

**UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"**  
FACULDADE DE CIÊNCIAS - CAMPUS BAURU  
DEPARTAMENTO DE COMPUTAÇÃO  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GABRIEL CARDOSO FRANCO

**INVESTIGAÇÃO DE ESTRATÉGIAS DE APRENDIZADO  
CONTÍNUO PARA DETECÇÃO DE OBJETOS DE TRÁFEGO  
URBANO**

BAURU  
Novembro/2025

GABRIEL CARDOSO FRANCO

**INVESTIGAÇÃO DE ESTRATÉGIAS DE APRENDIZADO  
CONTÍNUO PARA DETECÇÃO DE OBJETOS DE TRÁFEGO  
URBANO**

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.  
Orientador: Prof. Dr. André Luis Debiasso Rossi

BAURU  
Novembro/2025

F825i

Franco, Gabriel Cardoso

Investigação de estratégias de aprendizado contínuo para detecção de objetos de tráfego urbano / Gabriel Cardoso Franco. -- Bauru, 2025  
48 p.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru

Orientador: André Luis Debiasso Rossi

1. Aprendizado contínuo. 2. Esquecimento catastrófico. 3. Detecção de objetos. 4. Redes neurais convolucionais. 5. Aprendizado profundo.

I. Título.

Gabriel Cardoso Franco

# **Investigação de estratégias de aprendizado contínuo para detecção de objetos de tráfego urbano**

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

---

**Prof. Dr. André Luis Debiasso Rossi**

Orientador

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

---

**Profa. Dra. Simone das Graças Domingues Prado**

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

---

**Prof. Dr. Kelton Augusto Pontara da Costa**

Universidade Estadual Paulista "Júlio de Mesquita Filho"

Faculdade de Ciências

Departamento de Ciência da Computação

Bauru, 14 de novembro de 2025.

*A Deus, pela bênção da vida e por conduzir meu caminho.  
Aos meus pais, pela companhia, orientação, e principalmente  
o amor incondicional de vocês.*

# Agradecimentos

Agradeço a Deus por todas as bênçãos concedidas a mim e por Sua misericórdia, que me permitiram viver esta jornada até a conquista da graduação.

Agradeço aos meus Pais, pelos dias bons e difíceis em que sempre estiveram comigo, pelas orientações e pelos puxões de orelha que me mantiveram no caminho certo e, principalmente, pelo amor e pelo carinho incondicionais.

Agradeço à minha família, principalmente aos meus avós, pelo apoio e cuidado que vocês sempre tiveram comigo e por estarem sempre comigo para celebrar as minhas vitórias.

Agradeço aos meus Amigos, sejam eles veteranos, colegas de sala, calouros, virtuais ou de longa data pela companhia nos momentos de estresse do dia a dia e pelas risadas.

Agradeço a Todos que um dia fizeram parte da minha vida, sendo com experiências boas ou ruins, que contribuíram para que eu possa ser quem eu sou hoje.

*“Referi-vos essas coisas para que tenhais paz em mim.  
No mundo haveis de ter aflições. Coragem! Eu venci o mundo.”*

João 16,33

# Resumo

Com o avanço das Redes Neurais Convolucionais (CNNs), multiplicaram-se as aplicações em modelos de linguagem, visão computacional e análise de vídeos. Apesar desse progresso, o esquecimento catastrófico continua como um problema que dificulta o aprendizado contínuo e, na prática, leva ao retreinamento do zero sempre que novas tarefas, classes ou domínios são introduzidos. Essa estratégia é cara e ineficiente, principalmente quando já existem métodos capazes de mitigar tal efeito no paradigma de aprendizado contínuo. Neste trabalho, é analisado o impacto do uso de técnicas de aprendizado contínuo no treinamento de CNNs para detecção de objetos no contexto de tráfego urbano.

**Palavras-chave:** aprendizado contínuo; esquecimento catastrófico; detecção de objetos; redes neurais convolucionais; aprendizado profundo; tráfego urbano.

# Abstract

With the advancement of Convolutional Neural Networks (CNNs), applications in language models, computer vision, and video analysis have multiplied. Despite this progress, catastrophic forgetting remains a problem that hinders continual learning and, in practice, leads to retraining from scratch whenever new tasks, classes, or domains are introduced. This strategy is costly and inefficient, especially given that methods already exist to mitigate this effect within the continual learning paradigm. In this work, we analyze the impact of employing continual learning techniques in training CNNs for object detection in the context of urban traffic.

**Keywords:** continual learning; catastrophic forgetting; object detection; convolutional neural networks; deep learning; urban traffic.

# Lista de figuras

Figura 1 – Gráfico ilustrativo de problemas de classificação. . . . .	19
Figura 2 – Gráfico ilustrativo de problemas de regressão. . . . .	20
Figura 3 – Ilustração de uma arquitetura MLP. . . . .	21
Figura 4 – Representação geral de uma Rede Neural . . . . .	22
Figura 5 – Representação do LeNet5. . . . .	23
Figura 6 – Exemplo de <i>pipeline</i> de classificação de imagens. . . . .	23
Figura 7 – Exemplo de imagens rotuladas para detecção de objetos do conjunto de dados <i>Automotive</i> . . . . .	24
Figura 8 – Exemplo de imagem segmentada. . . . .	24
Figura 9 – Exemplo da arquitetura RCNN. . . . .	25
Figura 10 – Exemplo da arquitetura Fast RCNN. . . . .	26
Figura 11 – Imagem ilustrando o IOU. . . . .	27
Figura 12 – Exemplo do funcionamento da estratégia EWC. . . . .	30
Figura 13 – Exemplo do conjunto <i>Automotive</i> . . . . .	34
Figura 14 – Exemplo do conjunto <i>Rock, Papers, Scissors</i> . . . . .	34
Figura 15 – Ilustração da biblioteca Avalanche . . . . .	36
Figura 16 – Representação da <i>pipeline</i> do EWC durante o treinamento . . . . .	38

