre que duas subjanelas grandes o suficiente dentro de uma janela deslizante W apresentam médias significativamente diferentes, a subjanela mais antiga é descartada. O teste estatístico realizado pelo ADWIN verifica se a diferença entre as médias das duas subjanelas é maior que um valor limite ϵ_{cut}, calculado pela fórmula:

$$m = \frac{2}{1/|W_0| + 1/|W_1|}$$
$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \cdot \ln \frac{4|W|}{\delta}}$$

Onde $|W|$, $|W_0|$ e $|W_1|$ são os tamanhos da janela e subjanelas utilizadas pelo ADWIN.

O algoritmo ADWIN baseado em GAMA (2010), é descrito a seguir:

Algoritmo 1 Algoritmo do método de detecção de mudanças de conceito ADWIN.

1: Inicia Janela W;
2: **for** each $(t) > 0$ **do**
3: $\quad W \leftarrow W \cup \{x_t\}$ ▹ Adiciona x_t ao início da Janela W
4: $\quad$ **repeat**
5: $\quad\quad$ Exclui elementos mais antigos de W
6: $\quad$ **until** $|\hat{\mu}_{W_0} - \hat{\mu}_{W_1}| < \epsilon_{cut}$
7: $\quad W = W_0 \cdot W_1$
8: **end for**
9: Saída $\hat{\mu}_W$

3

Aprendizado Profundo

Ao longo de toda a história, a humanidade sempre almejou possuir máquinas que fossem capazes de raciocinar e realizar ações como trabalhos rotineiros, entender imagens ou falas e realizar diagnósticos médicos (GOODFELLOW; BENGIO; COURVILLE, 2016). Atualmente, a inteligência artificial cresce rapidamente, resolvendo problemas intelectualmente complexos para os seres humanos.

O que têm facilitado muito esse fenômeno é uma solução que permite que as máquinas aprendam a partir de uma experiência prévia, entendendo o mundo em termos de hierarquia de conceitos. Tal hierarquia habilita o computador a aprender conceitos complicados e transformá-los em outros mais simples. Essa abordagem é conhecida dentro do campo de computação como aprendizado profundo (GOODFELLOW; BENGIO; COURVILLE, 2016).

Dentro desse capítulo serão abordadas duas técnicas englobadas dentro do mundo de aprendizado profundo, sendo elas as redes neurais convolucionais (CNNs) na seção 3.1 e as redes generativas adversárias (GANs) na seção 3.2.

3.1 Redes Neurais Convolucionais

Nesta seção serão abordadas as Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs), introduzidas por LECUN et al. em 1998, suas características, aplicações e os cenários de extração de características, além do uso de modelos de CNNs pré-treinados.

As CNNs são técnicas de aprendizado profundo, frequentemente associadas à área de Visão Computacional. A principal diferença entre as CNNs e outros modelos de aprendizado de máquina, sejam rasos ou não, reside na capacidade das CNNs de aprenderem automaticamente filtros e representações profundas a partir das imagens, em vez de dependerem de características previamente definidas. Esse fenômeno é conhecido como *feature learning*.

Para ilustrar o conceito de *feature learning*, considere a imagem de um cachorro. A CNN aprenderá padrões como os olhos, orelhas, e nariz, entre outros aspectos distintivos da imagem. Em contraste, modelos tradicionais, que exigem características pré-definidas, limitam-se a analisar a imagem como um todo, sem identificar essas características de forma autônoma.

As CNNs apresentam duas propriedades fundamentais, destacadas por CHOLLET (2017):

1. **Invariância à translação**: Uma vez que um padrão é aprendido, a rede pode reconhecê-lo independentemente de sua posição na imagem, o que a torna eficiente no processamento de dados visuais e permite que o modelo exija menos amostras para treinar, mantendo boa capacidade de generalização.

2. **Capacidade de aprender hierarquias espaciais de padrões**: Camadas iniciais da rede identificam padrões locais, como bordas e contornos, enquanto camadas subsequentes aprendem padrões mais complexos, a partir dos padrões detectados anteriormente, permitindo um aprendizado progressivo e abstrato.

A estrutura clássica de uma CNN é ilustrada na Figura 3.1, onde se destacam as camadas de convolução e *pooling*.

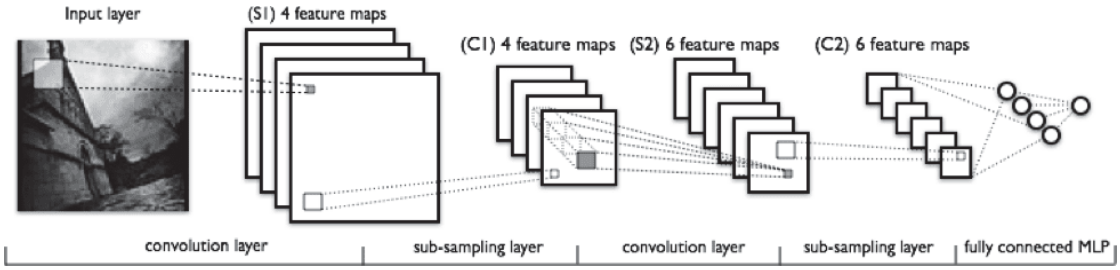


Figura 3.1: Estrutura de uma rede CNN.[1]

[1]Disponível em <https://www.kdnuggets.com/2017/08/convolutional-neural-networks-image-recognition.html>. Acesso em 01 nov. 2023.

A seguir, detalharemos as principais operações envolvidas nas redes neurais convolucionais, com o objetivo de aprofundar o entendimento sobre seu funcionamento.

3.1.1 Operação de Convolução

As redes neurais convolucionais (CNNs) se distinguem de outros modelos por sua capacidade de aprender representações profundas da imagem, por meio da operação de convolução. As convoluções operam sobre tensores tridimensionais chamados mapas de características, que possuem duas dimensões espaciais (altura e largura da imagem) e uma dimensão referente ao número de canais de cor. Por exemplo, uma imagem em preto e branco tem um único canal, enquanto uma imagem RGB possui três canais (vermelho, verde e azul).

A operação de convolução consiste em extrair regiões dos mapas de características de entrada e aplicar a mesma transformação a todas essas regiões, gerando novos mapas de características como saída. Esses atributos continuam representados por tensores 3D, com altura e largura, mas a profundidade da saída pode ser arbitrária. A profundidade não corresponderá mais a um canal de cor específico, mas sim ao número de filtros aplicados na convolução. Cada filtro codifica um aspecto específico dos dados como, por exemplo, a detecção de rostos em uma imagem.

Existem dois parâmetros principais nas convoluções: o tamanho da janela de convolução, geralmente de dimensões pequenas (como 3x3 ou 5x5), e a profundidade do mapa de características de saída, que corresponde ao número de filtros aplicados na operação. Assim, a saída de cada camada de convolução terá dimensões (d_1, d_2, d_3), onde d_1 e d_2 representam as dimensões de altura e largura após a convolução, e d_3 é o número de filtros aplicados.

A operação de convolução é realizada movendo a janela de convolução definida sobre a imagem de entrada, extraindo regiões locais com as dimensões especificadas pela janela e pela profundidade dos mapas de características definidos na parametrização da CNN. Cada região extraída é transformada por meio de um produto entre tensores utilizando a mesma matriz de pesos, chamada *kernel* de convolução, resultando em um vetor unidimensional, cuja dimensão depende do número de filtros aplicados.

Como resultado, há uma perda de informação nas dimensões de altura e largura, uma vez que a saída da convolução será menor do que a entrada. Essa

redução pode ocorrer devido a dois fatores: **efeitos de borda**, que podem ser evitados com o uso de preenchimento (*padding*) no mapa de características de entrada, e o uso de **avanço** (*strides*), que será discutido mais adiante.

Para ilustrar o efeito de borda e a função do *padding*, considere uma imagem de entrada 5x5 e uma janela 3x3. Nesse caso, a janela pode ser centralizada em apenas 9 posições, resultando em uma saída de 3x3. A redução nas dimensões de altura e largura pode ser generalizada: a imagem é reduzida em $n - 1$ nas dimensões de altura e largura, onde n é o tamanho da janela de convolução. Tal efeito pode ser melhor entendido na Figura 3.2.

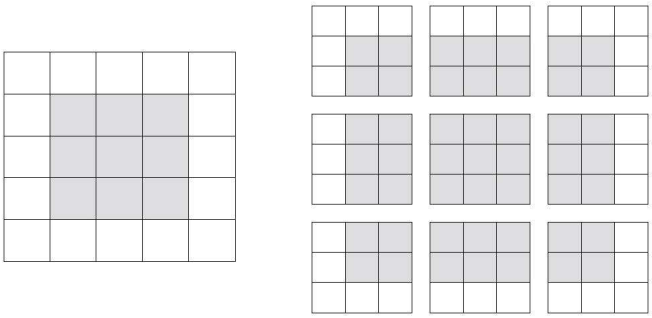


Figura 3.2: Efeito de borda em uma imagem 5x5 com uma janela 3x3 - Fonte: (CHOLLET, 2017)

A técnica de *padding* adiciona linhas e colunas ao redor do mapa de características de entrada, permitindo que a janela de convolução se centralize em cada ponto da imagem. Por exemplo, para uma janela 3x3, adiciona-se uma linha e uma coluna em cada lado da imagem, e para uma janela 5x5, adiciona-se duas linhas e colunas em cada lateral.

O parâmetro de ***stride*** refere-se à quantidade de pixels que a janela de convolução avança em cada iteração. Quanto maior o valor de *stride*, menor será a saída da camada de convolução, uma vez que as iterações são reduzidas. A diminuição das dimensões de saída pode ser calculada dividindo as dimensões esperadas (com ou sem *padding*) pelo valor de *stride*.

Na Figura 3.1, as camadas denominadas S1 e S2 são camadas de convolução. Nelas, o número de mapas de características gerados em cada camada é determinado pelo número de filtros de convolução especificado na parametrização da rede.

Vale destacar que é raro o uso de *strides* maiores que 1, mas é importante compreender seu conceito e os efeitos sobre o modelo, pois eles são úteis em

situações específicas. Em conjunto com as camadas de convolução, a operação de *pooling* (agrupamento) é frequentemente utilizada, e será abordada na seção seguinte.

3.1.2 Operação de *Pooling*

As camadas de *pooling* em uma CNN têm como objetivo reduzir o conjunto de atributos de entrada, diminuindo as dimensões dos mapas de características. Esse processo pode ser comparado ao uso de um *stride*, com a diferença de que o *pooling* não apenas reduz as dimensões, mas também realiza operações específicas sobre os dados. Para isso, são utilizadas janelas, geralmente de 2x2, nas quais são aplicados os cálculos adequados.

A principal finalidade do *pooling* é reduzir a quantidade de atributos da rede, ajudando a prevenir o *overfitting*. Além disso, o *pooling* aprimora o aprendizado de padrões locais na imagem já que, ao reduzir as dimensões, somente as características mais importantes tendem a ser preservadas.

Existem diferentes tipos de *pooling*, sendo os mais comuns o *max-pooling* e o *avg-pooling*. Como os nomes sugerem, no *max-pooling*, a operação seleciona o valor máximo de cada janela 2x2, enquanto no *avg-pooling*, a saída é a média de todos os valores presentes na janela de *pooling*.

3.1.3 Camadas Totalmente Conectadas

Ao final de uma CNN, encontram-se as camadas totalmente conectadas, que recebem como entrada a saída da última camada convolucional e possuem como saída N neurônios, onde N corresponde ao número de classes do modelo. Essas camadas reorganizam os mapas de características recebidos em sua entrada para uma forma unidimensional processando-os até a camada final, que representa a classe à qual o dado processado pertence. A reorganização é ilustrada na Figura 3.3, e a localização das camadas totalmente conectadas na rede pode ser vista na Figura 3.4 identificadas como "*Fully Connected Layer*".

O vetor reorganizado é então processado por camadas totalmente conectadas que realizam operações matemáticas ajustando seus neurônios ao problema. A operação matemática realizada pode ser expressa por:

$$g * (W * x + b)$$

Onde:

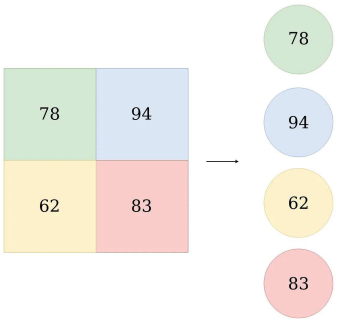


Figura 3.3: Reorganização dos dados em uma Camada Totalmente Conectada.

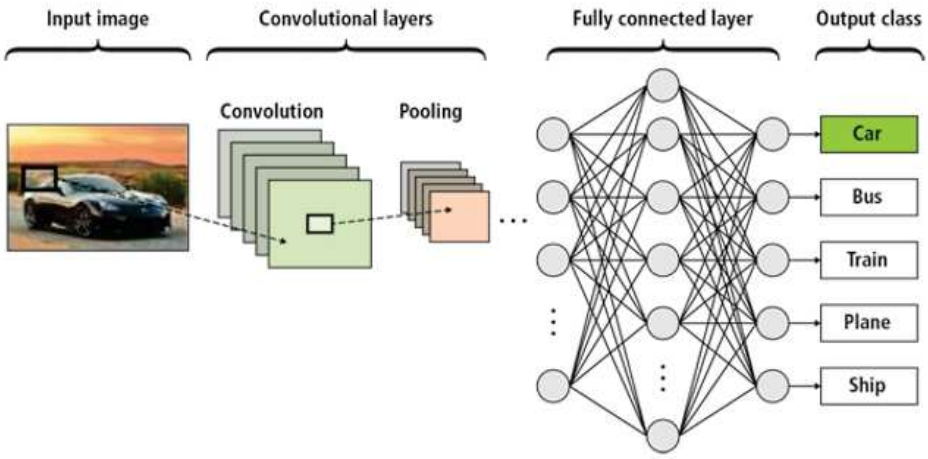


Figura 3.4: Estrutura completa de uma Rede Neural Convolucional. Fonte: NVidia

- x é o vetor de entrada da camada,
- W é a matriz de pesos com dimensões $[p_l, n_l]$, onde p_l é o número de neurônios da camada anterior e n_l é o número de neurônios da camada atual,
- b é o vetor de viés com o mesmo tamanho do número de neurônios da camada anterior,
- g é a função de ativação, frequentemente a *ReLU*.

Por fim, a camada final tem um número de neurônios igual ao número de classes do problema, utilizando a função de ativação *softmax* para representar, em cada neurônio, a probabilidade da amostra pertencer a uma classe específica.

Após a explicação das operações principais dentro de uma rede neural convolucional, na próxima seção serão discutidas suas principais aplicações.

3.1.4 Aplicações

As Redes Neurais Convolucionais (CNNs) são amplamente utilizadas em tarefas relacionadas a imagens, permitindo construir modelos de classificação de imagens, além de outras técnicas como transferência de aprendizado e extração de atributos. A seguir, serão abordadas algumas dessas aplicações.

Classificação de Imagens

O uso mais comum de uma CNN é para classificação de imagens. Nessa abordagem, define-se a estrutura da CNN e utiliza-se um conjunto de imagens para treinamento e validação. O modelo é treinado e validado com base em métricas como acurácia e erro, permitindo avaliar sua capacidade de classificar imagens que não fazem parte do conjunto de treinamento, tornando-o aplicável para resolver o problema proposto.

No processo de classificação, os mapas de características da última camada de convolução e *pooling* são "achatados"para se tornar um vetor unidimensional, que é então alimentado em camadas densas. As camadas densas ajustam os pesos até a camada final, que possui o número de neurônios correspondente ao número de classes.

Exemplos de aplicações incluem a classificação de imagens de animais, onde o modelo deve identificar a espécie de um animal em cada imagem, ou o uso da base de dados MNIST, com dígitos manuscritos, onde o modelo prevê o número retratado em cada imagem.

Transferência de Aprendizado

A transferência de aprendizado é uma técnica em que uma rede previamente treinada em um grande conjunto de dados é reutilizada para resolver um problema com um conjunto de dados diferente. Essa técnica é útil quando não há dados suficientes para treinar uma rede convolucional (CNN) do zero, o que pode levar ao *overfitting* (quando o modelo se ajusta excessivamente aos dados de treinamento, comprometendo sua generalização).

Uma das bases mais comuns para redes pré-treinadas é a *ImageNet* (DENG et al., 2009), uma base hierárquica com aproximadamente 3,2 milhões de imagens rotuladas em 5.247 categorias. A utilização dessa base é justificada pelo seu tamanho e variedade, permitindo que modelos treinados nela adquiram

conhecimento útil para tarefas gerais de classificação de imagens. Na Figura 3.5 podemos visualizar alguns exemplos de classes disponível dentro da *ImageNet*.

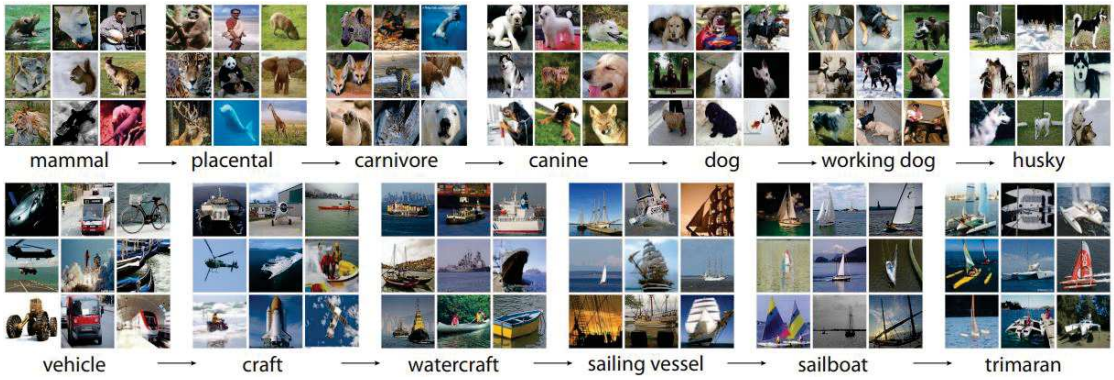


Figura 3.5: Exemplos de imagens e suas classes pertencentes à *ImageNet* - Fonte: (DENG et al., 2009)

Além disso, redes muito profundas exigem alto custo computacional para treinamento. Com redes pré-treinadas, esse custo é reduzido, pois o treinamento já foi realizado anteriormente.

Um exemplo de uso de uma rede CNN para esse tipo de aplicação ocorre quando há poucas amostras para treinamento, como quando supomos que, no caso da base MNIST por exemplo, têm-se apenas 10 instâncias por classe, a transferência de aprendizado permite que a rede seja treinada de forma mais eficiente. Nesse cenário, a rede pré-treinada pode ser adaptada para o problema atual, usando menos dados.