# Lista de tabelas

Tabela 1 – Valores dos Hiperparâmetros do FastRCNN . . . . .	35
Tabela 2 – Valores do Otimizador . . . . .	35
Tabela 3 – Ferramentas e Tecnologias utilizada . . . . .	39
Tabela 4 – Avaliação no conjunto <i>Automotive</i> . . . . .	40
Tabela 5 – Avaliação no conjunto <i>Rock, Paper and Scissor</i> . . . . .	41

# Lista de abreviaturas e siglas

AM	Aprendizado de máquina
API	Interface de programação de aplicações
AP	<i>Avarage precision</i>
AR	<i>Avarage recall</i>
CIL	Aprendizado incremental de Classes ( <i>Continual incremental learning</i> )
CoPE	<i>Continual Prototype Evolution</i>
CNN	Redes neurais convolucionais
DGR	<i>Deep Generative Replay</i>
EBLL	<i>Encoder Based Lifelong Learning</i>
ER	<i>Experience Replay</i>
EWC	<i>Elastic weight consolidation</i>
FP	Falsos positivos
HAT	<i>Hard Attention to the Task</i>
IA	Inteligência artificial
IMM	<i>Incremental Moment Matching</i>
IOU	Interseção sobre União ( <i>Intersection over Union</i> )
LwF	<i>Learn without Forgetting</i>
MLP	Perceptron Multicamadas ( <i>Multilayer perceptron</i> )
mAP	<i>Medium Avarage Precision</i>
NP	Falsos negativos
PR	<i>Pseudo-Recursal</i>
RCNN	<i>Region-based Convolutional Neural Networks</i>
RPN	<i>Region Proposal Network</i>

SDG	Gradiente descendente estocástico ( <i>Stochastic Gradient Descent</i> )
SI	<i>Synaptic Intelligence</i>
SVM	Máquina de Vetores de Suporte ( <i>Support Vector Machine</i> )
TP	Verdadeiros positivos

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Problema	15
1.2	Justificativa	16
1.3	Objetivos	17
1.4	Estrutura do Trabalho	17
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>18</b>
2.1	Aprendizado de Máquina	18
2.1.1	Tipos de Aprendizado Supervisionado	19
2.2	Aprendizado Profundo	20
2.2.1	Rede Neural Convolucional (CNN)	21
2.2.2	Redes Neurais Convolucionais em Visão Computacional	22
2.2.3	Redes Neurais Convolucionais em Detecção de Objetos	25
2.2.3.1	<i>Region-based Convolutional Neural Networks</i>	25
2.2.3.2	<i>Fast Region-based Convolutional Neural Networks</i>	25
2.2.3.3	Métricas de Avaliação	26
2.3	Aprendizado Contínuo	27
2.3.1	Estratégias de Aprendizado Contínuo	28
2.3.1.1	Estratégia de <i>Replay</i>	29
2.3.1.2	Estratégia de Regularização	29
2.3.1.3	Estratégias de Isolamento de Parâmetros	31
2.3.2	Métricas de Avaliação	31
<b>3</b>	<b>MATERIAL E MÉTODOS</b>	<b>33</b>
3.1	Sumário da Metodologia	33
3.2	Base de Dados	33
3.3	Modelo e Hiperparâmetros	34
3.4	Aprendizado Contínuo	35
3.4.1	Avalanche	36
3.4.2	<i>Replay</i>	36
3.4.3	Elastic Weight Consolidation	37
3.5	Métricas avaliativas	37
3.5.1	Detecção de Objetos	37
3.5.2	Aprendizado Contínuo	39
3.6	Ferramentas e Tecnologias	39

<b>4</b>	<b>RESULTADOS</b> . . . . .	<b>40</b>
<b>4.1</b>	<b>Desempenho da Detecção de Objetos</b> . . . . .	<b>40</b>
4.1.1	Desempenho das técnicas de Aprendizado Contínuo . . . . .	41
<b>5</b>	<b>CONCLUSÃO</b> . . . . .	<b>42</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>43</b>

# 1 Introdução

Atualmente, a Inteligência Artificial (IA) vem sendo adotada de forma ampla em diversos setores, impulsionada por ganhos comprovados de eficiência, escalabilidade e capacidade preditiva. Seus impactos já são observados em áreas como saúde (HAMET; TREMBLAY, 2017), agricultura (LIAKOS et al., 2018) e planejamento urbano (HUANG; KOROTEEV, 2021).

Esse movimento alcança também a mobilidade urbana, isso se torna visível com os sistemas de transporte inteligentes que podem ser entendidos como o conjunto de tecnologias e aplicações voltadas a monitorar, prever e gerenciar o tráfego, com vistas a melhorar a segurança, a fluidez e a qualidade do serviço prestado à população. Nesse contexto, modelos de aprendizado de máquina (AM) têm sido empregados em tarefas como previsão de demanda por táxis (YAO et al., 2018), estimativa de fluxo de pedestres e multidões (ZHANG; ZHENG; QI, 2017) e controle adaptativo de semáforos (BOUKTIF et al., 2023). Tais aplicações, isoladas ou integradas em plataformas maiores, demonstram potencial para elevar significativamente a eficiência operacional e a experiência do usuário na mobilidade urbana.

No contexto de visão computacional, as Redes Neurais Convolucionais (CNN) têm liderado as aplicações por aliarem alta capacidade de extração de padrões a boa eficiência computacional (ZHAO et al., 2024). Inspiradas na organização hierárquica do córtex visual, exploram campos receptivos locais, *compartilhamento de pesos* e a composição progressiva de características para aprender representações cada vez mais abstratas das imagens (LI et al., 2021). Conseqüentemente, as CNN também passaram a ser amplamente utilizadas para a tarefa de detecção de objetos como padrão (ZHAO et al., 2019). Esses modelos aprendem a reconhecer padrões visuais que identificam cada objeto e, ao mesmo tempo, a indicar em que parte da imagem ele aparece, o que é eficaz e suficientemente rápida para uso prático mesmo com variações de iluminação, ângulos e tamanhos, cenário frequentemente encontrado no tráfego urbano.

## 1.1 Problema

Mesmo com os resultados eficientes, a maioria das pesquisas mencionadas anteriormente foram desenvolvidas considerando cenários estáticos, nos quais se pressupõe que os dados não sofram alterações significativas ao longo do tempo. Essa limitação caracteriza o chamado Cenário Fechado, no qual, uma categoria desconhecida a priori é obrigatoriamente alocada a um rótulo existente, o que, ao tentar expor o sistema a um novo domínio, classe ou tarefa, gera inconsistências nos resultados de predição.

As CNN, embora tenham alcançado o estado da arte em diversas tarefas, enfrentam

desafios no aprendizado contínuo. Isso ocorre porque o impacto das novas informações depende de sua relação com o que a rede já aprendeu. Quando as novas informações são semelhantes ou complementares às já aprendidas, a rede tende a enriquecer seu conhecimento. Por outro lado, se forem significativamente diferentes, pode haver conflito, levando à substituição ou perda do conhecimento anteriormente adquirido (CHEN; LIU, 2018). Esse problema é conhecido como Esquecimento Catastrófico (*Catastrophic Forgetting*) (ZENKE; POOLE; GANGULI, 2017a), em que o conhecimento previamente adquirido é perdido.

O Cenário Aberto consiste em uma especificação do aprendizado contínuo e, conseqüentemente, aborda também a lacuna das “classes não vistas”, permitindo que os modelos reconheçam novas classes conforme elas surgem. Esse cenário de aprendizado contínuo, conhecido como aprendizado incremental de classes (CIL - *Class Incremental Learning*) é melhor descrito como uma sequência de tarefas baseadas em classificação em que cada tarefa contém classes diferentes e o sistema deve aprender a distinguir entre as classes disponíveis (VEN; TUYTELAARS; TOLIAS, 2022a).

Diante disso, o paradigma de CIL apresenta-se como uma abordagem promissora para aplicação em CNNs no contexto de tráfego urbano, uma vez que permite a esses algoritmos adaptarem-se à realidade dinâmica do dia a dia. Contudo, conforme destacado por Yang et al. (2024), o cenário de aprendizado incremental de classes ainda é um tema emergente, com poucos estudos voltados ao conceito de cidades inteligentes. Diante disso, o presente projeto de pesquisa tem como objetivo aplicar e comparar técnicas de aprendizado incremental de classes em redes neurais profundas, especificamente no domínio de cidades inteligentes, buscando contribuir para o avanço da área e explorar seu potencial em um ambiente urbano e dinâmico.

## 1.2 Justificativa

A detecção de objetos em ambientes urbanos é essencial para sistemas de transporte inteligentes, com aplicações em veículos autônomos, vigilância, planejamento de tráfego e segurança pública. Entretanto, o cenário urbano é dinâmico: novos modelos de veículos, alterações de infraestrutura viária, variações de iluminação e clima, além de mudanças nos sensores, causam instabilidade e reduzem progressivamente o desempenho de modelos treinados. Reexecutar todo o ciclo de treinamento a cada mudança é custoso, lento e, em muitos casos, inviável operacionalmente, o que limita a adoção desses sistemas em ambientes reais.

Diante desse contexto, o CIL capaz de incorporar novas informações sem perder conhecimento previamente adquirido, mitigando o esquecimento catastrófico. Ao permitir reduzir custos operacionais, tais estratégias tendem a aumentar a robustez e a longevidade de modelos de visão computacional implementados em tráfego urbano dinâmico. Assim, este estudo se justifica tanto pelo potencial impacto prático na melhoria de eficiência e qualidade dos serviços de mobilidade quanto pela contribuição científica à área emergente de aprendizado

contínuo aplicado a cidades inteligentes.

## 1.3 Objetivos

O objetivo geral do projeto é investigar a aplicação de técnicas de aprendizado contínuo para lidar com o problema de identificação de objetos relacionado ao tráfego de veículos e pedestres no contextos de cidades inteligentes. Neste projeto são definidos os seguintes objetivos específicos:

1. Coletar e pré-processar imagens de tráfego de veículos e pedestres para a aplicação de métodos de CIL;
2. Aplicar e refinar algoritmos de CNN para identificação de veículos e pedestres;
3. Investigar diferentes estratégias e métodos de CIL.
4. Investigar o desempenho de diferentes métodos de CIL para as imagens processadas.

## 1.4 Estrutura do Trabalho

Este trabalho está dividido em 5 capítulos. O Capítulo 1 **Introdução** contextualiza a IA e sua aplicação no tráfego urbano, apresenta o paradigma de aprendizado contínuo, delimita o problema de pesquisa, justifica sua relevância e explicita os objetivos geral e específicos. O Capítulo 2 **Fundamentação Teórica** reúne conceitos, definições e trabalhos relacionados que sustentam a pesquisa, estabelecendo a base teórica para as escolhas metodológicas dessa pesquisa. O Capítulo 3 **Material e Métodos** descreve o conjunto de dados, os procedimentos de coleta e pré-processamento, os métodos e as métricas de avaliação, além das ferramentas e configurações utilizadas. O Capítulo 4 **Resultados** apresenta e discute os achados obtidos com os métodos de aprendizado contínuo, comparando com abordagens de referência e analisando implicações, limitações e possíveis ameaças à validade. O Capítulo 5 **Conclusão** sintetiza as contribuições do trabalho e indica direções para pesquisas futuras e aplicações práticas.

## 2 Fundamentação Teórica

Este capítulo visa descrever os conceitos e técnicas necessários para o desenvolvimento deste trabalho.

### 2.1 Aprendizado de Máquina

AM é uma subárea de IA que busca o desenvolvimento de algoritmos que usam dados de entrada para executar uma tarefa sem depender de regras manuais para cada caso. Em vez disso, o sistema é ajustado automaticamente por meio da experiência. Durante o treinamento, recebe exemplos acompanhados de saídas desejadas e adapta sua configuração interna para reduzir o erro (NAQA; MURPHY, 2015).