O processo de transferência de aprendizado pode ser explicado com os seguintes passos:

1. **Congelamento das camadas convolucionais**: As camadas de convolução da rede pré-treinada são mantidas congeladas, ou seja, seus pesos não são atualizados.
2. **Descongelamento das camadas totalmente conectadas**: Apenas as camadas densas são treinadas, permitindo que o modelo se adapte ao novo problema.
3. **Adição de camadas**: É possível adicionar novas camadas ao final da rede para ajustar o modelo ao número de classes do problema atual.

Dessa forma, a transferência de aprendizado permite que redes pré-treinadas, com conhecimento adquirido de grandes bases de dados, sejam usadas para resolver problemas com menor quantidade de dados e menos custo computacional.

Extração de Atributos

A técnica de extração de atributos utiliza as representações aprendidas por uma rede neural convolucional para extrair características relevantes de novas instâncias de dados. Essas características podem ser aplicadas em outros classificadores, que podem ser treinados com um número reduzido de amostras. Nesse processo, utiliza-se apenas a parte convolucional da rede, isto é, as camadas de convolução e *pooling*, descartando as camadas densas. O objetivo é extrair as representações de alto nível fornecidas pelas camadas convolucionais e utilizá-las como entrada para outros modelos.

Um exemplo dessa abordagem é a combinação de transferência de aprendizado com extração de atributos, em que uma rede pré-treinada em uma grande base de dados é usada para extrair características de imagens, que são então aplicadas em outro classificador para resolver um problema distinto.

3.2 Redes Generativas Adversárias

As Redes Generativas Adversárias (GANs) são técnicas de aprendizado profundo utilizadas para a geração de conteúdo multimídia sintético a partir de dados de treinamento, com destaque para o caso de imagens. As GANs fazem parte da modelagem generativa, cujo objetivo é criar novos dados realistas a partir de dados reais, sejam eles rotulados ou não permitindo, por exemplo, alterar a distribuição dos dados de treinamento de um modelo preditivo.

Introduzidas por GOODFELLOW et al. (2014), as GANs combinam o poder de modelos discriminativos, usados em problemas de classificação e regressão, para construir modelos generativos eficientes. A principal teoria por trás das GANs é que o gerador de dados atinge seu objetivo quando um discriminador não é capaz de distinguir os dados gerados dos reais.

O treinamento das GANs envolve uma competição entre dois modelos: o gerador, que cria dados sintéticos, e o discriminador, que tenta distinguir os dados gerados dos reais. Ambos são treinados simultaneamente, com o gerador buscando melhorar suas gerações e o discriminador refinando sua capacidade de identificar dados falsificados. As seções seguintes detalham o processo de treinamento de uma GAN e a arquitetura específica de uma *Deep Convolutional Generative Adversarial Network* (DCGAN).

3.2.1 Estrutura e Funcionamento de uma GAN

Uma rede GAN é composta por dois componentes principais: o gerador e o discriminador. Ambos são redes neurais, com o gerador responsável pela criação de novas instâncias de dados e o discriminador encarregado de avaliar a autenticidade dos dados gerados.

O gerador recebe um vetor de ruído aleatório como entrada e gera uma nova instância (geralmente uma imagem). Essa imagem gerada é então fornecida ao discriminador, junto com imagens reais. O discriminador avalia a imagem e retorna uma probabilidade, variando entre 0 (imagem falsa) e 1 (imagem real), indicando a autenticidade da imagem recebida. Podemos observar a estrutura básica de uma rede GAN na Figura 3.6 abaixo.

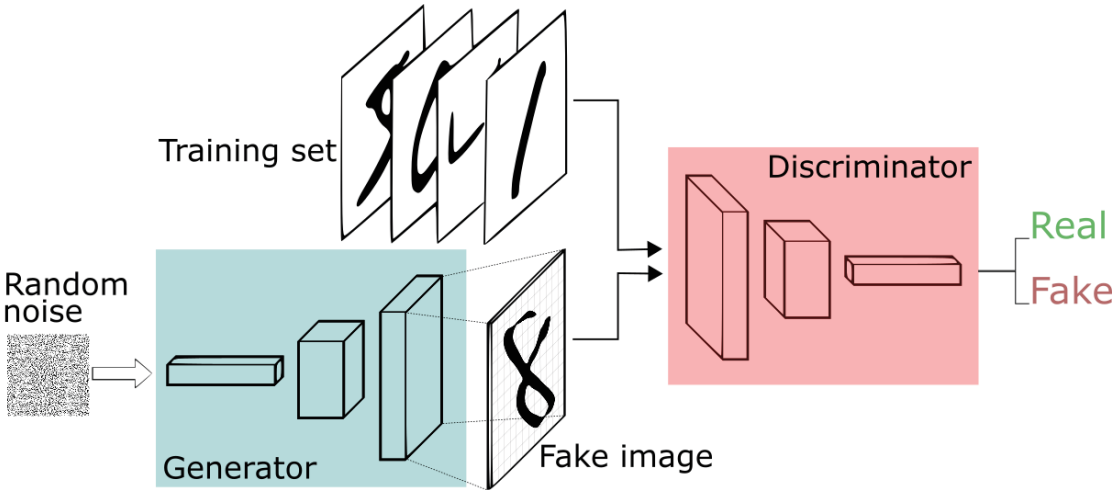


Figura 3.6: Estrutura de uma Rede GAN. Fonte: (SILVA, 2017)

A estrutura básica de uma GAN envolve uma rede convolucional no discriminador, com saída binária (classificando imagens como reais ou falsas). O gerador realiza o processo inverso: recebe um vetor de ruído e, por meio de operações de *upsampling*, aumenta as dimensões do vetor até que ele tenha o formato de uma imagem. Essa imagem gerada é então analisada pelo discriminador.

Ambas as redes buscam otimizar suas funções objetivo, com o gerador tentando enganar o discriminador e o discriminador tentando melhorar sua capacidade de distinguir imagens reais de falsas. Essa competição entre o gerador e o discriminador é o que dá origem ao termo "adversárias". Em resumo, o gerador busca minimizar sua função de perda enquanto o discriminador possui como objetivo maximizá-la.

O objetivo principal de uma GAN é aproximar as distribuições de probabilidade das imagens geradas (P_g) e das imagens reais (P_r), o que significa que o gerador aprende a criar imagens que são suficientemente realistas para enganar o discriminador, confundindo-as com imagens reais.

3.2.2 *Deep Convolutional Generative Adversarial Network*

Com o avanço da pesquisa em geração de conteúdo artificial, diversas variantes de redes GAN foram desenvolvidas para atender a diferentes propósitos. Nessa seção, é abordada uma dessas variantes, que foi utilizada em trabalhos mencionados no Capítulo 5 e foi aplicada durante os experimentos deste trabalho.

A *Deep Convolutional Generative Adversarial Network* (DCGAN), proposta por RADFORD; METZ; CHINTALA (2016), é uma modificação da rede GAN original, utilizando redes convolucionais profundas em vez de redes totalmente conectadas. A Figura 3.7 ilustra a estrutura do gerador de uma DCGAN e suas camadas convolucionais.

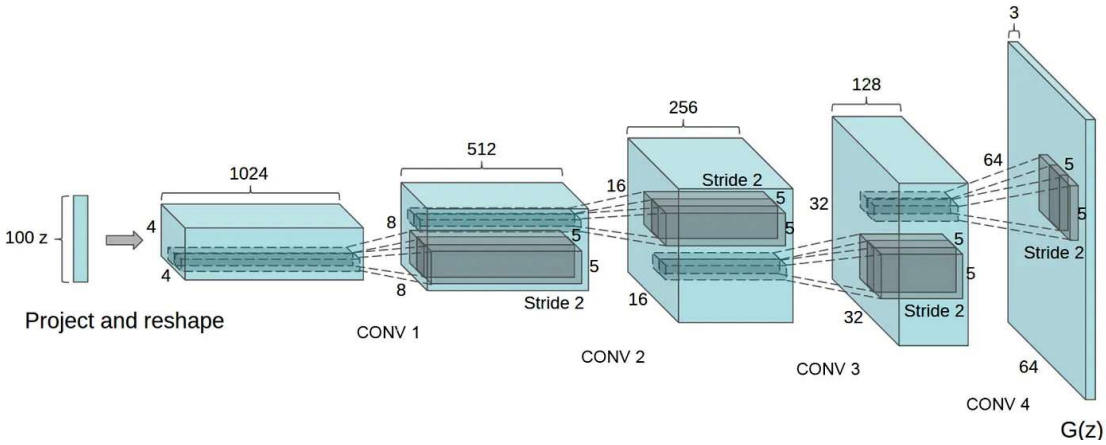


Figura 3.7: Estrutura do gerador de uma DCGAN. Fonte: (RADFORD; METZ; CHINTALA, 2016)

O uso de camadas convolucionais profundas na DCGAN é justificado pelo fato de que camadas mais profundas conseguem capturar padrões em áreas menores da imagem, permitindo identificar correlações espaciais de forma mais eficaz. Isso torna a DCGAN mais adequada para trabalhar com conteúdo visual, como imagens e vídeos, enquanto a rede GAN original possui uma aplicação mais generalista, podendo ser utilizada em uma variedade maior de contextos.

Semelhante à GAN, a DCGAN também possui um gerador e um discriminador, com funcionamento análogo ao explicado anteriormente. A principal diferença está na função de perda utilizada, que na DCGAN é a *binary cross-entropy*. A fórmula da função de perda é dada por:

$$\text{logloss} = \sum_i^N \sum_j^M y_{ij} \log(p_{ij})$$

Onde N é o número de instâncias, M é o número de classes, y_{ij} corresponde à classe real da instância e p_{ij} é a probabilidade de que o classificador determine a instância como real ou falsa (no caso de DCGANs ou GANs em geral).

Por fim, na Figura 3.8 , é possível visualizar uma comparação entre instâncias da base de dados MNIST, as imagens geradas por uma GAN e aquelas geradas por uma DCGAN.

Figura 3.8: Comparação entre imagens reais, geradas por uma GAN e uma DCGAN. Fonte: (RADFORD; METZ; CHINTALA, 2016)

4

Bases de Dados

Nesta seção, são apresentadas as bases de dados utilizadas neste estudo. Foram selecionadas duas bases públicas de imagens de estacionamentos: a base PKLot (DE ALMEIDA et al., 2015) e a base CNRPark-Ext (AMATO et al., 2017). A seguir, cada uma é descrita detalhadamente, justificando sua escolha para o trabalho.

4.1 *PKLot*

Devido à falta de bases públicas robustas para o problema de detecção de vagas de estacionamento, DE ALMEIDA et al. (2015) realizaram uma coleta abrangente de imagens de estacionamentos em duas universidades na cidade de Curitiba, Brasil. A coleta foi realizada com uma câmera dedicada, que capturava uma imagem a cada cinco minutos durante vários dias.

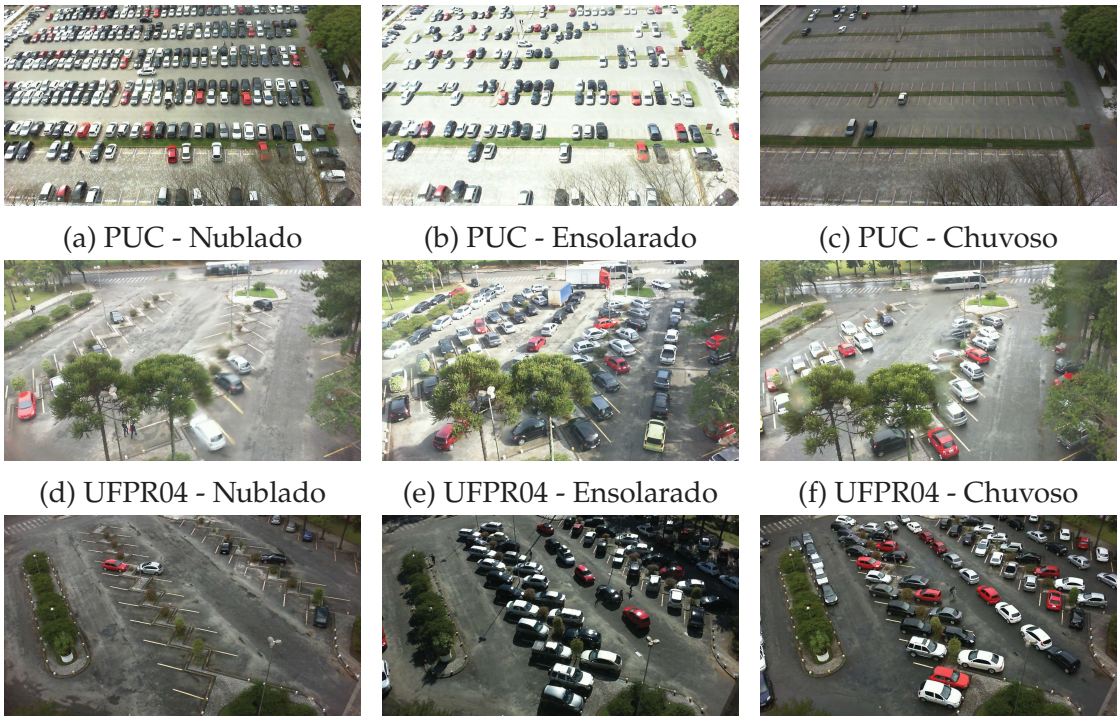
A base PKLot inclui tanto imagens completas dos estacionamentos (12.417 imagens), adequadas para segmentação e detecção de objetos, quanto imagens de vagas já segmentadas (695.851 imagens). As imagens segmentadas permitem cenários específicos de classificação para determinar se uma vaga está livre ou ocupada, aplicável tanto em aprendizado em lote quanto em fluxo de dados.

As Figuras 4.1a e 4.1b mostram exemplos de vagas segmentadas (uma ocupada e outra disponível). Já a Figura 4.2 apresenta exemplos de imagens completas, com variações de clima e ocupação.

A base PKLot pode ser utilizada em fluxos contínuos de dados, pois as ima-

(a) Exemplo de vaga ocupada - PKLot (b) Exemplo de vaga disponível - PKLot

Figura 4.1: Exemplos de vagas segmentadas da base PKLot



(a) PUC - Nublado (b) PUC - Ensolarado (c) PUC - Chuvoso

(d) UFPR04 - Nublado (e) UFPR04 - Ensolarado (f) UFPR04 - Chuvoso

(g) UFPR05 - Nublado (h) UFPR05 - Ensolarado (i) UFPR05 - Chuvoso

Figura 4.2: Amostras da PKLot, seus estacionamentos e climas captados

gens possuem informações temporais, permitindo uma ordenação cronológica. Além disso, para simular mudanças de conceito, como alterações de ângulo de câmera e diferentes estacionamentos, os subconjuntos da base são ordenados, conforme ilustrado na Figura 4.3.

Figura 4.3: Exemplo de imagens dos subconjuntos da base *PKLot* ordenados temporalmente.

A base é subdividida em três subconjuntos: PUC, UFPR04 e UFPR05. Enquanto o subconjunto PUC refere-se a um estacionamento específico, UFPR04 e UFPR05 contêm imagens de um mesmo estacionamento, porém com a câmera posicionada em diferentes locais resultando em ângulos variados e visualizações de vagas distintas devido a possíveis obstruções.

Os autores também separaram as imagens por dia e condições climáticas (ensolarado, chuvoso e nublado). A base inclui imagens capturadas ao longo do dia, desde o amanhecer até o anoitecer. Com essas variações de clima, luminosidade e probabilidade de ocupação de acordo com diferentes fatores (como dias úteis e feriados), a base PKLot é ideal para a implementação de modelos de monitoramento de vagas em tempo real, além de ser aplicável em problemas de *streaming*, onde características das imagens podem variar significativamente.

A Tabela 4.1 mostra a distribuição das imagens segmentadas e o número de dias de coleta por subconjunto:

Estacionamento	**Número de Vagas**	**Número de Dias**
PUC	424.223	38
UFPR04	105.843	30
UFPR05	165.785	33
Total	695.851	101

Tabela 4.1: Quantidade de imagens segmentadas coletadas e dias por estacionamento.

A Tabela 4.2 detalha a distribuição de vagas ocupadas e disponíveis na base PKLot.

Status	Número de Vagas
Disponível	358.071 (51,46%)
Ocupada	337.780 (48,54%)
Total	695.851 (100%)

Tabela 4.2: Distribuição entre as classes disponíveis na base PKLot.

A Tabela 4.3 mostra a distribuição das classes conforme as condições climáticas. .

Clima	Status	Número de Vagas
Ensolarado	Disponível	180.289 (25,91%)
	Ocupada	186.489 (26,80%)
	Subtotal	366.778 (52,71%)
Nublado	Disponível	141.397 (20,32%)
	Ocupada	87.734 (12,61%)
	Subtotal	229.131 (32,93%)
Chuvoso	Disponível	36.409 (5,23%)
	Ocupada	63.533 (9,13%)
	Subtotal	99.942 (14,36%)
Total		695.851 (100%)

Tabela 4.3: Distribuição entre as classes disponíveis na base PKLot de acordo com o clima.

Observando a Tabela 4.3, percebe-se que, em condições chuvosas, cerca de dois terços das vagas estão ocupadas, indicando maior demanda nesses períodos. No caso de clima nublado, a diferença entre vagas ocupadas e disponíveis pode ser explicada pela época do ano em que as imagens foram coletadas.

Abaixo, é realizada uma análise detalhada dos dados por subconjunto para aprofundar o entendimento sobre a base PKLot.

4.1.1 PUC

A coleta de imagens do estacionamento da PUC foi realizada entre 11 de setembro e 20 de novembro de 2012, totalizando 38 dias de captura (com lacunas em alguns dias desse intervalo). Em cada imagem coletada, foram capturadas cerca de 100 vagas de estacionamento. Esse período inclui dias úteis, finais de semana e feriados, como o feriado nacional de 12 de outubro (Dia de Nossa Senhora Aparecida), além de variações climáticas e de iluminação ao longo do dia.

Dos 38 dias de coleta, 9 ocorreram em finais de semana, além do feriado mencionado, resultando em 28 dias úteis de captura. No entanto, como o objetivo desta análise é geral, não serão diferenciados dias úteis de finais de semana ou feriados.

A Figura 4.4 mostra a distribuição das classes de acordo com o clima no momento da captura. Observa-se que a demanda pelo estacionamento é significativamente maior em dias de chuva, enquanto em dias nublados a demanda é menor.