Essa área vem sendo aplicado nos mais diferentes domínios nos últimos anos. Por exemplo, modelos como *Naive Bayes*, Máquina de Vetores de Suporte (SVM) para mapear padrões de imagem em comandos (SHINDE; SHAH, 2018). Em segurança da informação, sistemas de detecção de intrusão utilizam algoritmos como SVM e *Random Forest* para classificar tráfego de rede, avaliando desempenho por precisão, revocação e F1. Na área da saúde, a detecção e o diagnóstico de câncer de mama empregam SVM, Redes Bayesianas e *Random Forest* (SHINDE; SHAH, 2018). Dessa forma, esses modelos são classificadas em 3 tipos: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço (BURKOV, 2020).

- No aprendizado supervisionado trabalha-se com um conjunto de exemplos rotulados  $\{(x_1, y_1), \dots, (x_N, y_N)\}$ . Cada  $x_i$  é um vetor de atributos de dimensão  $D$ , em que a  $j$ -ésima posição contém sempre o mesmo tipo de informação para todos os exemplos. O rótulo  $y_i$  pode ser uma classe, quando o problema é de classificação, ou um número real, quando o problema é de regressão. O objetivo é aprender uma função  $f$  que mapeie  $x$  em  $y$  e *generalize* para dados não vistos.
- No aprendizado não supervisionado, trabalha-se com um conjunto de exemplos *sem rótulos*  $\{x_1, x_2, \dots, x_N\}$ , em que cada  $x_i$  é um vetor de características. O objetivo é induzir um modelo que, dado  $x$ , produza uma transformação útil ou um valor que ajude a resolver um problema prático, descobrindo estrutura nos dados sem supervisão explícita.
- No aprendizado por reforço, um agente interage com um ambiente observando seu estado e escolhe ações em estados não terminais e recebe recompensas que dependem da ação tomada que podem levar a novos estados. O objetivo é aprender uma política ótima, isto é, uma função que mapeia estados em ações maximizando a recompensa esperada

de longo prazo. Diferentemente do aprendizado supervisionado, a decisão é sequencial e o *feedback* é retardado: escolhas atuais afetam recompensas futuras. Modelos de RL abordam o dilema do quanto deve explorar optando entre tentar ações novas para coletar informação ou repetir ações conhecidas que rendem boa recompensa. São tipicamente formulados como Processos de Decisão de Markov.

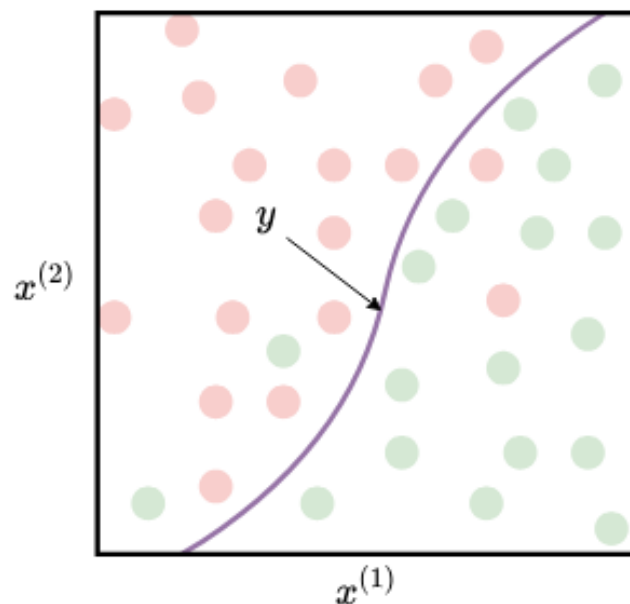
Neste trabalho foram empregados algoritmos de aprendizado supervisionado. Por esse motivo, os tipos de aprendizado supervisionado serão apresentados e discutidos a seguir contextualizando as técnicas utilizadas e esclarecendo suas particularidades dentro do escopo da pesquisa.

### 2.1.1 Tipos de Aprendizado Supervisionado

O aprendizado supervisionado pode ser subdividido em duas categorias principais: classificação e regressão. A diferença se encontra no tipo de variável a ser prevista e, por consequência, na forma do modelo e na avaliação.

- Na classificação, o objetivo é atribuir uma categoria discreta a cada exemplo. Dessa forma, o modelo aprende um hiperplano que separa regiões do espaço de atributos por classe e costuma produzir probabilidades por classe ou diretamente o rótulo mais provável. Na figura 1 o hiperplano busca uma forma ótima de separar as 2 classes.

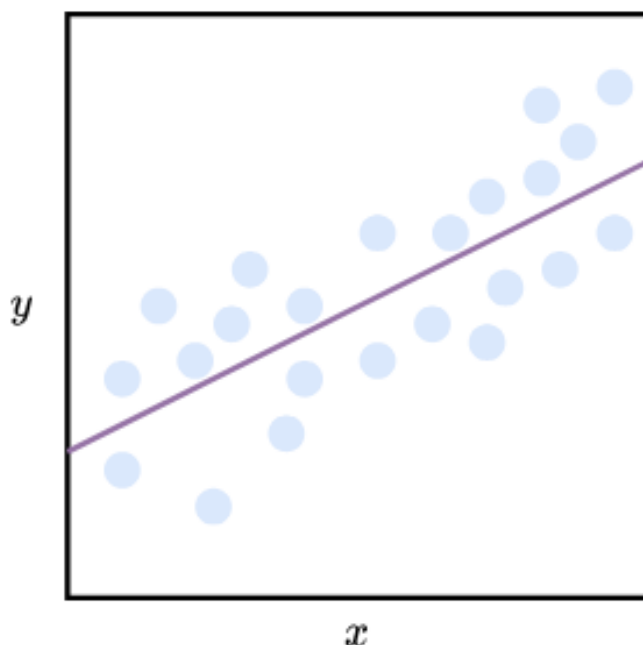
Figura 1 – Gráfico ilustrativo de problemas de classificação.



Fonte: Adaptado de [Burkov \(2020\)](#)

- Na regressão, o alvo é um valor contínuo ( $y \in \mathbb{R}$ ); o modelo aprende uma função  $f : \mathbb{R}^D \rightarrow \mathbb{R}$  que aproxima a relação entre os atributos  $x$  e o alvo  $y$  minimizando o erro. No caso da figura 2 é representado um hiperplano em um modelo linear.

Figura 2 – Gráfico ilustrativo de problemas de regressão.



Fonte: Adaptado de [Burkov \(2020\)](#)

## 2.2 Aprendizado Profundo

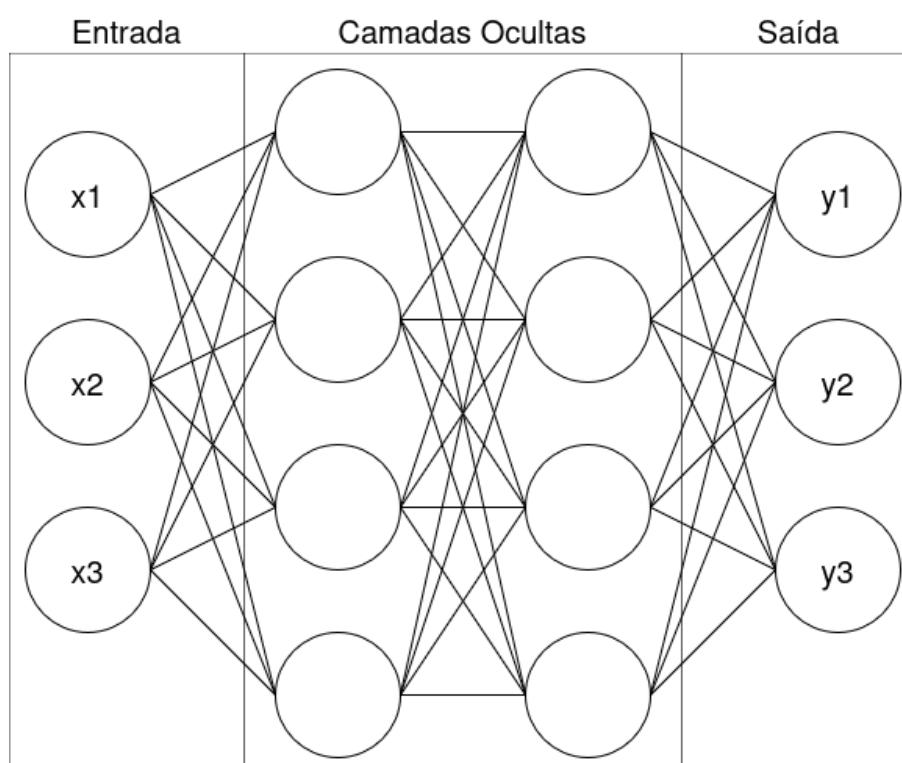
Em AM, parte-se do pressuposto de que os problemas a serem resolvidos não apresentem muitos fatores de variação que afetem simultaneamente cada observação, e caso isso ocorra, é necessário abstrações de alto nível para cada caso em que houver essa variação. Na prática, isso exige grandes volumes de dados, estratégias cuidadosas de pré-processamento e tratamento do conjunto de treino para atenuar tais limitações ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)).

Essa limitação motivou o desenvolvimento da abordagem de aprendizado profundo, subárea de AM que permite aos sistemas aprenderem a partir da experiência, organizando o conhecimento em hierarquias de conceitos distribuídas ao longo de múltiplas camadas ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)). Diferentemente de técnicas que dependem de atributos manuais, o foco está em aprender representações diretamente dos dados, reduzindo a necessidade de especificar antes todo o conhecimento que o modelo precisa.

Os *perceptrons*, inspirados no funcionamento dos neurônios biológicos ([MCCULLOCH; PITTS, 1943](#)), foram propostos como uma abordagem em que não é mais necessário definir

manualmente cada atributo que o modelo deve aprender, delegando ao próprio algoritmo a tarefa de ajustar pesos e extrair características relevantes a partir dos dados. O *perceptron* é generalista pois o modelo artificial mapeia sua entrada e saída ajustando seus pesos e viés durante a otimização de uma função de perda. O acúmulo desses *perceptrons* gera o Perceptron Multicamadas (MLP), ilustrada na figura 3 que consiste em uma camada de entrada, uma de saída e qualquer quantidade de camadas ocultas. Apenas um neurônio sozinho não é capaz de representar modelos não-lineares (HAYKIN, 2009), entretanto, com a arquitetura MLP essas representações se tornam possíveis.

Figura 3 – Ilustração de uma arquitetura MLP.



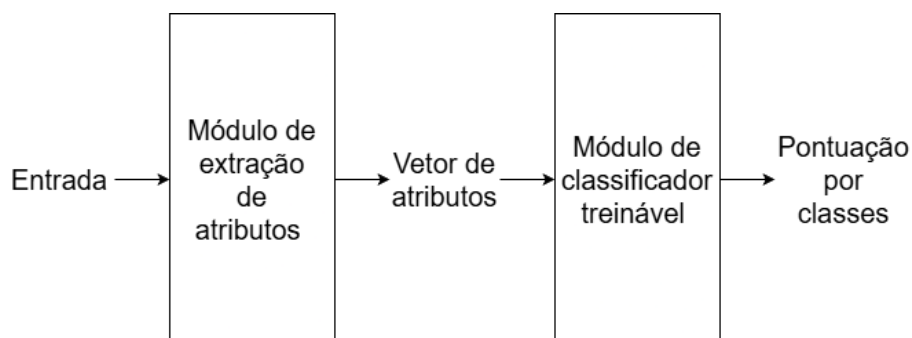
Fonte: Autoria Própria

### 2.2.1 Rede Neural Convolutacional (CNN)

CNN são ideias para lidar com a variabilidade de figuras bidimensionais e apresentam resultados melhores que outros algoritmos nessas tarefas (LECUN et al., 1998). Uma definição viável para CNN são um conjunto de redes neurais que usam convoluções em pelo menos uma das suas camadas (GOODFELLOW; BENGIO; COURVILLE, 2016) que de forma geral possuem dois objetivos: extração de atributos e classificação, representado na figura 4.