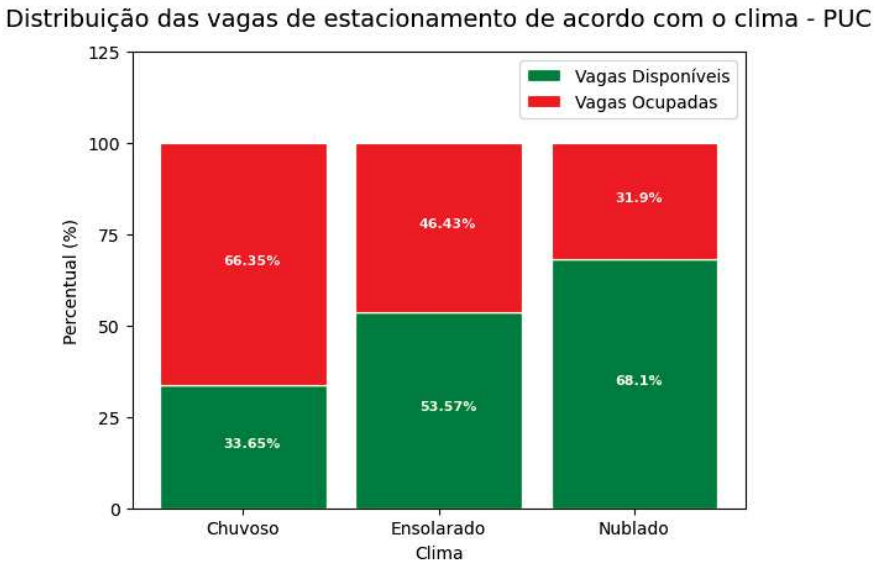


Figura 4.4: Distribuição das vagas de acordo com o clima - PUC.

A discrepância na ocupação em dias nublados pode ser parcialmente explicada pelo fato de que, entre os dias nublados registrados, 4 são finais de semana. Em um total de 69.298 instâncias coletadas nesses dias, apenas 0,91% das vagas estavam ocupadas.

Ao analisar a ocupação diária, observa-se que em 8 dias menos de 10% das vagas estavam ocupadas, enquanto em 11 dias a ocupação foi inferior a 50%. Em dias úteis, com período letivo em andamento, a ocupação ultrapassa 50% da capacidade do estacionamento, com 13 dias registrando ocupação acima de 70%.

Além disso, em 3 dias o número de vagas segmentadas foi inferior a 200, possivelmente devido a problemas técnicos como falha na bateria da câmera. A média diária de vagas capturadas nesse subconjunto é de 11.163, e a baixa quantidade de instâncias nesses dias pode impactar a detecção de mudanças de

contexto uma vez que nesses dias todas as vagas pertenciam a uma única classe (ocupada).

4.1.2 UFPR04

No subconjunto UFPR04, a câmera foi posicionada no quarto andar de um prédio da UFPR, permitindo a segmentação de 28 vagas individuais por imagem. A coleta ocorreu entre 7 de dezembro de 2012 e 29 de janeiro de 2013, um período com potencial menor demanda por vagas devido às férias letivas e às festividades de fim de ano. Esse intervalo inclui dias como 25 de dezembro (Natal) e datas adjacentes. Ao todo, as imagens foram coletadas em 30 dias.

Entre esses 30 dias, 8 são finais de semana. Além disso, o período de 22 a 29 de dezembro também está incluído na base, somando mais 6 dias (excluindo finais de semana) em que uma menor demanda de vagas era esperada. Portanto, a coleta foi realizada em 16 dias úteis nesse subconjunto, resultando em uma predominância de vagas disponíveis, como ilustrado na Figura 4.5.

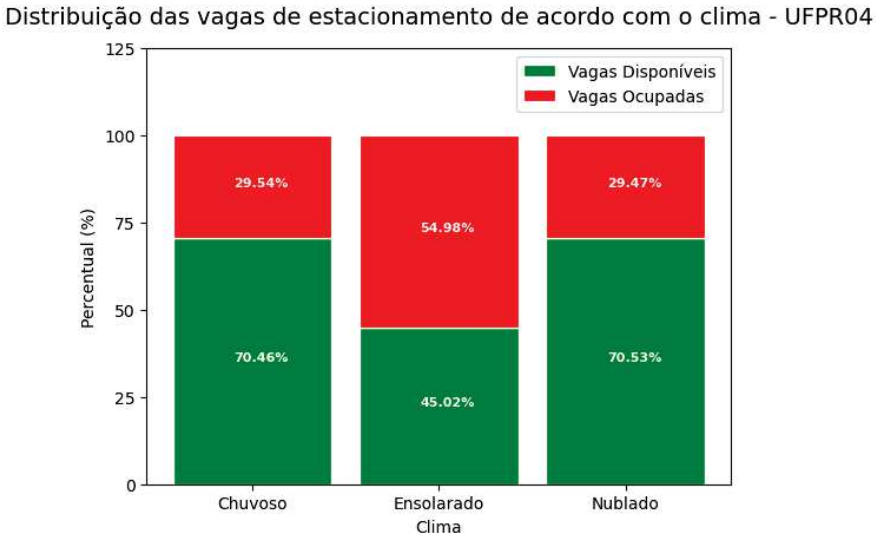


Figura 4.5: Distribuição das vagas de acordo com o clima - UFPR04.

Do total de 105.843 vagas segmentadas, 56,42% são classificadas como disponíveis. Em 12 dias, a taxa de vacância superou 90%, e em 5 desses dias houve vacância total. Diferente do subconjunto PUC, o UFPR04 não apresenta dias com uma quantidade atipicamente baixa de imagens coletadas, com uma média de 3.528 imagens por dia.

A variação na demanda de vagas conforme o clima neste subconjunto pode ser explicada pelo fato de que, dos 14 dias classificados como finais de semana ou feriados, 5 apresentaram períodos de chuva e 8 registraram clima nublado em algum momento.

4.1.3 UFPR05

O subconjunto UFPR05 difere do UFPR04 em relação à angulação da câmera. Embora ambos capturem imagens do mesmo estacionamento, a câmera no UFPR05 foi posicionada no quinto andar, permitindo a segmentação de 40 vagas por imagem. As imagens foram coletadas entre 22 de fevereiro e 16 de abril de 2013, em um total de 33 dias.

Durante esse período, aparentemente não houve coleta em feriados. Dos 33 dias, 10 correspondem a finais de semana, nos quais, como observado anteriormente a taxa de vacância tende a ser maior.

A distribuição de classes no subconjunto UFPR05 conforme o clima está representada na Figura 4.6. Observa-se uma maior demanda por vagas, independentemente das condições climáticas, possivelmente devido ao período letivo durante o qual as imagens foram coletadas.

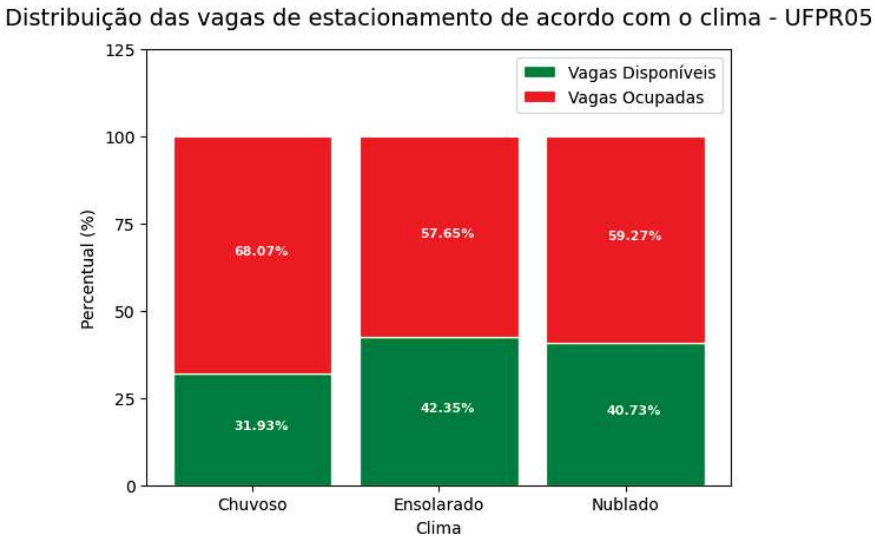


Figura 4.6: Distribuição das vagas de acordo com o clima - UFPR05.

Na distribuição diária de imagens, houve apenas um dia (8 de abril) com coleta reduzida, registrando 160 imagens, enquanto a média diária do subconjunto é de 5.023 imagens. Em relação à vacância, 4 dias apresentaram taxa superior a

90% (com vacância total em 3 dias e 99,97% em outro). Por outro lado, 21 dias tiveram ocupação acima de 70%, explicada pelo período letivo durante a coleta.

4.2 *CNRPark-Ext*

O CNRPark-Ext, proposto por AMATO et al. (2017), é uma extensão da base de dados CNRPark (AMATO et al., 2016) e também é focado no monitoramento de vagas de estacionamento via câmeras. A base de dados foi coletada no estacionamento do Câmpus da CNR (National Research Council) em Pisa, Itália, e apresenta 144.965 amostras coletadas a partir de nove câmeras, superando a base CNRPark original, que contava com 12.584 amostras de apenas duas câmeras. A coleta ocorreu entre novembro de 2015 e fevereiro de 2016, em 23 dias distintos, com todas as câmeras capturando imagens simultaneamente.

Segundo os autores, a CNRPark-Ext oferece maiores desafios para a classificação de vagas de estacionamento, devido à presença de diversas oclusões, causadas por elementos como árvores, postes e sombras de veículos próximos às vagas. Esse nível de complexidade é também influenciado pela altura relativamente baixa das câmeras em comparação com a PKLot, aumentando a probabilidade de obstruções.

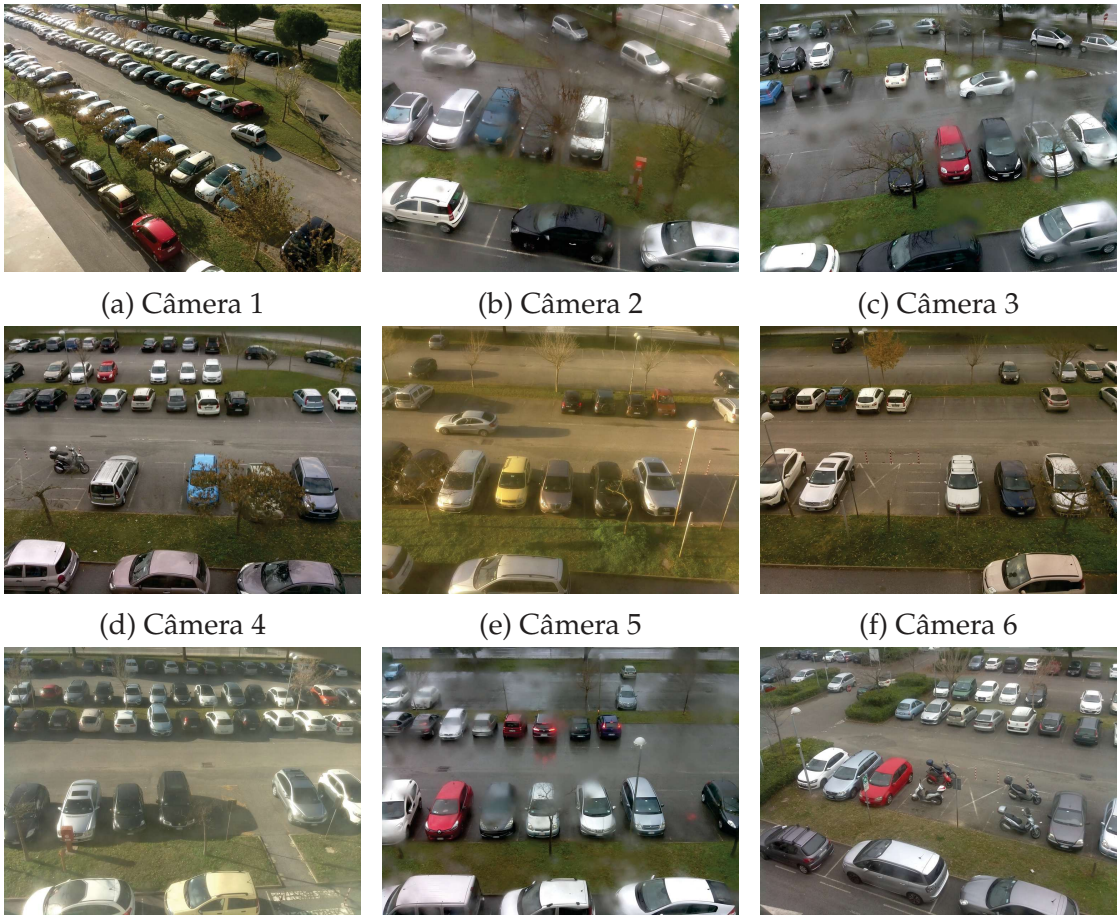
Outro fator que contribui para a complexidade da base é a anotação por *bounding boxes* (sem tratamento adicional, como rotação das imagens) e a variação no tamanho das imagens, que depende da proximidade da vaga em relação à câmera.

A Figura 4.7 mostra exemplos de cada classe do problema representado no *dataset*, enquanto a Figura 4.8 apresenta exemplos de imagens sem segmentação, destacando as diferenças entre PKLot e CNRPark-Ext em relação a ângulos, posições e obstruções das vagas. A diversidade de ângulos, posições e distâncias entre as nove câmeras usadas é evidente.

(a) Exemplo de vaga ocupada - câmera 7.

(b) Exemplo de vaga disponível - câmera 1.

Figura 4.7: Amostras da *CNRPark-Ext*



(a) Câmera 1 (b) Câmera 2 (c) Câmera 3

(d) Câmera 4 (e) Câmera 5 (f) Câmera 6

(g) Câmera 7 (h) Câmera 8 (i) Câmera 9

Figura 4.8: Amostras da *CNRPark-Ext*, suas câmeras e diferentes climas capturados.

Observando a Figura 4.8, podemos observar a ampla diversidade de ângulos, posições e distância adquirida pela base ao longo das nove câmeras utilizadas para a captação das imagens.

Como a coleta foi simultânea entre as câmeras, a Tabela 4.4 apresenta a distribuição de amostras disponíveis em cada um dos nove subconjuntos, mostrando uma discrepância entre o número de amostras por câmera, atribuída às posições e ângulos específicos de instalação.

Câmera	Vagas Disponíveis	Vagas Ocupadas	Total de Vagas
camera1	6.407 (4,42%)	9.308 (6,42%)	15.715 (10,84%)
camera2	1.454 (1,00%)	2.641 (1,82%)	4.095 (2,82%)
camera3	4.101 (2,83%)	5.370 (3,70%)	9.471 (6,53%)
camera4	7.219 (4,98%)	9.357 (6,45%)	16.576 (11,43%)
camera5	9.582 (6,61%)	11.256 (7,76%)	20.838 (14,37%)
camera6	9.462 (6,53%)	10.646 (7,34%)	20.108 (13,87%)
camera7	10.595 (7,31%)	10.519 (7,26%)	21.114 (14,56%)
camera8	11.237 (7,75%)	12.847 (8,86%)	24.084 (16,61%)
camera9	5.601 (3,86%)	7.363 (5,08%)	12.964 (8,94%)
Total	65.658 (45,29%)	79.307 (54,71%)	144.965 (100,00%)

Tabela 4.4: Distribuição de amostras entre as câmeras da base *CNRPark-Ext*.

A Tabela 4.5 resume a distribuição das amostras conforme as condições climáticas. Ao contrário da PKLot, a CNRPark-Ext registra uma menor taxa de ocupação nos dias chuvosos, enquanto a taxa de ocupação aumenta em dias nublados e ensolarados.

Clima	Status	Número de Vagas
Ensolarado	Disponível	25.665 (17,70%)
	Ocupada	37.513 (25,88%)
	Subtotal	63.178 (43,58%)
Nublado	Disponível	21.067 (14,53%)
	Ocupada	23.176 (15,99%)
	Subtotal	44.243 (30,52%)
Chuvoso	Disponível	18.926 (13,06%)
	Ocupada	18.618 (12,84%)
	Subtotal	37.544 (25,90%)
Total		144.965 (100,00%)

Tabela 4.5: Distribuição entre as classes disponíveis na base *CNRPark-Ext* de acordo com o clima.

Ambas as bases discutidas, PKLot e CNRPark-Ext, apresentam grande diversidade de informações e registram o momento de captura de cada imagem, o que as torna adequadas para utilização em cenários de fluxos contínuos de dados.

5

Estado da arte

Com o crescente interesse em fluxos contínuos de dados, o estudo sobre o uso de redes neurais nesse contexto também evoluiu. No entanto, a pesquisa sobre técnicas de aumento de dados (*data augmentation*) para cenários de *streaming* ainda está em estágio inicial, com poucos trabalhos publicados.

Esta seção apresenta alguns estudos relevantes sobre o uso de aumento de dados em fluxos contínuos, com foco especial na aplicação de Redes Generativas Adversárias (GANs) nesse cenário.

No trabalho de GUZY; WOźNIAK; KRAWCZYK (2021), são propostos métodos para avaliar a performance de redes GAN em cenários de fluxo de dados com mudanças de conceito. Os autores utilizaram três datasets: CelebA, Cifar10 e Fashion MNIST.

Como essas bases não possuem mudanças de conceito explícitas, foram feitas manipulações para simular mudanças, dividindo as classes e ajustando os rótulos. A CelebA por exemplo, foi dividida em rostos masculinos e femininos, simulando uma transição de conceito. Para a base *Cifar10*, a base é dividida de forma que contenha diferente tipos de máquinas e/ou animais em cada divisão. Já para a *Fashion MNIST*, os autores realizam a divisão de modo que as mesmas contenham diferentes tipos de peças de roupa em cada subdivisão.

Deste modo, as classes alvo das bases também são alteradas, de modo que para a base *CelebA*, é utilizado como rótulo a cor do cabelo da pessoa (se loiro ou moreno), enquanto para a *Cifar10* o *target* é indicar se a imagem se trata de um animal ou máquina, e, por fim, para a *Fashion MNIST* o objetivo é indicar se a roupa da imagem possui ou não mangas.

Para avaliar as GANs, foram propostas duas métricas: IGI (*Incremental Generative Inference*) e R^2D^2 (*Relevant Regions for Dynamic Data*). A IGI monitora a convergência da rede GAN ao longo do tempo com dois classificadores auxiliares (um para dados reais e outro para dados gerados), avaliando o progresso com métricas como AUC (área sob a curva). Já a R^2D^2 utiliza mapas de calor para identificar as regiões mais relevantes nas imagens, permitindo observar como a rede se adapta às mudanças.

Os autores também formularam três perguntas de pesquisa para avaliar a flexibilidade das métricas, a eficácia da IGI no monitoramento de performance e o potencial explicativo das métricas IGI e R^2D^2 em cenários de mudanças de conceito. As redes DCGAN e WGAN, junto com classificadores ResNet18, foram testadas, demonstrando a adaptabilidade e eficácia das métricas. Os resultados estão apresentados na Tabela 5.1 e na Figura 5.1.

Embora o trabalho de GUZY; WOźNIAK; KRAWCZYK tenha focado no monitoramento de desempenho das GANs, ele não explora o uso das redes GAN como um método de aumento de dados para melhorar a acurácia e o tempo de resposta em mudanças de conceito.

	DCGAN		WGAN-WC	
	IGI	time	IGI	time
CelebA	0.8612	0.1999	0.8213	0.5034
Cifar10	0.8974	0.2220	0.8576	0.51656
FMNIST	0.9716	0.2308	0.9310	0.52345

Tabela 5.1: IGI médios e tempos de processamento para as redes DCGAN e WGAN. Fonte: (GUZY; WOźNIAK; KRAWCZYK, 2021).

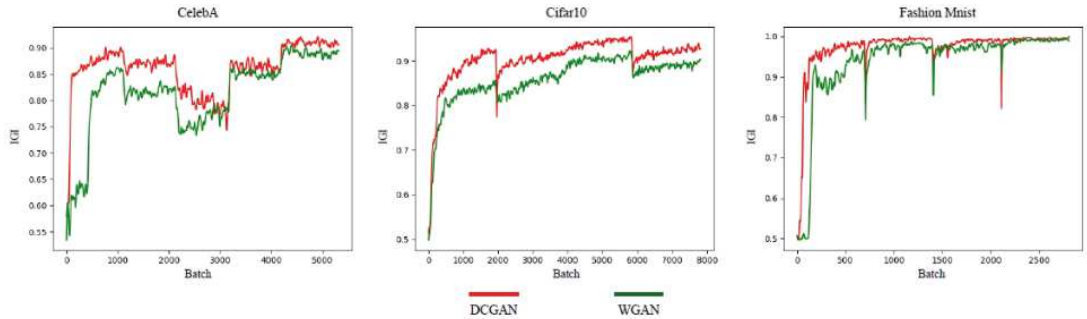


Figura 5.1: Resultados em forma gráfica da IGI para os *datasets* testados. Fonte: (GUZY; WOźNIAK; KRAWCZYK, 2021).

Já em relação à terceira questão formulada, é observada, além da IGI já mencionada, a R^2D^2, como pode ser visualizado na Figura 5.2, demonstrando a mudança da posição das posições relevantes para a rede ao longo do tempo com a mudança de conceito, tanto para as imagens reais quanto para as imagens geradas.

Figura 5.2: R^2D^2 gerada utilizando a base *CelebA*. Fonte: (GUZY; WOźNIAK; KRAWCZYK, 2021).

De modo a garantir a reprodutibilidade do experimento, os autores disponibilizam os artefatos de seu trabalho em um repositório da internet, o qual foi devidamente baixado e reproduzido afim de se verificar a consistência dos resultados apresentados, sendo observada a reprodutibilidade e os resultados obtidos.

Outro estudo relevante é o de MALIALIS et al. (2022), que aborda o problema de rotulação limitada e desbalanceamento de dados em fluxo de dados. Inspirados nos desafios de obter dados rotulados em tempo real, os autores propuseram o método *Augmented Queues* (Filas Aumentadas), que utiliza aprendizado ativo para adicionar instâncias de dados aumentadas dinamicamente, conforme a incerteza nas predições. Técnicas geométricas de aumento de dados (como rotação e alteração de brilho) foram aplicadas a imagens, e métodos como recortes e distorção temporal foram usados para séries temporais.

A implementação do método baseia-se no *ActiQ* (MALIALIS; PANAYIOTOU; POLYCARPOU, 2020), adaptado para lidar com as limitações de rotulação e memória.

Na Figura 5.3 pode ser visualizada a visão geral do método proposto, onde as filas representadas por q são as filas com dados originais e as filas representadas pela letra a são as filas com os dados provenientes das técnicas aplicadas de aumento de dados. Tais filas alimentam o treinamento do classificador, que por

sua vez realiza as predições dos dados do fluxo e, de acordo com a estratégia aplicada é decidido se o rótulo da instância corrente deve ser solicitado ou não para adaptação do modelo.

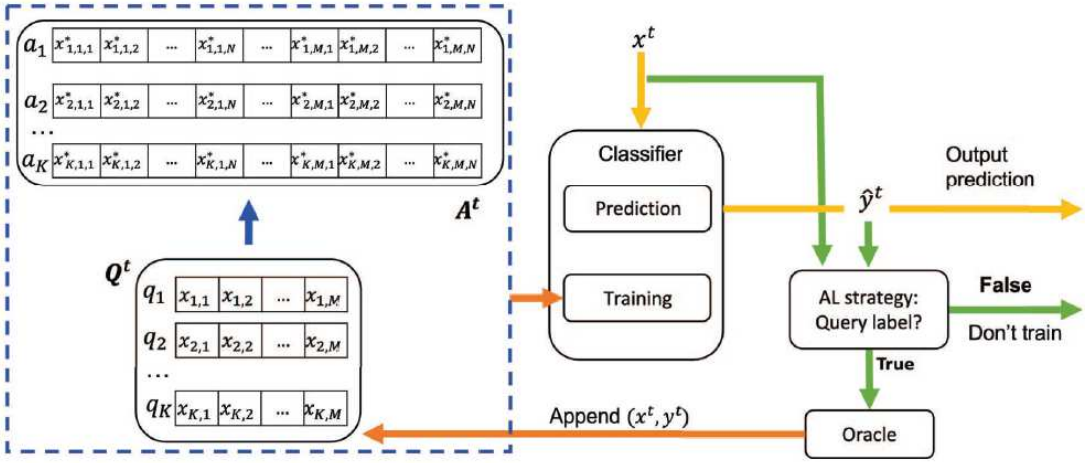


Figura 5.3: Visão geral da arquitetura do modelo proposto. Fonte: (MALIALIS et al., 2022)

Para os experimentos, são utilizados os *datasets* MNIST, *Two Patterns* (séries temporais simuladas, com quatro classes e 5.000 instâncias) e *uWave Geature Library Z* (base de dados de séries temporais com um conjunto de oito simples gestos gerados por acelerômetros utilizando o controle de um *Nintendo Wii*). Para os casos onde a base apresenta-se balanceada, são simulados casos de desbalanceamento. Além disso, são utilizadas uma rede neural totalmente conectada simples e uma rede VGG, sem pré-treino (testadas diversas variações da VGG) e diferentes percentuais de dados rotulados disponíveis (50%, 25%, 10% e 1%).

A métrica utilizada para avaliação é a média geométrica, dada por:

$$\textit{Média Geométrica} = \sqrt[K]{\prod_{c=1}^{K} R_c}$$

Onde K é o número de classes, e $R_c = N_{cc}/N_c$ é a métrica de revocação da classe c onde N_{cc} é o número de instâncias corretamente preditas para a classe c e N_c é o número total de amostras da classe. A métrica de média geométrica foi selecionada por não ser, segundo os autores, sensível ao problema de desbalanceamento de dados.

Avaliando-se os papeis exercidos pelo percentual de amostras rotuladas disponíveis e a profundidade da rede aplicada, os autores observam que de acordo

com a queda do percentual de rótulos disponíveis, a performance cai, além das filas serem alimentadas em uma menor velocidade e o treino se torna menos frequente; com o aumento do desbalanceamento também é observada a queda de performance e é verificado também o papel crucial que o tipo de modelo aplicado exerce, onde a VGG superou a rede totalmente conectada, devido sua profundidade, porém com grande aumento da profundidade da VGG também é percebida uma queda na performance do modelo.

Avaliando-se o papel de memória disponível, foi observado que com o aumento de memória, a performance melhora significativamente e que, em cenários de extremo desbalanceamento dos dados tal papel se torna ainda mais importante. Ainda assim, como geralmente a quantidade de memória é uma restrição em cenários de fluxos de dados, tal observação ainda pode ser entendida como uma limitação.

Assim, onde o papel do aumento de dados avaliado pelos autores é notado que tais técnicas exercem uma melhora drástica na melhora de performance do modelo, sendo tal fenômeno atribuído ao aumento do conjunto de treinamento do modelo e o método proposto usa menos espaço em memória, uma vez que o aumento de dados acontece dinamicamente apenas nos momentos de treinamento do modelo.

MALIALIS et al. demonstraram que o aumento de dados melhora significativamente a performance, mas o estudo se limita a técnicas geométricas de aumento e não considera o uso de dados sintéticos, como redes GAN, para aumento de dados.

Outro trabalho interessante é o de POUYANFAR et al. (2018), que explora o desbalanceamento de dados em imagens de câmeras de monitoramento em tempo real. Os autores desenvolveram um modelo de classificação baseado em uma CNN, aplicando aumento de dados em tempo real conforme a métrica de F1-score por classe. A taxa de aumento foi inversamente proporcional ao F1-score, visando melhorar a acurácia de classes minoritárias sem sacrificar o desempenho nas classes majoritárias.

Durante tal trabalho, os pesquisadores discorrem sobre a dificuldade de se coletar imagens de câmeras disponibilizadas de forma pública em tempo real, devido aos diferentes equipamentos e protocolos utilizados para tal. Mesmo assim, realizam um trabalho de coleta de dados e disponibilizam um novo *dataset* com tais imagens para o público em geral.

Para tal, os autores propõem um modelo de classificação baseado em uma

CNN em conjunto com a aplicação de técnicas de *data augmentation* em tempo real, sendo definida a taxa de aumento de dados por classe em relação à métrica de *F1-score* de cada classe, sendo o número de imagens geradas por tal técnica inversamente proporcional ao resultado obtido por classe a cada iteração de treinamento da rede convolucional. A métrica de *F1-score* foi selecionada por tratar de forma mais justa todas as classes existentes no problema, mesmo com os dados estando desbalanceados.

O modo com que são calculadas o número de imagens a serem geradas por classe é dada pela seguinte fórmula:

$$\textit{Update Sample Size}(F1_i, c_j) = \frac{1-f1_{i,j}}{\sum_{c_k \in C}(1-f1_{i,k})} \times N$$

Onde $f1_{i,j}$ diz respeito ao F1-score obtido para a classe j em questão na iteração i do modelo, C é o conjunto de classes do problema e c_k são as classes pertencentes à tal conjunto. Para finalizar, N é o tamanho definido do conjunto de treinamento da rede.

A Figura 5.4 apresenta a arquitetura do modelo proposto, que combina uma *InceptionV3* pré-treinada com um módulo de amostragem dinâmica e aumento de dados. Experimentos mostraram que o modelo alcançou alta precisão para classes minoritárias, sem perda significativa para as majoritárias, demonstrando a eficácia do método em cenários de desbalanceamento.

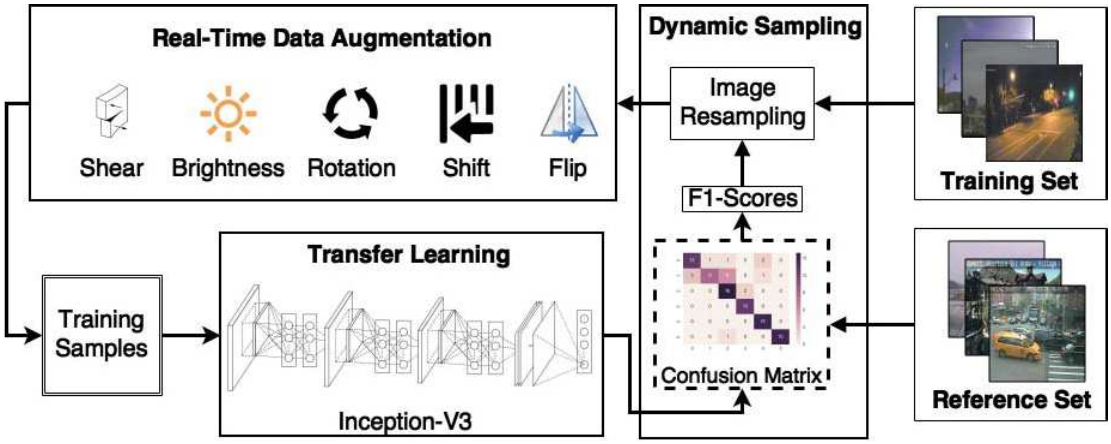


Figura 5.4: Modelo proposto por POUYANFAR et al. (2018)

A Tabela 5.2 resume os métodos e resultados dos trabalhos discutidos, destacando os pontos fortes e limitações de cada abordagem no uso de técnicas de aumento de dados em fluxo de dados.

Trabalho	**Características**
GUZY et al.	- Técnica utilizada: Redes GAN - Bases de Dados utilizadas: CelebA; Cifar10 e Fashion MNIST. - Conclusões: Sugestão e validação de métricas para avaliação de desempenho de redes GAN em cenários de *streaming data*. Não avalia o efeito do uso de Redes Generativas como método auxiliar em um problema de classificação de fluxo de dados.
MALIALIS et al.	- Técnica utilizada: Técnicas Geométricas - Bases de Dados utilizadas: MNIST. além das bases de séries temporais Two Patterns e uWave Geature Library Z. - Conclusões: Validação do aumento de dados para melhorar o desempenho em cenários desbalanceados. Não utiliza Redes Generativas em seu trabalho.
POUYANFAR et al.	- Técnica utilizada: Técnicas Geométricas - Bases de Dados utilizadas: Imagens de câmeras de monitoramento e vigilância públicas. - Conclusões: Aumento de dados proporcional ao F1-score melhora a acurácia para classes minoritárias sem sacrificar majoritárias. Não utiliza Redes Generativas em seu trabalho.

Tabela 5.2: Comparação dos trabalhos do estado da arte.

Em resumo, embora as redes GAN já tenham sido aplicadas em cenários de fluxos de dados, seu uso como método de aumento de dados para melhorar a acurácia e resposta a mudanças de conceito ainda não foi totalmente explorado. Por outro lado, técnicas geométricas de aumento de dados já demonstraram benefícios, apesar de haver poucos trabalhos sobre o tema.

6

Metodologia

Em cenários de mineração de fluxos contínuos de dados, os modelos de aprendizado de máquina tendem a sofrer degradação de desempenho ao longo do tempo devido a mudanças de conceito. Para identificar quando essa queda de performance ocorre e atualizar o modelo preditivo de forma apropriada, é recomendada a utilização de um detector de mudanças de conceito. Além disso, para melhorar o desempenho do modelo após a atualização, podem ser aplicadas técnicas de aumento de dados, fornecendo mais instâncias rotuladas para o processo de treinamento/atualização.

Embora redes GAN já sejam usadas como técnica de aumento de dados em alguns contextos, seu uso em fluxos de imagens ainda é limitado devido às restrições de memória e processamento típicas de cenários de fluxos de dados.

As bases de dados PKLot e CNRPark-Ext são amplamente utilizadas em pesquisas de aprendizado de máquina e inteligência artificial. Ambas as bases apresentam alta diversidade entre as amostras, bem como atributos temporais (data e hora de coleta), o que permite a realização de estudos em cenários de fluxos contínuos de imagens.

Este Capítulo está estruturado de forma a descrever os métodos utilizados para o desenvolvimento do trabalho, apresentando desde o ambiente onde os experimentos foram realizados, o tratamento das bases de dados, passando pelas arquiteturas das redes CNN e DCGAN utilizadas, suas respectivas estratégias de treinamento, os métodos de detecção de mudanças de conceito e métricas de avaliação.

6.1 Ambiente de Execução

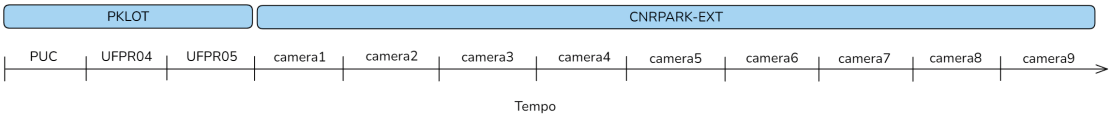
Todos os experimentos realizados ao longo do trabalho foram desenvolvidos em um ambiente com Sistema Operacional Ubuntu 20.04, contando com um processador Intel Xeon W-1290P com 3.7GHz de clock, com 64GB de memória RAM. O ambiente também contava com uma GPU Nvidia GeForce RTX 2080 Ti, com 11GB de memória disponível.

6.2 Fluxo de Imagens

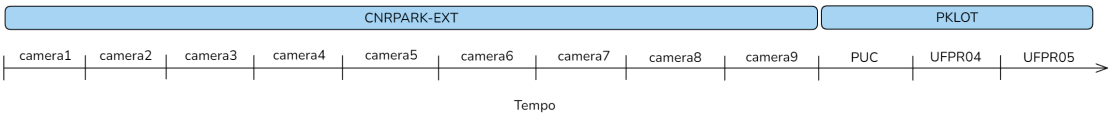
Nos experimentos, foram utilizadas duas bases de dados públicas que contêm imagens de estacionamento: PKLot (DE ALMEIDA et al., 2015) e CNRPark-Ext (AMATO et al., 2017).

Como essas bases possuem informações temporais relacionadas ao momento da captura das imagens, foi possível ordená-las no formato de um fluxo contínuo de dados. As bases foram ordenadas de duas formas distintas, sendo um fluxo ordenado iniciando-se na base PKLot e posteriormente passando a processar as imagens da base CNRPark-Ext, enquanto a segunda realiza o caminho inverso, partindo da CNRPark-Ext e finalizando com a base PKLot.

Esses dois fluxos de imagens testados podem ser melhor entendidos com a Figura 6.1.



(a) Sequência do Fluxo de Imagens #01 - *CNRPark-Ext* para *PKLot*.



(b) Sequência do Fluxo de Imagens #02 - *PKLot* para *CNRPark-Ext*.

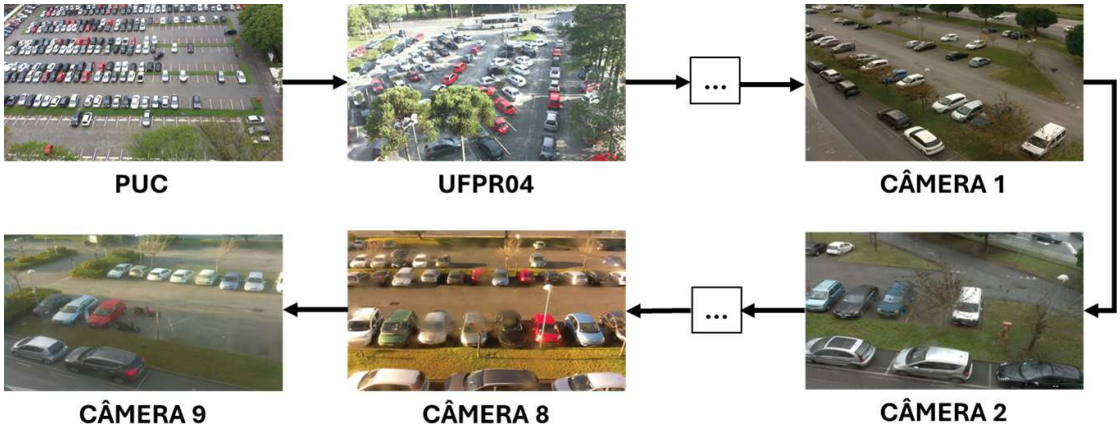
Figura 6.1: Fluxos de imagens utilizados durante os experimentos.