De modo geral uma CNN é organizada em 5 camadas: entrada, convolução, pooling, totalmente conectada e saída. A camada de entrada é definida pela natureza desses dados.

Figura 4 – Representação geral de uma Rede Neural



Fonte: Adaptado de [Lecun et al. \(1998\)](#)

Após isso, a camada convolucional é responsável por organizar os parâmetros em filtros (kernels). A convolução, um filtro de suporte finito é transladado sobre o volume de entrada em todas as posições válidas o chamado varrimento por “janela deslizante” (*sliding window*) garantindo que o filtro permaneça totalmente contido no domínio considerado. Em cada posição, extrai-se da entrada um subvolume com formato idêntico ao do filtro onde ambos são aplainados para formar dois vetores de mesma dimensão. O valor local da operação resulta do produto interno desses vetores, isto é, da soma das multiplicações elemento a elemento ([AGGARWAL, 2018](#)).

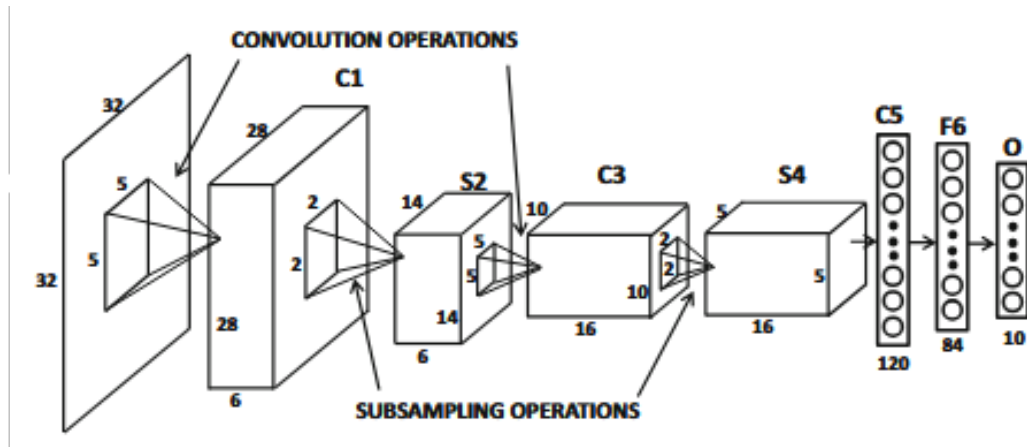
Já o *pooling* é uma operação que atua separadamente em mapa de ativação para reduzir as dimensões espaciais enquanto preserva a profundidade. Define-se uma janela local e um *stride*, a cada posição válida, aplica-se um operador de agregação sobre os valores contidos na janela, produzindo um único valor e, assim, realizando *downsampling*, diminuindo custo computacional e uso de memória ([AGGARWAL, 2018](#)). Por fim, a classificação é realizada por um camada totalmente conectado, de arquitetura análoga a uma MLP, que mapeia as características extraídas para as pontuações de cada classe. Na camada de saída, aplica-se tipicamente *softmax* para converter esses escores em probabilidades e definir a predição final.

Na figura 5, é ilustrada a arquitetura LeNet5, uma das primeiras arquiteturas desenvolvidas. Nessa arquitetura, são utilizadas duas camadas convolucionais (C1, C3) intercaladas por duas camadas de *pooling* (S2, S4) e por fim, 3 camadas totalmente conectadas (C5, F6, 0).

## 2.2.2 Redes Neurais Convolucionais em Visão Computacional

A visão computacional é uma subárea de IA que visa permitir a inferência de uma imagem a partir de um modelo que pode extrair, representar ou interpretar informações de imagens e vídeos, automatizando tarefas que, para humanos, dependem da percepção visual. Nos últimos anos, a área de visão computacional ganhou muita visibilidade principalmente pela disponibilidade de processamento, memória e armazenamento computacional disponível ([PRINCE, 2012](#)). Por consequência, sua aplicação foi gerando diversos cenários explorados pela

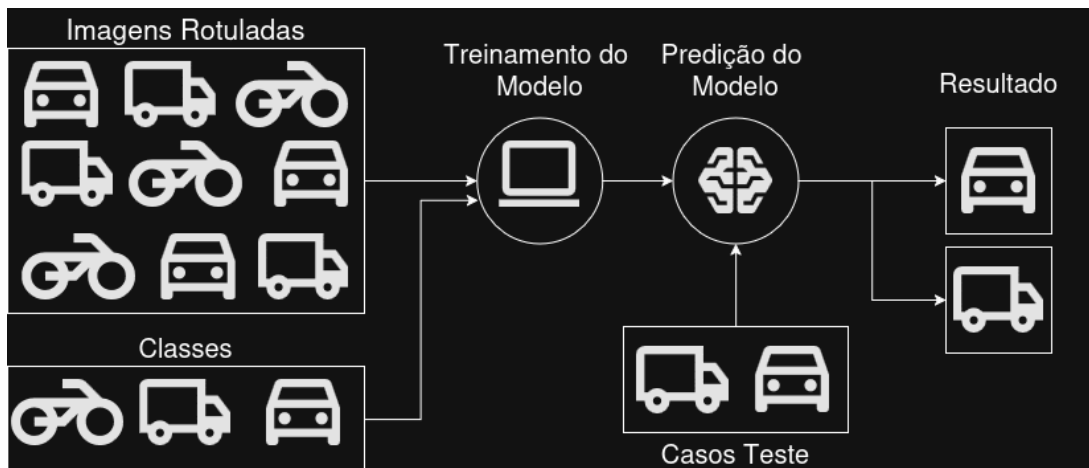
Figura 5 – Representação do LeNet5.



Fonte: Adaptado de [Aggarwal \(2018\)](#)

área como é o caso da classificação, detecção de objetos, segmentação.

- Na tarefa de *classificação de imagens*, treina-se um modelo supervisionado a partir de um conjunto de dados rotulado, no qual cada exemplo está associado a uma entre as classes disponíveis. Na etapa de predição, o modelo recebe imagens não rotuladas e deve atribuir a classe mais provável. A Figura 6 ilustra um fluxo típico dessa *pipeline*.

Figura 6 – Exemplo de *pipeline* de classificação de imagens.

Fonte: Autoria Própria

- A detecção de objetos é uma tarefa de visão computacional que identifica e localiza instâncias em imagens, retornando uma rótulo, responsáveis por classificar o objeto e caixas delimitadoras que determina onde o objeto está. Ao combinar classificação e localização em um único processo, ela fornece um mapa dos elementos relevantes da cena como representado na figura 7.

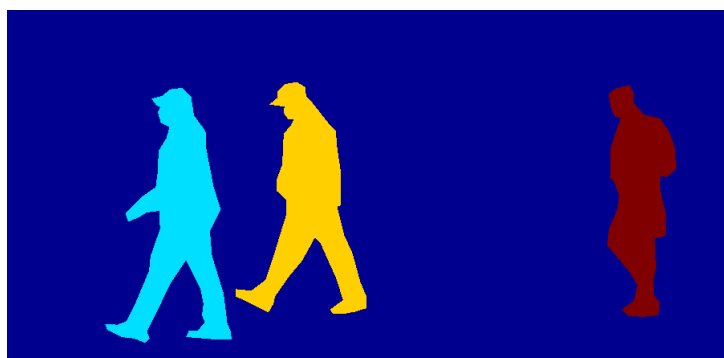
Figura 7 – Exemplo de imagens rotuladas para detecção de objetos do conjunto de dados *Automotive*



Fonte: Autoria Própria

- A segmentação semântica tem como objetivo atribuir, a cada pixel da imagem, um rótulo pertencente a um conjunto finito de classes (SHOTTON et al., 2009). Isso tem como objetivo produzir um mapa denso de classes, permitindo entender o que está presente e onde em nível de pixel. Modelos de segmentação combinam pistas locais com contexto espacial, e frequentemente aplicam etapas de refinamento para suavizar fronteiras e coerência regional, demonstrado na figura 8.

Figura 8 – Exemplo de imagem segmentada.



Fonte: <[https://www.cis.upenn.edu/~jshi/ped\\_html/](https://www.cis.upenn.edu/~jshi/ped_html/)>. Acesso em: 20 out de 2025.

Nesta pesquisa, as CNN foram empregadas exclusivamente em tarefas de detecção de

objetos. Por este motivo, este t pico ser  aprofundado a seguir, detalhando suas principais arquiteturas e m tricas de avalia o

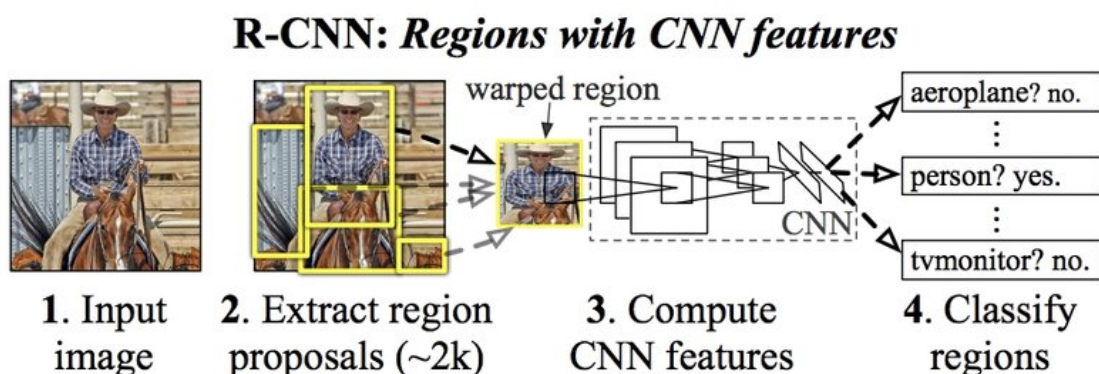
### 2.2.3 Redes Neurais Convolucionais em Detec o de Objetos

No in cio dos anos 2000, m todos heur sticos eram popularmente utilizados para problemas de reconhecimento visual, como foram os casos do *Scale Invariant Feature Transform* e o *Histograms of oriented gradients* (LOWE, 2004; DALAL; TRIGGS, 2005). Com o avan o das CNN, principalmente com a cria o do (DENG et al., 2009), o uso dessas arquiteturas come ou se tornou o padr o para problemas de classifica o e detec o de objetos.

#### 2.2.3.1 Region-based Convolutional Neural Networks

O *Region-based Convolutional Neural Networks* (RCNN) foi a CNN pioneira em detec o de objetos. Essa arquitetura introduz um fluxo em duas etapas para detec o de objetos gerar propostas de regi es utilizando de *Selective Search* (UIJLINGS et al., 2013) e para cada regi o, recortar/normalizar, extrair caracter sticas com uma CNN. A etapa de classifica o   feita com SVMs, esse processo est  representado na figura 9. O m todo inaugurou o uso efetivo de em detec o, mas   lento por causa da velocidade do processo de infer ncia, consome muito armazenamento (caracter sticas em disco) e requer treinamento em m ltiplos est gios (GIRSHICK et al., 2014).

Figura 9 – Exemplo da arquitetura RCNN.