Assim, já se possui algumas possíveis mudanças de conceito devido à mudança de estacionamento (entre os subconjuntos de imagens contidos na base PKLot) e ao ângulo da câmera (existente entre as mudanças de subconjuntos das duas bases de imagens utilizadas). Além disso, na transição entre as duas bases,

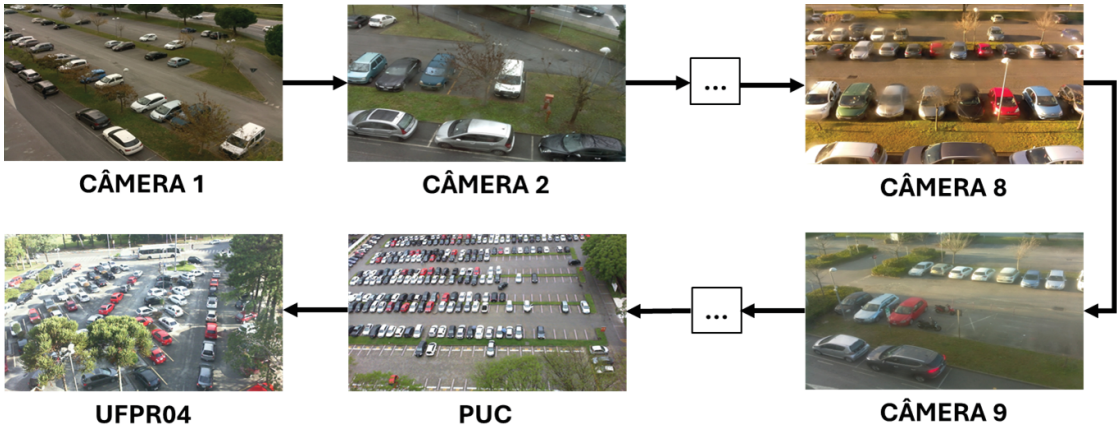
há uma mudança da própria câmera, o que interfere na qualidade e resolução das imagens disponíveis para o experimento.

A escolha de trabalhar com dois fluxos de dados diferentes visou generalizar o método, evitando a avaliação em um único cenário com um fluxo de imagens específico, variando assim a ordem das câmeras e consequentemente influenciando no processo de detecção de mudanças de conceito.

Dessa forma, os fluxos de imagens são representados na Figura 6.2, ilustrando mudanças de conceito entre as câmeras.



(a) Exemplos de Mudanças de ângulo/câmera no fluxo de imagens *CNRPark-Ext* → *PKLot*.



(b) Exemplos de Mudanças de ângulo/câmera no fluxo de imagens *PKLot* → *CNRPark-Ext*.

Figura 6.2: Exemplos de mudanças de conceito por ângulo/câmera presentes nos fluxos de imagens utilizados.

6.3 Pré-processamento das imagens

As imagens já são disponibilizadas de forma segmentada, sendo assim, o pré-processamento das amostras ao longo do fluxo de dados se dá pelo redimensionamento das mesmas de forma com que fiquem com o tamanho de $64x64$ pixels.

Após o redimensionamento, as imagens passam pela respectiva função de pré-processamento da CNN utilizada no experimento, com a função *preprocess_input* disponível no pacote *TensorFlow* do Python. Esse processo assegura que as imagens sejam normalizadas de acordo com os requisitos da rede utilizada.

Para o treinamento dos modelos generativos, as imagens redimensionadas são normalizadas para valores entre -1 e 1. As imagens sintetizadas pelas redes generativas passam pelo mesmo pré-processamento das imagens originais antes de serem utilizadas no modelo preditivo, sendo geradas no tamanho final de $64x64$ pixels.

6.4 Arquitetura das CNNs

Neste trabalho, a generalização da solução foi buscada de forma independente do classificador utilizado, para isso, foram utilizadas três redes neurais convolucionais (CNNs) com arquiteturas distintas:

- **MobileNetV3:** Escolhida por ser uma rede de menor tamanho, com menos parâmetros, oferecendo maior facilidade na aplicação em câmeras inteligentes.
- **ResNet50:** Selecionada devido ao seu número relativamente maior de parâmetros e ao uso inovador do conceito de skip connections. Essas conexões residuais ajudam a solucionar o problema de desaparecimento do gradiente em redes neurais profundas.
- **VGG16:** Escolhida por ter um maior número de parâmetros em comparação com as outras redes, mas ainda ser simples, com camadas convolucionais empilhadas seguidas por camadas de pooling.

Em todos os experimentos, as redes CNN foram carregadas com seus pesos pré-treinados na base ImageNet. Após isso, a última camada convolucional foi descongelada para permitir o treinamento, e foram adicionadas três camadas densas com 256, 128 e 64 neurônios, seguidas por uma camada de classificação

softmax no final. A estrutura completa das redes pode ser visualizada na Figura 6.3.

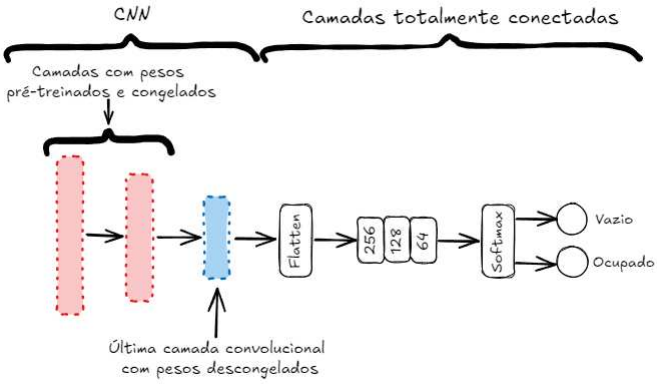


Figura 6.3: Estrutura das redes convolucionais utilizadas nos experimentos.

6.5 Arquitetura das DCGANs

Para a geração de imagens sintéticas visando o aumento de dados para as atualizações do modelo classificador ao longo do fluxo de imagens, foram utilizadas duas redes DCGANs (Deep Convolutional Generative Adversarial Networks). Cada rede foi responsável por gerar imagens de uma classe específica do problema, ou seja, uma para a classe ocupada e outra para a classe vazia.

Essa abordagem se justifica pelo fato de que, ao utilizar a arquitetura DCGAN, não é possível inferir diretamente a classe da imagem gerada. Isso torna inviável o uso de apenas uma única rede sendo treinada com amostras de ambas as classes, uma vez que a geração de imagens para ambas as classes de forma simultânea não seria possível com esse tipo de arquitetura.

Embora existam outras alternativas de redes generativas, como as *Conditional GANs* (cGANs) e *Variational Autoencoders* (VAE), que permitirão inferir a classe da amostra sintética gerada, estas não estavam previstas no escopo inicial do trabalho, ficando como sugestões para trabalhos futuros.

A arquitetura das DCGANs utilizadas foi baseada na estrutura proposta por WU; CHEN; MENG, (2020), com uma modificação que se mostrou mais eficaz em testes prévios para o problema abordado. A modificação consistiu na adição de uma camada convolucional extra no discriminador da rede, com 128 filtros e um *kernel size* de $4x4$, o que proporcionou melhores resultados.

A arquitetura das redes DCGANs utilizadas é ilustrada na Figura 6.4.

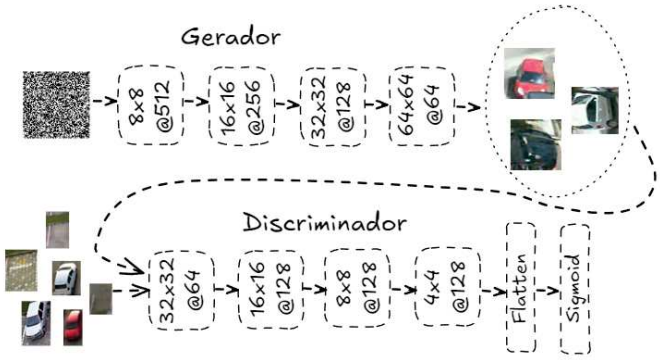


Figura 6.4: Estrutura da rede DCGAN utilizada nos experimentos.

6.6 Estratégia de Treinamento

Para a execução dos experimentos, foram adotadas as seguintes estratégias de treinamento para as redes CNN e DCGAN:

- Treinamento inicial dos modelos:
  - O primeiro dia do fluxo de imagens foi utilizado para o treino inicial dos modelos. As imagens foram balanceadas de acordo com suas classes, sendo divididas em dois conjuntos:
    - Treinamento: 70% das imagens
    - Validação: 30% das imagens
  - O treinamento da CNN foi realizado com as seguintes configurações:
    - Épocas: 20
    - Otimizador: Adam
    - Taxa de aprendizado: 0.001
    - Tamanho do lote (*batch_size*): 32 amostras
  - Para as DCGANs, o treinamento inicial foi realizado com as mesmas imagens, com as amostras de cada classe separadas para sua respectiva DCGAN. O treinamento das DCGANs foi configurado da seguinte forma:
    - Épocas: 200
    - Otimizador: Adam
    - Taxa de aprendizado: 0.002
    - Beta_1: 0.5 (para o Adam)
    - Função de perda: Entropia cruzada (*binary cross-entropy*)

- Retreinamento das redes:

- O retreinamento das CNNs e DCGANs foi realizado utilizando os mesmos parâmetros do treinamento inicial. No entanto, as amostras para o processo de retreinamento foram fornecidas pela fila utilizada pelo detector de mudanças de conceito no momento da detecção de uma mudança.
- Quando o detector de mudanças de conceito identificava uma mudança, aguardava-se o final do dia do fluxo que estava sendo processado. Após o término desse dia, as redes eram atualizadas:
  * As redes generativas (DCGANs) eram atualizadas antes, permitindo a síntese de novas imagens para a atualização da rede classificadora (CNN).
  * 1000 imagens de cada classe eram sintetizadas pelas DCGANs, número baseado nos resultados do trabalho de HOCHULI et al., (2022).

Esse processo de atualização contínua e geração de imagens sintéticas visa melhorar a acurácia do modelo e minimizar o impacto das mudanças de conceito durante o fluxo contínuo de dados.

6.7 Métodos de Detecção de Mudanças de Conceito

Durante os experimentos, o algoritmo ADWIN foi selecionado como detector de mudanças de conceito devido a suas características, comparado com dois outros métodos: DDM (GAMA et al., 2004) e KSWIN (RAAB; HEUSINGER; SCHLEIF, 2020). A escolha do ADWIN se baseou em suas vantagens sobre os outros dois algoritmos, como descrito abaixo:

- ADWIN utiliza janelas dinâmicas de dados, com o tamanho ajustado com base na estabilidade dos dados, enquanto o DDM e o KSWIN utilizam janelas de tamanho fixo, o que exige a definição prévia desse tamanho.
- Janelas de tamanho fixo podem ter maior dificuldade em detectar mudanças de conceito graduais, enquanto o ADWIN se adapta melhor a esses cenários devido ao seu ajuste dinâmico.
- Algoritmos com janelas fixas podem ser mais sensíveis a ruídos nos dados, resultando em falsas detecções de mudanças de conceito. O ADWIN, por outro lado, minimiza esse problema.

Assim, foram realizados testes afim de comparar o comportamento dos três métodos em função do número de detecções de mudanças de conceito e acuracidade ao longo do fluxo de imagens, utilizando cenários sem atualização

do preditor e com atualização utilizando-se de aumento de dados para auxiliar a atualização dos pesos dos classificadores. Os resultados obtidos estão representados na Tabela 6.1. O classificador utilizado para os testes foi a rede ResNet50 e o cenário utilizado foi o fluxo CNRPark-Ext → PKLot. Cada cenário foi executado 10 vezes com sementes aleatórias distintas.

	DDM		KSWIN		ADWIN	
	Mudanças	Acurácia (%)	Mudanças	Acurácia (%)	Mudanças	Acurácia (%)
Sem atualização	$210,30 \pm 21,66$	$79,37 \pm 3,40$	$82,80 \pm 18,07$	$79,37 \pm 3,40$	$70,30 \pm 6,33$	$79,37 \pm 3,40$
Com Atualização	$145,60 \pm 2,07$	$98,60 \pm 0,41$	$7,00 \pm 2,11$	$95,76 \pm 0,90$	$16,80 \pm 5,53$	$97,04 \pm 0,55$

Tabela 6.1: Resultados obtidos para seleção do detector de mudanças de conceito.

Como pode ser percebido, os algoritmos apresentam alto volume de detecções de mudanças de conceito quando não há atualizações do classificador, enquanto esse valor é drasticamente reduzido, principalmente nos casos dos algoritmos KSWIN e ADWIN. O algoritmo DDM, apesar de apresentar maior valor de acurácia geral quando há atualização dos pesos da rede ResNet50, apresenta um número de detecções de mudanças muito alto, beirando 50% dos dias do fluxo de imagem, o que pode justificar a alta acurácia obtida. Dentre os alforitmos restantes, em termos de acurácia o algoritmo ADWIN destacou-se em relação ao KSWIN, mesmo apresentando pouco mais de o dobro do número de detecções de mudanças de conceito.

Os valores altos dos números de detecções de mudanças de conceito, principalmente nos cenários com detectores que utilizam janela de dados com tamanho fixo refletem a dificuldade em parametrizar um tamanho fidedigno para tal, que devidamente realize o balanceamento entre número de detecções e acurácia. Deste modo, o algoritmo ADWIN foi selecionado para os testes, por não requerer a configuração do tamanho da janela e demonstrar um maior custo-benefício quando leva-se em conta as duas métricas avaliadas.

Quanto à configuração de parâmetros do algoritmo ADWIN, o parâmetro *delta*, que é o nível de significância utilizado para a detecção estatística de mudanças de conceito foi ajustado de seu valor padrão de 0.002 para 0.005, aumentando assim sua "sensibilidade"em relação à possíveis detecções.

6.8 Cenários Avaliados

Afim de buscar uma comparação justa, obtendo conclusões robustas sobre a técnica proposta, foram realizados experimentos em quatro cenários diferentes, sendo eles:

1. **Cenário sem atualização do modelo preditivo:** O modelo preditivo é treinado inicialmente e utilizado para acompanhar o processamento do fluxo de imagens, identificando mudanças de conceito por meio do ADWIN. Contudo, não são realizadas atualizações no classificador nem o treinamento ou uso de redes generativas.

2. **Cenário com atualização do modelo preditivo apenas com dados reais:** O modelo preditivo é atualizado sempre que uma mudança de conceito é detectada. A atualização utiliza exclusivamente os dados reais contidos na janela de detecção fornecida pelo ADWIN no momento da mudança, sem o uso das redes DCGAN.

3. **Cenário com atualização do modelo preditivo com dados reais e aumento de dados por técnicas convencionais:** Além dos dados reais presentes na janela de detecção do ADWIN, são geradas imagens utilizando-se técnicas convencionais de aumento de dados com o uso do pacote *Tensorflow*. Esse procedimento busca o aumento do volume e diversidade das amostras, utilizando as seguintes técnicas em conjunto com fatores aleatórios para tal:

   - Alteração de brilho;
   - Alteração de contraste;
   - Alteração de saturação;
   - Alteração de matiz da imagem;
   - Rotação da imagem;
   - Cortes aleatórios na imagem.

4. **Cenário com atualização do modelo preditivo com dados reais e aumento de dados com o uso de redes generativas:** Além dos dados reais presentes na janela de detecção do ADWIN, são geradas imagens pelas redes DCGAN, que também têm seus pesos atualizados com os dados reais. Esse procedimento aumenta o volume e a diversidade das amostras, fornecendo um conjunto de dados mais robusto para a atualização do modelo preditivo.

Avaliando-se os resultados desses quatro diferentes cenários torna-se possível obter conclusões fundamentadas a ponto de se verificar se a técnica proposta apresenta algum tipo de ganho em comparação com os demais métodos que representam abordagens comuns adotadas para esse tipo de problema.

6.9 Métodos de Avaliação

Os resultados obtidos após a execução dos experimentos foram avaliados sobre duas principais métricas, sendo elas a acurácia média geral obtida para cada cenário e fluxo de dados, isso se devendo ao fato de que o tipo de problema envolvido demanda alta acuracidade, sendo que como as bases de dados utilizada são praticamente balanceadas torna-se uma métrica viável para medir o seu desempenho. Já a segunda métrica avaliada foi o número médio de mudanças de conceito detectadas ao longo do fluxo de imagens, se justificando pelo fato de medir a quantidade de vezes necessárias de retreino para cada cenário.

Para a comparação e a aceitação ou rejeição de hipóteses relacionadas aos resultados, foi utilizado o teste estatístico de Friedman (FRIEDMAN, 1937), com o fim de determinar se há diferença significativa entre os cenários testados. Após esse teste, foi ainda realizado o teste de Nemenyi (CORDER; FOREMAN, 2011) com o objetivo de determinar quais cenários apresentam diferença significativa. Ambos os testes estatísticos foram realizados utilizando um nível de significância $\alpha = 0.05$.

7

Resultados

Conforme os métodos e critérios de avaliação descritos no Capítulo 6, esta seção apresenta o protocolo experimental adotado, os resultados obtidos e sua respectiva análise.

A organização segue a seguinte estrutura: inicialmente, descreve-se o protocolo experimental adotado, seguido pela apresentação e análise dos resultados dos dois fluxos de imagens avaliados (PKLot → CNRPark-Ext e CNRPark-Ext → PKLot). Por fim, compara-se os diferentes cenários testados, conforme descrito na Seção 6.8.

7.1 Protocolo Experimental

Este estudo avalia o impacto do uso de Redes Generativas Adversárias Convolucionais Profundas (DCGANs) em fluxos contínuos de dados baseados em imagens com mudanças de conceito. Além do classificador, os pesos das DCGANs são atualizados sempre que uma mudança de conceito é detectada. Foram utilizadas duas redes DCGAN, cada uma especializada em uma classe, pois uma única rede não permitiria inferir a qual classe pertence a imagem gerada. A Figura 7.1 ilustra o protocolo experimental.

Inicialmente, utilizam-se as amostras de vagas de estacionamento do primeiro dia do fluxo para o treinamento das DCGANs e do modelo preditivo. Em seguida, as imagens dos dias subsequentes são processadas pelo modelo, com a detecção de mudanças de conceito monitorada a cada predição via algoritmo

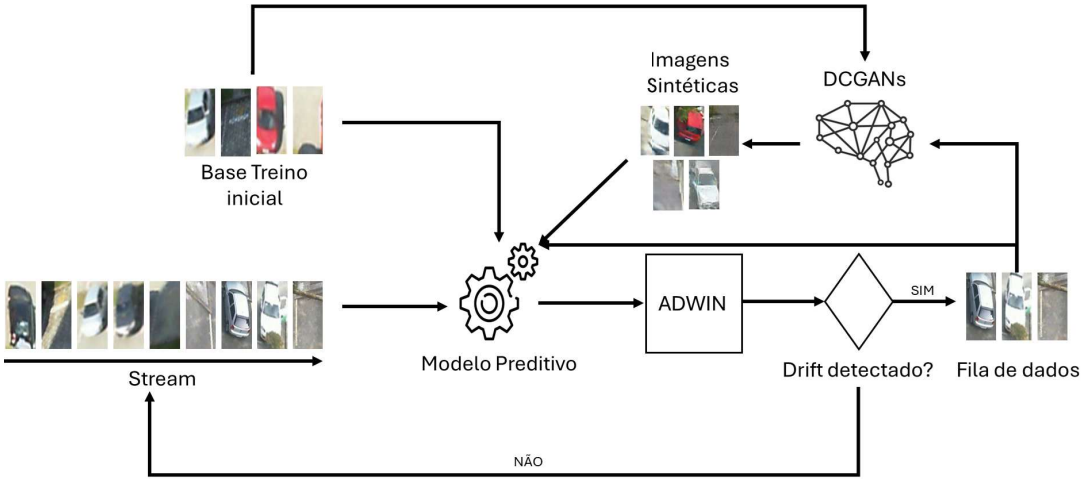


Figura 7.1: Protocolo Experimental realizado.

ADWIN.

Quando uma mudança de conceito é detectada, a janela de dados correspondente é separada, e, ao final do dia processado, as redes generativas são atualizadas. A partir desse ponto, as DCGANs sintetizam 2000 novas imagens (1000 de cada classe) para aumentar o conjunto de dados na atualização da rede convolucional, realizada em conjunto com as imagens reais presentes na janela do detector de mudanças de conceito. O processamento das imagens subsequentes é então retomado, com monitoramento contínuo das mudanças de conceito e atualização dos modelos conforme necessário.

Para uma análise comparativa do desempenho dos modelos e da influência dos dados sintéticos nas atualizações preditivas, consideraram-se quatro cenários:

- Cenário sem atualização do modelo preditivo.
- Cenário com atualização do modelo preditivo apenas com dados reais (da janela do detector de mudanças de conceito).
- Cenário com atualização do modelo preditivo usando dados reais (da janela do detector de mudanças de conceito) e aumento de dados gerado por técnicas convencionais com o auxílio do pacote *Tensorflow*.
- Cenário com atualização do modelo preditivo usando dados reais (da janela do detector de mudanças de conceito) e dados sintéticos gerados pelas redes DCGAN.

7.2 Resultados Obtidos

Esta seção apresenta e discute os resultados dos experimentos. A partir deste ponto, adotam-se as seguintes notações para identificar os cenários testados nas tabelas apresentadas:

- C1: Modelo preditivo sem atualização ao longo do processamento do fluxo de imagens.
- C2: Atualização do modelo preditivo utilizando apenas os dados reais contidos na janela do detector de mudanças de conceito no momento da detecção.
- C3: Atualização do modelo preditivo com dados reais da janela do detector de mudanças de conceito e aumento de dados por técnicas convencionais utilizando o pacote TensorFlow.
- C4: Atualização do modelo preditivo com dados reais da janela do detector de mudanças de conceito e dados sintetizados pelas DCGANs.

As tabelas apresentam as médias e desvios padrão (D.P.) obtidos a partir das execuções realizadas.

7.2.1 Fluxo de Imagens CNRPark-Ext → PKLot

Os experimentos realizados no fluxo de imagens CNRPark-Ext → PKLot resultaram nos valores de acurácia apresentados na Tabela 7.1, conforme o cenário e modelo utilizado.

	MobileNetV3				ResNet50				VGG16			
Execução	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
1	90,11	95,80	97,24	97,08	81,01	97,03	97,41	97,99	86,19	98,67	98,71	97,55
2	81,08	96,52	96,95	98,28	78,81	97,42	97,20	97,75	94,55	98,28	98,48	97,21
3	62,33	97,57	97,54	97,25	76,85	97,27	96,50	97,65	93,66	98,26	98,95	98,40
4	91,62	95,59	96,53	97,21	74,64	96,84	96,79	97,77	94,69	98,57	98,24	97,93
5	93,19	98,37	96,69	97,64	84,38	97,37	97,31	97,63	92,86	98,16	98,85	98,39
6	93,39	96,99	97,64	97,73	77,94	97,72	96,19	96,36	94,33	97,81	98,54	98,12
7	87,90	96,93	96,62	97,28	84,64	97,01	97,59	97,13	95,18	96,18	98,71	96,89
8	92,48	96,89	97,23	97,83	81,58	96,83	96,64	97,61	92,71	98,13	97,66	97,12
9	92,53	97,93	97,00	97,35	77,42	97,29	96,79	97,16	90,44	98,30	98,38	96,89
10	90,51	96,72	97,25	97,80	79,45	97,10	97,96	97,12	95,24	98,33	98,31	97,52
Média:	**87,51**	**96,93**	**97,07**	**97,65**	**79,37**	**97,19**	**97,04**	**97,42**	**92,99**	**98,07**	**98,48**	**97,60**
D.P.:	**9,57**	**0,92**	**0,38**	**0,44**	**3,40**	**0,28**	**0,55**	**0,48**	**2,79**	**0,70**	**0,37**	**0,54**

Tabela 7.1: Acurácias obtidas (%) envolvendo o fluxo de imagens CNRPark-Ext → PKLot.

Observa-se que as redes MobileNetV3 e ResNet50 tendem a apresentar melhores resultados médios quando as redes generativas são empregadas como método auxiliar na atualização do classificador ao longo do fluxo. No entanto, a rede VGG16 apresenta uma queda de desempenho nesse cenário, em comparação com os cenários que utilizam apenas dados reais ou técnicas convencionais de aumento de dados.

Para verificar estatisticamente a melhora observada, foram realizados os testes de hipótese de Friedman e post-hoc de Nemenyi.

A Tabela 7.2 apresenta os valores-p obtidos no teste de Friedman para comparação dos quatro cenários testados, considerando cada classificador. As hipóteses adotadas no teste foram:

- H_0: Não há diferença significativa entre as estratégias de atualização dos modelos diante de mudanças de conceito.
- H_1: Há diferença significativa entre as diferentes estratégias de atualização dos modelos diante de mudanças de conceito.

Classificador	**Valor-p**
MobileNetV3	$1,485x10^{-4}$
ResNet50	$7,459x10^{-5}$
VGG16	$3,740x10^{-5}$

Tabela 7.2: Valores-p relacionados às acurácias obtidas no cenário com o fluxo CNRPark-Ext → PKLot

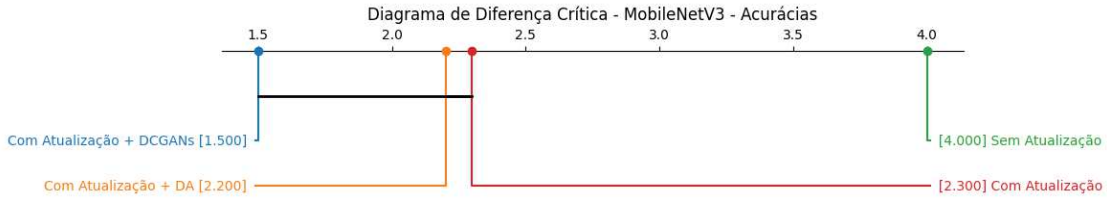
Como todos os valores-p são inferiores a 5%, rejeita-se a hipótese nula para todos os classificadores. Assim, é necessário aplicar o teste post-hoc de Nemenyi para identificar quais cenários apresentam diferenças estatisticamente significativas em termos de acurácia.

Os resultados do teste de Nemenyi são apresentados na Tabela 7.3, com diferenças estatisticamente significativas destacadas em negrito. Para melhor visualização da comparação entre cenários e classificadores, os diagramas de diferença crítica estão ilustrados na Figura 7.2.

Assim, temos que no caso da rede MobileNetV3 todos os cenários com atualização do modelo apresentam diferenças significativas em relação ao cenário sem atualização. A Figura 7.2a mostra que, embora os cenários de atualização não diferenciem estatisticamente entre si, o uso de redes generativas apresenta desempenho superior.

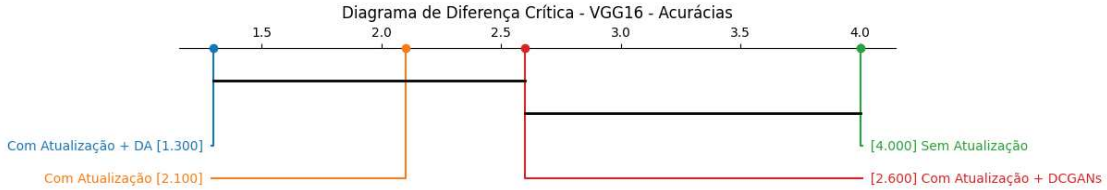
	MobileNetV3				ResNet50				VGG16			
	C1	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**
C1	1,000	**0,017**	**0,010**	**0,000**	1,000	**0,006**	**0,046**	**0,000**	1,000	**0,006**	**0,000**	0,072
C2	**0,017**	1,000	0,998	0,508	**0,006**	1,000	0,900	0,619	**0,006**	1,000	0,508	0,822
C3	**0,010**	0,998	1,000	0,619	**0,046**	0,900	1,000	0,226	**0,000**	0,508	1,000	0,110
C4	**0,000**	0,508	0,619	1,000	**0,000**	0,619	0,226	1,000	0,072	0,822	0,110	1,000

Tabela 7.3: Valores-p obtidos com o teste de Nemenyi em relação às acurácias obtidas para o fluxo de imagens CNRPark-Ext → PKLot.



(a) MobileNetV3

(b) ResNet50



(c) VGG16

Figura 7.2: Diagramas de diferenças obtidos em termos de acurácia dentre os diferentes cenários e redes para o fluxo de imagens CNRPark-Ext → PKLot.

Para a rede ResNet50, temos um resultado similar aos experimentos realizados para a rede MobileNetV3. O cenário sem atualização apresenta desempenho significativamente inferior aos demais, que não diferem entre si. No entanto, o cenário que utiliza redes generativas tende a se destacar, conforme ilustrado na Figura 7.2b.

Por fim, os experimentos realizados com a rede VGG16 demonstram comportamento diferentes às demais redes utilizadas. O cenário que emprega aumento de dados convencional apresenta melhor desempenho, enquanto o uso de redes generativas não demonstra diferença estatisticamente significativa em relação ao cenário sem atualização (Figura 7.2c).

A Tabela 7.4 apresenta o número de detecções de mudanças de conceito por modelo, cenário e execução, além da média de detecções e seu respectivo desvio padrão.

	MobileNetV3				ResNet50				VGG16			
Execução	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
1	44	22	19	23	68	10	12	11	58	8	12	13
2	59	16	31	14	72	18	23	12	13	14	13	13
3	87	17	32	16	70	12	15	15	24	11	6	11
4	25	43	37	24	68	15	25	13	17	8	9	6
5	23	21	31	20	72	16	18	23	41	10	10	12
6	42	21	23	18	68	16	12	11	14	9	12	9
7	46	23	29	23	62	15	17	18	18	16	9	15
8	38	23	21	20	64	14	24	16	24	7	14	10
9	27	24	37	21	85	18	11	12	35	10	12	11
10	46	20	28	20	74	14	11	18	26	12	7	13
Média:	**43,70**	**23,00**	**28,80**	**19,90**	**70,30**	**14,80**	**16,80**	**14,90**	**27,00**	**10,50**	**10,40**	**11,30**
D.P.:	**18,89**	**7,94**	**6,20**	**3,18**	**6,33**	**2,49**	**5,53**	**3,90**	**14,09**	**2,84**	**2,63**	**2,54**

Tabela 7.4: Número de detecções de mudanças de conceito envolvendo o fluxo de imagens CNRPark-Ext → PKLot.

Por fim, os valores-p obtidos no teste de Friedman para os três classificadores estão na 7.5. Como todos os valores-p são inferiores a 5%, rejeita-se a hipótese nula, confirmando que as diferentes estratégias de atualização dos modelos apresentam diferenças estatisticamente significativas.

Classificador	**Valor-p**
MobileNetV3	$4,944x10^{-4}$
ResNet50	$2,581x10^{-4}$
VGG16	$2,051x10^{-3}$

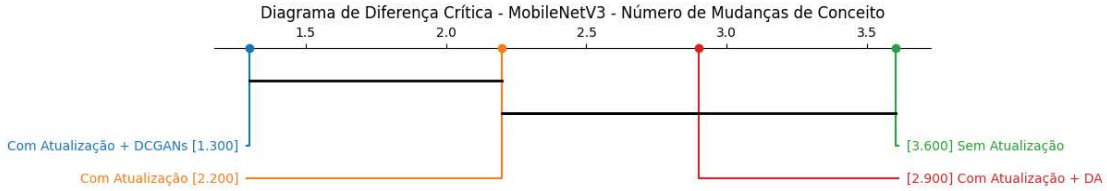
Tabela 7.5: Valores-p relacionados ao número de detecções de mudanças de conceito no cenário com o fluxo CNRPark-Ext → PKLot

Para identificar quais cenários apresentam diferenças significativas, o teste post-hoc de Nemenyi foi realizado para todos os classificadores. Os resultados estão disponíveis na Tabela 7.6, enquanto os diagramas de diferença crítica obtidos são apresentados na Figura 7.3.

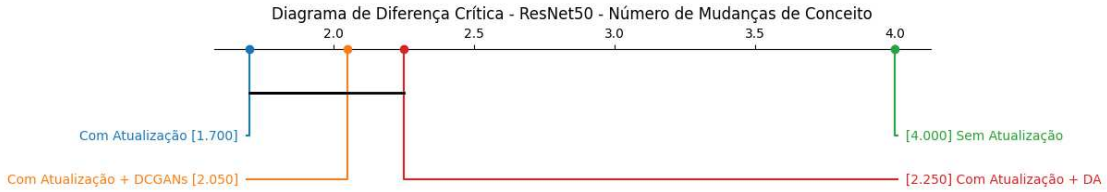
Como pode-se observar na Tabela 7.6 e na Figura 7.3a, no caso do classificador MobileNetV3, embora não haja diferença significativa entre o cenário com redes generativas e o cenário com atualização usando apenas dados reais, o uso de redes generativas se destaca como ligeiramente melhor em termos de número de mudanças de conceito detectadas. Isso indica que o modelo apresenta ganhos

	MobileNetV3				ResNet50				VGG16			
	C1	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**
C1	1,000	0,072	0,619	**0,000**	1,000	**0,000**	**0,013**	**0,004**	1,000	**0,007**	**0,013**	**0,029**
C2	0,072	1,000	0,619	0,402	**0,000**	1,000	0,776	0,930	**0,007**	1,000	0,998	0,973
C3	0,619	0,619	1,000	**0,028**	**0,013**	0,776	1,000	0,986	**0,013**	0,998	1,000	0,994
C4	**0,000**	0,402	**0,028**	1,000	**0,004**	0,930	0,986	1,000	**0,029**	0,973	0,994	1,000

Tabela 7.6: Valores-p obtidos com o teste de Nemenyi em relação ao número de mudanças de conceito detectadas para o fluxo de imagens CNRPark-Ext → PKLot.



(a) MobileNetV3



(b) ResNet50

(c) VGG16

Figura 7.3: Diagramas de diferenças obtidos em termos de detecção de mudanças de conceito dentre os diferentes cenários e redes para o fluxo de imagens CNRPark-Ext → PKLot.

significativos em estabilidade quando comparado ao uso de aumento de dados convencional e ao cenário sem atualização do classificador.

Analisando os resultados para a rede ResNet50, os cenários com atualização apresentam diferenças significativas em relação ao cenário sem atualização, mas não diferem entre si. A Figura 7.3b mostra que o cenário com atualização apenas usando dados reais apresenta maior estabilidade ao longo do fluxo de imagens, resultando em um menor número de mudanças de conceito detectadas e, consequentemente, menor necessidade de retreinamento do modelo.

Com base na Figura 7.3c, temos que o comportamento da rede VGG16 é semelhante ao da ResNet50, com os cenários atualizados apresentando diferenças estatisticamente significativas em relação ao cenário sem atualização. No entanto, o cenário com aumento de dados convencional apresenta desempenho ligeiramente melhor que o cenário com redes generativas. Isso se reflete em um menor número de detecções de mudanças de conceito ao longo do fluxo, reduzindo a necessidade de ajustes frequentes do modelo

7.2.2 Fluxo de Imagens PKLot → CNRPark-Ext

Os experimentos realizados para o fluxo de imagens PKLot → CNRPark-Ext produziram os resultados de acurácia apresentados na Tabela 7.7, variando conforme o classificador e o cenário de atualização do modelo.

	MobileNetV3				ResNet50				VGG16			
Execução	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
1	96,64	96,97	98,30	98,60	95,95	96,76	97,58	97,78	97,04	97,71	98,43	98,54
2	98,05	98,21	97,73	97,46	94,55	97,10	97,39	96,42	95,45	97,94	98,34	98,42
3	95,92	97,03	96,15	96,62	94,48	97,59	97,55	97,89	93,33	98,35	98,06	97,53
4	96,30	97,89	98,05	98,52	94,70	97,82	98,20	97,40	96,23	98,43	98,39	98,20
5	97,38	95,75	95,12	98,49	95,05	97,39	97,75	96,99	94,24	97,95	98,56	98,22
6	92,54	97,10	96,91	96,54	94,10	98,13	97,68	97,57	95,90	98,53	98,49	98,40
7	74,76	97,97	97,71	97,59	93,61	97,05	97,73	97,76	96,35	98,29	98,72	98,25
8	88,62	96,41	97,03	96,19	94,35	97,22	97,64	97,27	95,48	97,88	92.78	98,01
9	91,82	96,53	97,05	97,64	95,61	96,76	96,86	97,32	95,77	98,27	98,39	97,32
10	92,58	97,68	97,15	98,23	93,53	97,37	97,22	97,03	95,76	97,20	94,19	97,25
Média:	**92,46**	**97,15**	**97,12**	**97,59**	**94,59**	**97,32**	**97,56**	**97,34**	**95,56**	**98,06**	**97,44**	**98,01**
D.P.:	**6,90**	**0,79**	**0,94**	**0,89**	**0,78**	**0,44**	**0,35**	**0,45**	**1,06**	**0,40**	**2,11**	**0,45**

Tabela 7.7: Acurácias obtidas envolvendo o fluxo de imagens PKLot → CNRPark-Ext.

Com o objetivo de verificar a hipótese de que há diferença significativa entre os cenários testados, foram realizados testes estatísticos de Friedman para cada rede. Seus respectivos valores-p são apresentados na Tabela 7.8. Como todos os valores-p são menores do que 5% (nível de significância utilizado para o teste, rejeita-se a hipótese nula que diz que não há diferença significativa entre o desempenho dos diferentes cenários testados.

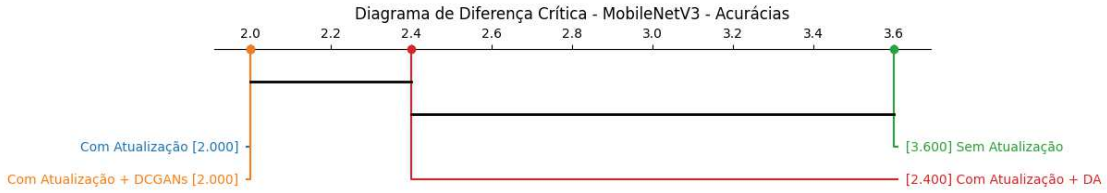
Como a hipótese nula foi rejeitada, o teste post-hoc de Nemenyi foi realizado para todos os classificadores com o objetivo de observamos quais cenários apresentam desempenho significativamente diferente dos demais, com os valores-p obtidos por meio do teste sendo representados na Tabela 7.9. Os diagramas de diferença crítica obtidos estão representados na Figura 7.4.

Classificador	Valor-p
MobileNetV3	$1,603x10^{-2}$
ResNet50	$2,950x10^{-4}$
VGG16	$3,570x10^{-3}$

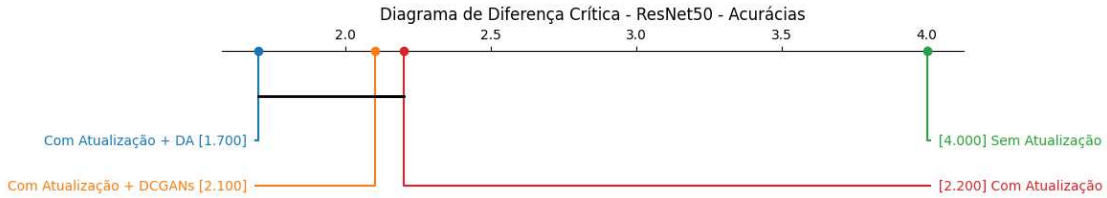
Tabela 7.8: Valores-p relacionados ao número de detecções de mudanças de conceito no cenário com o fluxo PKLot → CNRPark-Ext.

	MobileNetV3				ResNet50				VGG16			
	C1	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**
C1	1,000	**0,029**	0,160	**0,029**	1,000	**0,010**	**0,000**	**0,006**	1,000	**0,010**	**0,017**	**0,017**
C2	**0,029**	1,000	0,900	1,000	**0,010**	1,000	0,822	0,998	**0,010**	1,000	0,998	0,998
C3	0,160	0,900	1,000	0,900	**0,000**	0,822	1,000	0,900	**0,017**	0,998	1,000	1,000
C4	**0,029**	1,000	0,900	1,000	**0,006**	0,998	0,900	1,000	**0,017**	0,998	1,000	1,000

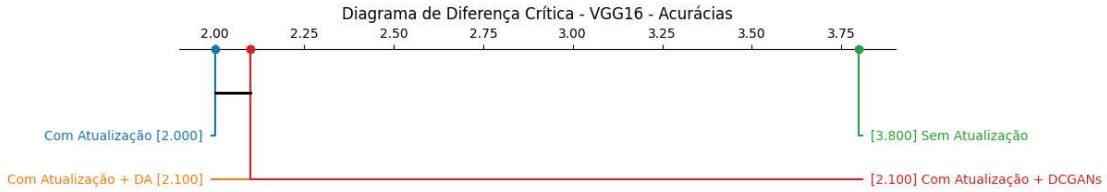
Tabela 7.9: Valores-p obtidos com o teste de Nemenyi em relação às acurácias obtidas para o fluxo de imagens PKLot → CNRPark-Ext.



(a) MobileNetV3



(b) ResNet50



(c) VGG16

Figura 7.4: Diagramas de diferenças obtidos em termos de detecção de acurácia dentre os diferentes cenários e redes para o fluxo de imagens PKLot → CNRPark-Ext.

Os testes realizados com a rede MobileNetV3 indicam que o cenário em que são utilizadas redes generativas durante o processo de atualização do modelo apresenta uma média de desempenho ligeiramente superior em comparação

aos cenários onde o modelo é atualizado de outras formas ao longo do fluxo de imagens. O teste estatístico de Friedman (Tabela 7.8) revelou uma diferença significativa entre os cenários ($p = 1,60310^{-2}$), o que motivou a realização do teste post-hoc de Nemenyi. Com base nos valores da Tabela 7.9 e na Figura 7.4a, observa-se que os cenários onde o modelo foi atualizado com dados reais ou imagens sintéticas possuem desempenho similar, enquanto ambos apresentam diferença estatisticamente significativa em relação ao cenário onde o classificador não passou por atualização.

No caso da ResNet50, os resultados indicam que os modelos que passaram por alguma estratégia de atualização ao longo do fluxo de imagens obtiveram acurácia significativamente maior quando comparados ao cenário em que a rede não foi atualizada. Os valores médios da Tabela 7.7 sugerem que o aumento de dados convencional proporciona um desempenho ligeiramente superior aos demais métodos de atualização. O teste de Friedman rejeitou a hipótese nula ($p = 2,95010^{-4}$), confirmando que existem diferenças entre os cenários. A análise da Tabela 7.9 e da Figura 7.4b indica que o cenário sem atualização apresenta desempenho estatisticamente inferior, enquanto os cenários com alguma forma de atualização apresentam desempenhos semelhantes, com o aumento de dados convencional mostrando uma leve vantagem.

Já para a VGG16, os resultados observados são semelhantes aos obtidos na ResNet50, com a atualização do modelo garantindo desempenho significativamente melhor em relação ao cenário sem atualização. O teste de Friedman confirmou a existência de diferenças entre os cenários ($p = 3,57010^{-3}$). A análise da Tabela 7.7 sugere que o cenário em que somente dados reais são utilizados para a atualização do modelo apresenta um desempenho ligeiramente superior aos demais cenários de atualização. Além disso, a Tabela 7.9 e a Figura 7.4c mostram que o cenário sem atualização é significativamente pior do que os demais, enquanto o cenário baseado exclusivamente em dados reais possui um pequeno destaque positivo em comparação aos demais métodos de atualização.

Os valores obtidos para o número de mudanças de conceito detectadas ao longo do fluxo de imagens das vagas de estacionamento são apresentados na Tabela 7.10.

A análise dos dados da Tabela 7.10 revela que a quantidade de mudanças de conceito varia conforme o classificador utilizado e o cenário de atualização do modelo. Para verificar a significância estatística dessas diferenças, foi aplicado o teste de Friedman, cujos valores-p estão representados na Tabela 7.11. Como

	MobileNetV3				ResNet50				VGG16			
Execução	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
1	28	17	17	12	26	16	15	8	26	7	17	7
2	19	22	24	26	38	16	16	10	27	17	10	11
3	24	21	26	12	36	13	15	12	33	5	12	13
4	32	15	12	9	40	24	13	12	25	13	12	5
5	26	32	20	19	24	11	13	16	24	12	9	8
6	60	16	22	21	48	20	12	16	15	10	14	12
7	77	20	28	18	46	20	19	15	20	13	10	7
8	95	35	15	22	38	15	20	14	35	12	136	9
9	29	15	27	19	22	11	18	17	21	10	13	12
10	21	20	17	14	30	10	8	13	9	10	70	9
Média:	**41,10**	**21,30**	20,80	**17,20**	**34,80**	**15,60**	14,90	**13,30**	**23,50**	**10,90**	30,30	**9,30**
D.P.:	**26,59**	**6,93**	5,47	**5,31**	**9,00**	**4,60**	3,60	**2,87**	**7,78**	**3,35**	41,42	**2,87**

Tabela 7.10: Número de detecções de mudanças de conceito envolvendo o fluxo de imagens PKLot → CNRPark-Ext.

todos os valores-p são menores do que 5%, rejeita-se a hipótese nula de que não há diferença significativa no número de mudanças de conceito detectadas entre os diferentes cenários testados.

Classificador	**Valor-p**
MobileNetV3	$1,111x10^{-2}$
ResNet50	$1,990x10^{-4}$
VGG16	$3,091x10^{-3}$

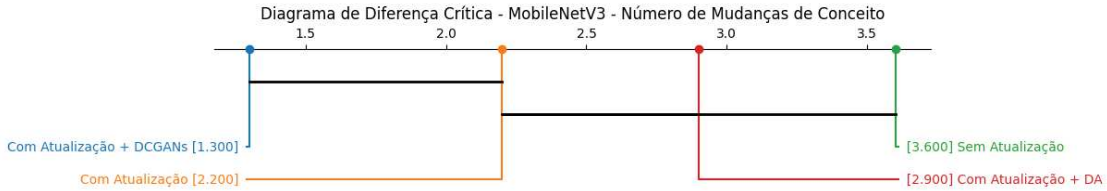
Tabela 7.11: Valores-p relacionados ao número de detecções de mudanças de conceito no cenário com o fluxo PKLot → CNRPark-Ext.

Com a rejeição da hipótese nula, foi realizado o teste post-hoc de Nemenyi para todos os classificadores, conforme ilustrado na Tabela 7.12 e no diagrama de diferença crítica da Figura 7.5.

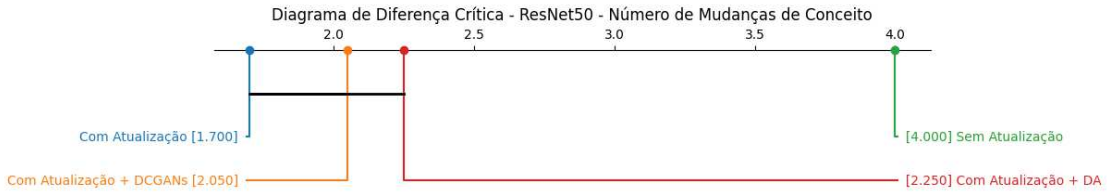
	MobileNetV3				**ResNet50**				**VGG16**			
	C1	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**	**C1**	**C2**	**C3**	**C4**
C1	1,000	0,191	0,353	**0,006**	1,000	**0,013**	**0,007**	**0,000**	1,000	**0,046**	0,264	**0,022**
C2	0,191	1,000	0,986	0,564	**0,013**	1,000	0,998	0,674	**0,046**	1,000	0,864	0,776
C3	0,353	0,986	1,000	0,353	**0,007**	0,998	1,000	0,776	0,264	0,864	1,000	0,307
C4	**0,006**	0,564	0,353	1,000	**0,000**	0,674	0,776	1,000	**0,022**	0,776	0,307	1,000

Tabela 7.12: Valores- p obtidos com o teste de Nemenyi em relação ao número de detecções de mudanças de conceito para o fluxo de imagens PKLot → CNRPark-Ext.

A análise da Figura 7.5a indica que, para a MobileNetV3, o cenário em que são utilizadas redes generativas no processo de atualização do modelo apresenta um desempenho significativamente melhor quando comparado ao uso do



(a) MobileNetV3



(b) ResNet50

(c) VGG16

Figura 7.5: Diagramas de diferenças obtidos em termos de detecção de mudanças de conceito dentre os diferentes cenários e redes para o fluxo de imagens PKLot → CNRPark-Ext.

aumento de dados convencional ou à ausência de atualização dos pesos da rede durante o processamento do fluxo de imagens. No entanto, embora esse cenário seja ligeiramente superior, ele não apresenta uma diferença estatisticamente significativa quando comparado ao cenário que utiliza apenas dados reais para atualização do modelo.

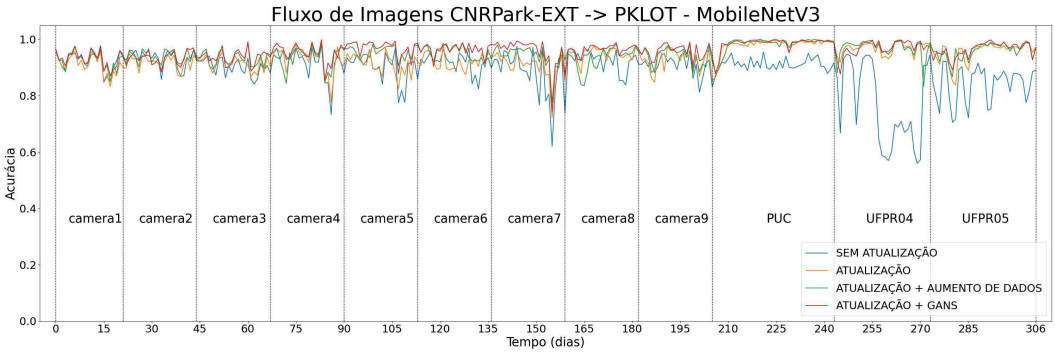
Para a ResNet50, conforme representado na Figura 7.5b, os três cenários que envolvem a atualização do modelo não apresentam diferenças estatisticamente significativas entre si, mas todos resultam em um menor número de detecções de mudanças de conceito ao longo do fluxo de imagens em comparação ao cenário em que o classificador não passa por atualizações.

Por fim, a Figura 7.5c mostra que a VGG16 exibe resultados semelhantes aos da ResNet50 no que diz respeito ao número de detecções de mudanças de conceito. No entanto, observa-se que, apesar de não haver diferença estatisticamente significativa, o cenário baseado no aumento de dados apresenta um desempenho ligeiramente superior ao cenário onde são utilizados dados sintéticos para atualização do modelo.

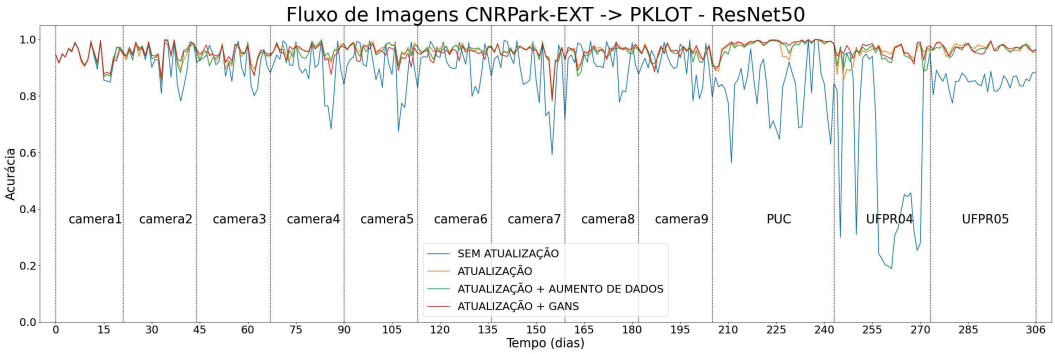
7.2.3 Avaliação de Mudanças de Conceito

CNRPark-Ext → PKLot

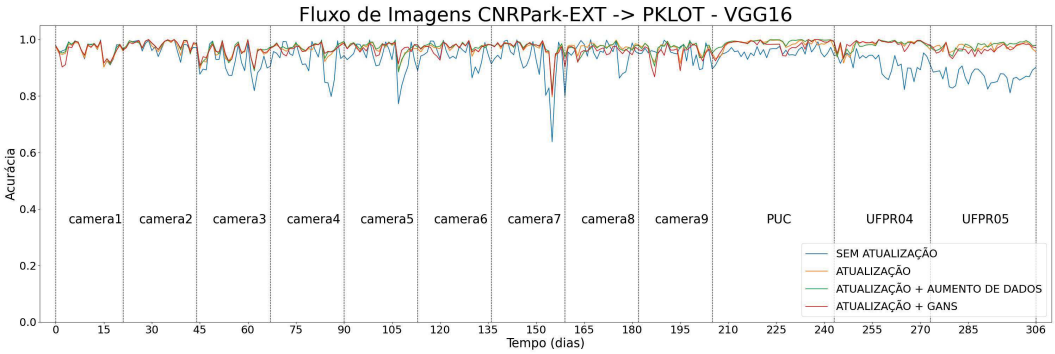
A Figura 7.6 apresenta a evolução da acurácia ao longo dos dias no fluxo de dados CNRPark-Ext → PKLot, onde é possível identificar pontos de queda de desempenho, indicando potenciais mudanças de conceito.



(a) MobileNetV3.



(b) ResNet50.



(c) VGG16.

Figura 7.6: Desempenho em termos de acurácia ao longo do fluxo de imagens CNRPark-EXT → PKLot.

A mudança de conceito mais notável nos gráficos ocorre dentro do subcon-

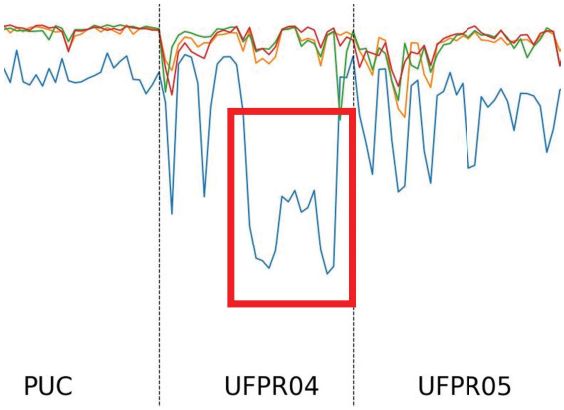


Figura 7.7: Mudança de conceito relacionada a alteração da distribuição entre classes.

UFPR04 - Dia 22/12/2012 UFPR04 - Dia 23/12/2012

Figura 7.8: Ilustração da Mudança de Conceito encontrada na Figura 7.7.

junto UFPR04, conforme destacado principalmente nas Figuras 7.6a e 7.6b no cenário sem atualização do classificador, e com maior evidência na Figura 7.7. A principal justificativa para esse comportamento está no fato de esse período corresponder ao final do ano e período de férias. Durante esse intervalo, o estacionamento passa a permanecer predominantemente desocupado, resultando na queda de desempenho observada. Esse comportamento é ilustrado na Figura 7.8.

Essa mudança de conceito continua impactando o desempenho no subconjunto UFPR05, quando há a transição entre dias úteis e finais de semana. Nessa mudança, ocorre uma alteração abrupta na taxa de ocupação do estacionamento, contribuindo para a redução do desempenho do modelo.

Outra mudança de conceito perceptível pode ser observada nos subconjuntos camera4, camera5, camera6 e camera7 da base CNRPark-Ext, conforme evidenciado na Figura 7.9. Nesse caso, a queda de desempenho está associada a mudanças climáticas, onde o clima ensolarado passa a ser chuvoso e,

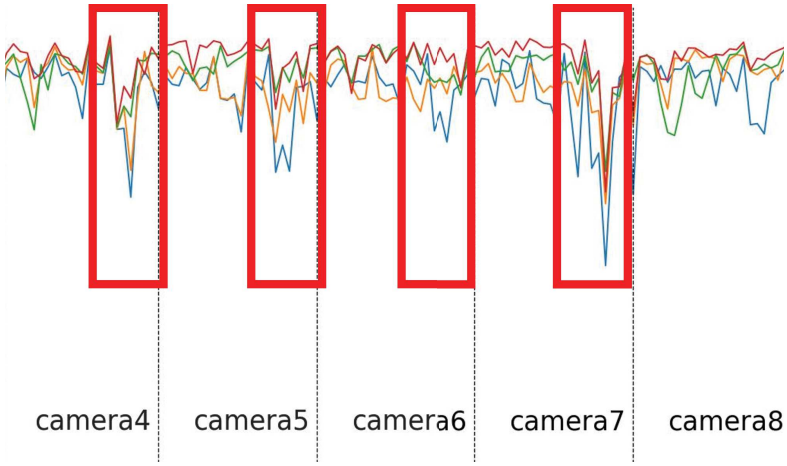


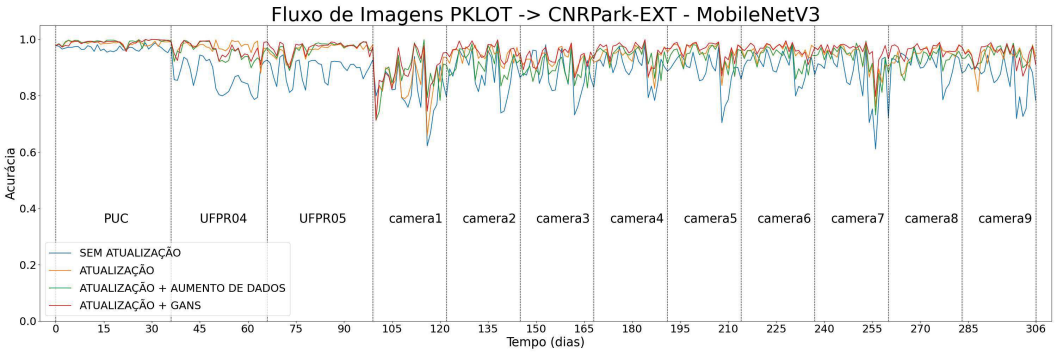
Figura 7.9: Mudança de conceito relacionada a alterações climáticas.

Figura 7.10: Ilustração da Mudança de Conceito encontrada na Figura 7.9.

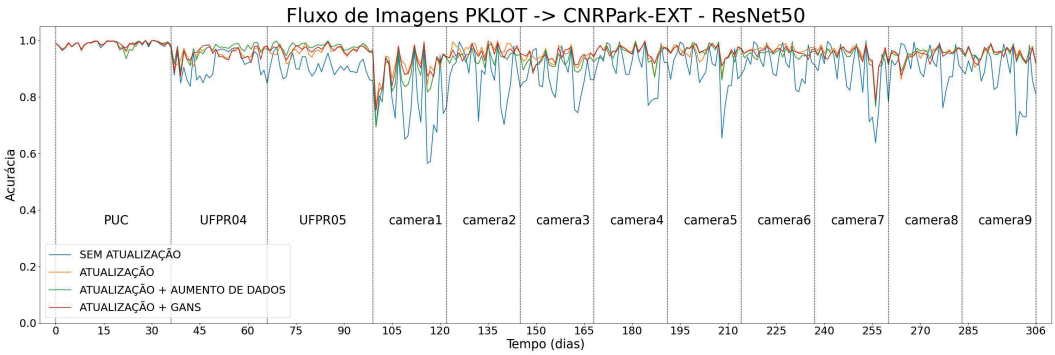
posteriormente, retorna ao tempo ensolarado, como ilustrado na Figura 7.10.

PKLot → CNRPark-Ext

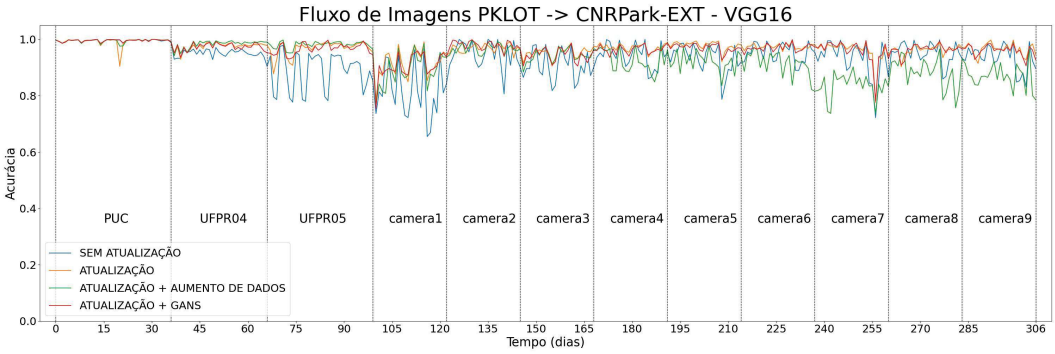
A figura 7.11 apresenta os resultados obtidos para os três classificadores em diferentes cenários experimentados no fluxo de imagens PKLot → CNRPark-Ext.



(a) MobileNetV3.



(b) ResNet50.



(c) VGG16.

Figura 7.11: Desempenho em termos de acurácia ao longo do fluxo de imagens PKLot → CNRPark-EXT.

A partir dos gráficos, observa-se que o desempenho dos modelos sofre quedas em pontos de transição entre subconjuntos da mesma base, como no caso

da mudança entre PUC e UFPR04. Esse fenômeno é destacado nas Figuras 7.12 e 7.13 e pode ser explicado pela mudança na angulação da câmera em relação ao estacionamento, além de modificações na posição das vagas monitoradas.

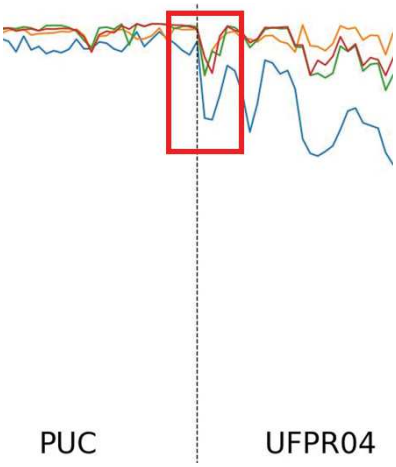


Figura 7.12: Mudança de conceito relacionada a alterações entre estacionamentos.

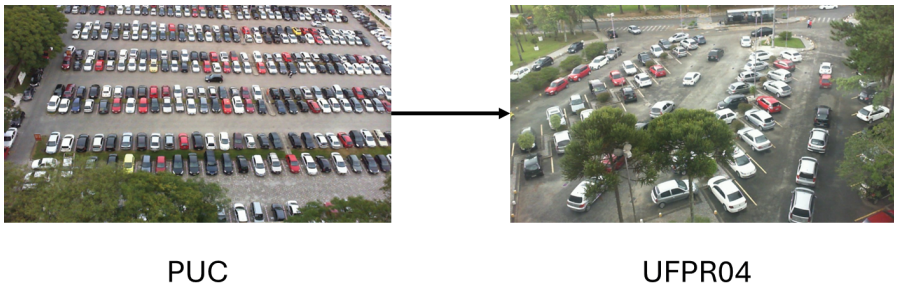


Figura 7.13: Ilustração da Mudança de Conceito encontrada na Figura 7.12.

Além disso, há a transição entre as bases de imagens PKLot e CNRPark-Ext, onde ocorrem mudanças no estacionamento e no ângulo da câmera, bem como a troca da própria câmera utilizada na captura das imagens. Essas alterações afetam a qualidade e a resolução das imagens captadas, influenciando o desempenho dos modelos. O momento exato da detecção da mudança de conceito, bem como exemplos dos dados envolvidos na transição, podem ser visualizados nas Figuras 7.14 e 7.15.

Apesar de serem analisadas diferentes causas de mudanças de conceito para cada fluxo de imagens neste trabalho, ao comparar os gráficos representados nas Figuras 7.6 e 7.11 podemos notar que as mesmas mudanças de conceito ocorrem nos dois casos, não sendo restritos à um ou outro fluxo de dados em específico.

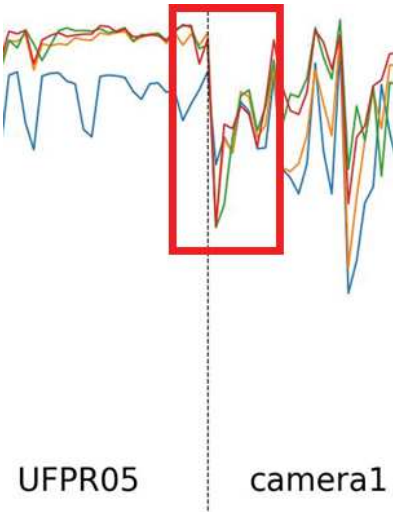


Figura 7.14: Mudança de conceito relacionada a alterações entre bases de dados.

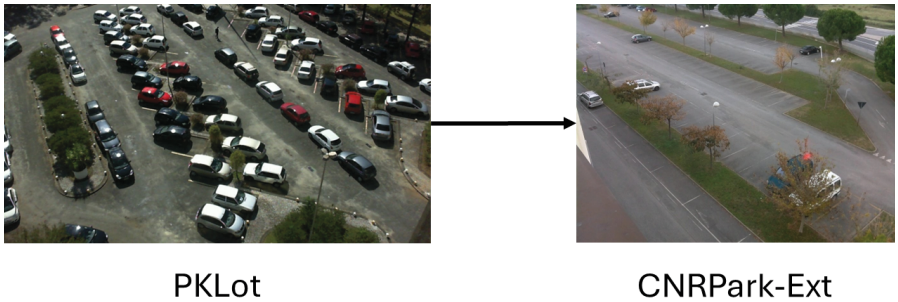


Figura 7.15: Ilustração da Mudança de Conceito encontrada na Figura 7.14.

Por fim, os gráficos que ilustram o desempenho dos modelos ao longo do processamento do fluxo de imagens indicam que os cenários onde o classificador é atualizado conseguem recuperar rapidamente seu desempenho, enquanto os cenários sem atualização das redes sofrem impacto negativo, resultando em um desempenho inferior durante esses eventos.

8

Conclusões

Este trabalho investigou o uso de Redes Generativas Adversárias (GANs) para mitigação de mudanças de conceito em fluxos contínuos de dados, aplicando a abordagem em bases de imagens de estacionamento. A hipótese inicial propôs que a aplicação de GANs como método auxiliar reduziria os efeitos das mudanças de conceito, melhorando a acurácia preditiva e reduzindo o número de mudanças de conceito.

Respondendo a pergunta de pesquisa formulada no Capítulo 1, que questiona sobre o impacto no desempenho do modelo quanto são aplicadas redes DCGANs como método de aumento de dados para sua atualização, temos que

- Para a rede MobileNetV3, em ambos os fluxos de dados experimentados, o cenário que utiliza DCGANs como estratégia para a atualização dos pesos do modelo apresenta melhor desempenho, tanto em acurácia quanto em número de mudanças de conceito detectadas. Esse cenário demonstra a aquisição de maior estabilidade (redução da necessidade de retreinamento do classificador), além de apresentar desempenho ligeiramente melhor em comparação aos outros cenários. Isso sugere que a contribuição do uso das redes DCGANs é eficaz para melhorar o desempenho geral durante o processamento do fluxo de imagens em redes com menor número de parâmetros.

- Já para a rede ResNet50, observa-se que os cenários com atualização dos dados variam em relação ao desempenho, tanto em acurácia quanto no número de detecções de mudanças de conceito. Não há diferença significativa entre esses cenários em nenhum momento. Isso indica que, em redes com um número relativamente maior de parâmetros, a estratégia de utilização de redes generativas não apresenta eficácia quando comparada aos demais cenários que envolvem atualização do classificador.

- Por fim, para a rede VGG16, o cenário que utiliza dados sintetizados por redes generativas acaba apresentando o pior desempenho entre os três cenários de atualização do modelo, considerando as métricas avaliadas. No entanto, não há diferenças significativas entre os três cenários. Com esse classificador, o cenário que utiliza apenas dados reais para atualização destaca-se em ambos os fluxos de dados para ambas as métricas, apresentando desempenho ligeiramente superior aos demais cenários em todos os casos.

Com isso, conclui-se que o impacto do uso de redes generativas é limitado a redes com menor número de parâmetros, não havendo melhoria em redes maiores, como a ResNet50 e a VGG16, quando comparadas a outras estratégias de atualização do modelo, incluindo aumento de dados convencional.

Dessa forma, pode-se concluir que a hipótese é parcialmente confirmada, pois as redes DCGANs mostraram maior eficácia em melhorar a acurácia, além de reduzir a necessidade de atualização dos modelos por meio da detecção de mudanças de conceito em redes menores. No entanto, para modelos mais robustos, houve queda de performance quando foram usados dados sintetizados, evidenciando que o método não é capaz de melhorar o desempenho com a utilização desses modelos.

Possíveis trabalhos futuros para enriquecer o trabalho apresentado neste presente documento são:

- Exploração de outras arquiteturas de redes generativas, como, por exemplo, *Conditional* GANs e WGANs.
- Ampliação dos experimentos para cenários com maior diversidade de mudanças de conceito, envolvendo bases de imagens de outras naturezas.
- Utilização de uma abordagem não supervisionada para o problema, uma vez que, com o uso do ADWIN, há a necessidade do rótulo real das amostras processadas no fluxo de dados.

Referências

AGGARWAL, C. C. (Ed.). *Data Streams - Models and Algorithms*. Springer, 2007. v. 31. (Advances in Database Systems, v. 31). ISBN 978-0-387-28759-1. Disponível em: <http://dblp.uni-trier.de/db/series/ads/ads31.html>.

AGRAHARI, S.; SINGH, A. K. Concept drift detection in data stream mining : A literature review. *Journal of King Saud University - Computer and Information Sciences*, v. 34, n. 10, Part B, p. 9523–9540, 2022. ISSN 1319-1578. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1319157821003062>.

AMATO, G.; CARRARA, F.; FALCHI, F.; GENNARO, C.; VAIRO, C. Car parking occupancy detection using smart camera networks and deep learning. In: IEEE. *Computers and Communication (ISCC), 2016 IEEE Symposium on*. [S.l.], 2016. p. 1212–1217.

AMATO, G.; CARRARA, F.; FALCHI, F.; GENNARO, C.; MEGHINI, C.; VAIRO, C. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, Pergamon, v. 72, p. 327–334, 2017.

BARDDAL, J. P. *Agrupamento Online: Uma Abordagem Baseada na Teoria de Redes Sociais*. 158 p. Dissertação (Mestrado em Informática) — Pontifícia Universidade Católica do Paraná, Curitiba, 2015.

BIFET, A.; GAVALDà, R. Learning from time-changing data with adaptive windowing. In: . [S.l.: s.n.], 2007. v. 7.

CAUWENBERGHS, G.; POGGIO, T. Incremental and decremental support vector machine learning. In: LEEN, T.; DIETTERICH, T.; TRESP, V. (Ed.). *Advances in Neural Information Processing Systems*. MIT Press, 2000. v. 13. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2000/file/155fa09596c7e18e50b58eb7e0c6ccb4-Paper.pdf>.

CHOLLET, F. *Deep Learning with Python*. Manning Publications Company, 2017. ISBN 9781617294433. Disponível em: <https://books.google.de/books?id=Yo3CAQAACAAJ>.

CORDER, G.; FOREMAN, D. *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*. Wiley, 2011. ISBN 9781118211250. Disponível em: <https://books.google.com.br/books?id=T3qOqdpSz6YC>.

DE ALMEIDA, P. R.; OLIVEIRA, L. S.; BRITTO, A. S.; SILVA, E. J.; KOERICH, A. L. Pklot – a robust dataset for parking lot classification. *Expert Systems with Applications*, v. 42, n. 11, p. 4937–4949, 2015. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417415001086>.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255.

DOMINGOS, P.; HULTEN, G. "catching up with the data: Research issues in mining data streams.". In: *Proc. of Workshop on Research Issues in Data Mining and Knowledge Discovery*. [S.l.: s.n.], 2001.

EMC Education Services. *Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. Wiley, 2015. ISBN 9781119183686. Disponível em: <https://books.google.com.br/books?id=qoT-jgEACAAJ>.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, ASA Website, v. 32, n. 200, p. 675–701, 1937. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1937.10503522>.

GAMA, J. *Knowledge Discovery from Data Streams.* [S.l.]: CRC Press, 2010. I-XIX, 1-237 p. (Chapman and Hall / CRC Data Mining and Knowledge Discovery Series). ISBN 978-1-4398-2611-9.

GAMA, J.; MEDAS, P.; CASTILLO, G.; RODRIGUES, P. Learning with drift detection. In: BAZZAN, A. L. C.; LABIDI, S. (Ed.). *Advances in Artificial Intelligence – SBIA 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 286–295. ISBN 978-3-540-28645-5.

GAMA, J.; ŽLIOBAITĖ, I.; BIFET, A.; PECHENIZKIY, M.; BOUCHACHIA, A. A survey on concept drift adaptation. *ACM Computing Surveys*, Association for Computing Machinery (ACM), v. 46, n. 4, p. 1–37, abr. 2014. Disponível em: <https://doi.org/10.1145%2F2523813>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. (Adaptive computation and machine learning). ISBN 9780262035613. Disponível em: <https://books.google.co.in/books?id=Np9SDQAAQBAJ>.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. In: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N.; WEINBERGER, K. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. v. 27. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.

GUZY, F.; WOźNIAK, M.; KRAWCZYK, B. Evaluating and explaining generative adversarial networks for continual learning under concept drift. In: *2021 International Conference on Data Mining Workshops (ICDMW)*. [S.l.: s.n.], 2021. p. 295–303.

HOCHULI, A. G.; BRITTO, A. S.; DE ALMEIDA, P. R. L.; ALVES, W. B. S.; CAGNI, F. M. C. *Evaluation of Different Annotation Strategies for Deployment of Parking Spaces Classification Systems*. 2022.

KUNCHEVA, L. Classifier ensembles for changing environments. In: . [S.l.: s.n.], 2004. v. 3077, p. 1–15. ISBN 978-3-540-22144-9.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

MALIALIS, K.; PANAYIOTOU, C. G.; POLYCARPOU, M. M. Data-efficient online classification with siamese networks and active learning. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020. Disponível em: <https://doi.org/10.1109%2Fijcnn48605.2020.9206730>.

MALIALIS, K.; PAPATHEODOULOU, D.; FILIPPOU, S.; PANAYIOTOU, C. G.; POLYCARPOU, M. M. *Data augmentation on-the-fly and active learning in data stream classification*. 2022.

POUYANFAR, S.; TAO, Y.; MOHAN, A.; TIAN, H.; KASEB, A. S.; GAUEN, K.; DAILEY, R.; AGHAJANZADEH, S.; LU, Y.-H.; CHEN, S.-C.; SHYU, M.-L. Dynamic sampling in convolutional neural networks for imbalanced data classification. In: *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. [S.l.: s.n.], 2018. p. 112–117.

RAAB, C.; HEUSINGER, M.; SCHLEIF, F.-M. Reactive soft prototype computing for concept drift streams. *Neurocomputing*, Elsevier BV, v. 416, p. 340–351, nov. 2020. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2019.11.111>.

RADFORD, A.; METZ, L.; CHINTALA, S. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016.

SILVA, T. S. A short introduction to generative adversarial networks. *https://sthalles.github.io*, 2017. Disponível em: <https://sthalles.github.io/intro-to-gans/>.

WEBB, G. I.; HYDE, R.; CAO, H.; NGUYEN, H. L.; PETITJEAN, F. Characterizing concept drift. *Data Mining and Knowledge Discovery*, Springer Science and Business Media LLC, v. 30, n. 4, p. 964–994, apr 2016. Disponível em: <https://doi.org/10.1007%2Fs10618-015-0448-4>.

WIDMER, G.; KUBAT, M. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, Kluwer Academic Publishers, Boston, v. 23, n. 1, p. 69–101, 1996.

WU, Q.; CHEN, Y.; MENG, J. Dcgan-based data augmentation for tomato leaf disease identification. *IEEE Access*, v. 8, p. 98716–98728, 2020.