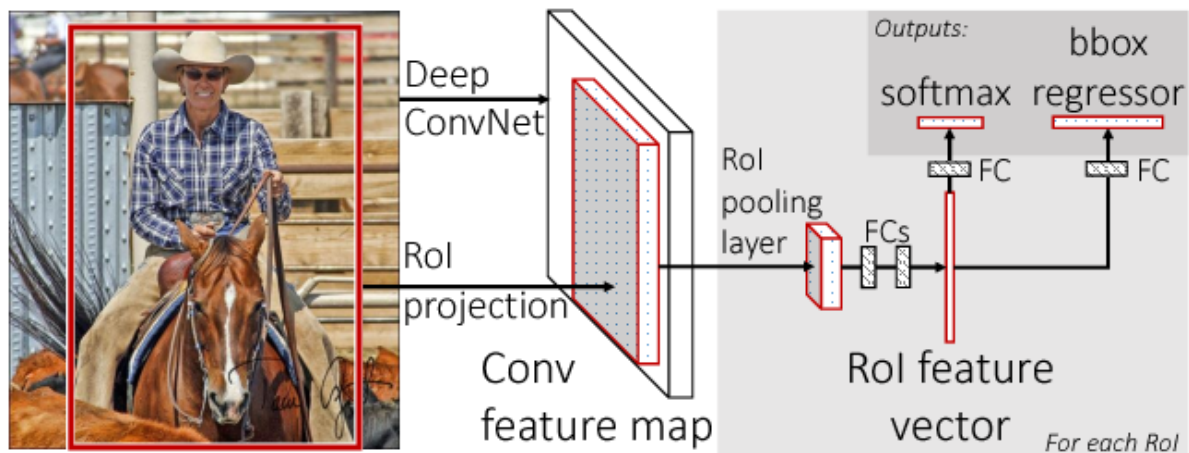
Fonte: Adaptado de Girshick et al. (2014)

#### 2.2.3.2 Fast Region-based Convolutional Neural Networks

O *Fast RCNN*   um sucessor do RCNN com algumas diferen as com o objetivo de acelerar e simplificar a RCNN original ao computar os mapas convolucionais da imagem apenas uma vez e, a partir deles, extrair descri es para cada regi o proposta por meio do *RoI*

*Pooling*. Sobre essas regiões alinhadas, uma camada totalmente conectada realiza, de forma conjunta, a classificação das instâncias via *softmax* e o refinamento das caixas delimitadoras por meio da regressão, permitindo treinamento com uma perda multi-tarefa. Essa estratégia reduz drasticamente o custo e o uso de armazenamento do RCNN, mantendo alta precisão; seu principal gargalo remanescente é a dependência de um gerador externo de propostas posteriormente integrado no *Faster RCNN* com a *Region Proposal Network* (RPN) (GIRSHICK, 2015). Com exceção da RPN, essa sequência pode ser observada na figura 10.

Figura 10 – Exemplo da arquitetura Fast RCNN.



Fonte: Adaptado de Girshick (2015)

### 2.2.3.3 Métricas de Avaliação

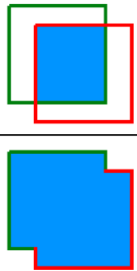
A métrica Intersecção sobre União (IOU) compara a área de sobreposição entre a caixa predita e a anotação de referência com a área de união dessas caixas, produzindo um escore entre 0 e 1. Derivada do coeficiente de Jaccard amplamente discutido na literatura (SALTON; MCGILL, 1983), representado na equação 2.1 onde  $B_{pred}$  são as coordenadas previstas e  $B_{gt}$  são as coordenadas certas.

$$\text{Coeficiente de Jaccard} = \frac{\text{area}(B_{pred} \cap B_{gt})}{\text{area}(B_{pred} \cup B_{gt})}. \quad (2.1)$$

A IOU, consolidada e popularizada como critério de avaliação em detecção de objetos pelas competições PASCAL VOC (EVERINGHAM et al., 2010), calcula sua métrica dividindo a união entre a intersecção da área prevista e a área certas pela união de ambos, representado na figura 11.

O valor resultante representa o quanto da porcentagem de acerto da predição do modelo foi um verdadeiro positivo. As duas métricas principais para comparar modelos de detecção são as *average precision* (AP) e *average recall* (AR), a primeira trata da razão entre os

Figura 11 – Imagem ilustrando o IOU.

$$\text{IOU} = \frac{\text{área de intersecção}}{\text{área de união}} = \frac{\text{Imagem 1}}{\text{Imagem 2}}$$


Fonte: Adaptado de [Padilla, Netto e Silva \(2020\)](#)

verdadeiros positivos divididos pela soma dos verdadeiros positivos (TP) e os falsos positivos (FP), representado pela equação 2.2, e o segundo representa a razão entre os verdadeiros positivos divididos pela soma dos falsos negativos (NP) mais os verdadeiros positivos, representado na equação 2.3.

$$\text{AP} = \frac{TP}{TP + FP} = \frac{TP}{\text{todas as detecções}}, \quad (2.2)$$

$$\text{AR} = \frac{TP}{TP + NP} = \frac{TP}{\text{todas as ground truths}}, \quad (2.3)$$

## 2.3 Aprendizado Contínuo

O aprendizado contínuo é a capacidade de um sistema de incorporar, ao longo do tempo, novas tarefas, classes ou domínios evitando a degradação do desempenho nas competências previamente adquiridas ([THRUN, 1995](#)). Em outras palavras, o sistema aprende de modo incremental, preservando e reutilizando conhecimento passado e mitigando o chamado esquecimento catastrófico ([ZENKE; POOLE; GANGULI, 2017a](#)), ao equilibrar plasticidade e estabilidade. Neste contexto, plasticidade refere-se à capacidade do modelo de se adaptar rapidamente a novas tarefas, classes ou domínios, modificando seus parâmetros para incorporar conhecimento recente, já a estabilidade diz respeito à habilidade de manter o desempenho nas tarefas antigas, evitando que atualizações sucessivas sobrescrevam o que foi aprendido anteriormente. Essa propriedade é essencial em ambientes dinâmicos, nos quais a distribuição dos dados muda com o tempo e seria inviável reinicializar o treinamento a cada atualização.

Com o crescimento da quantidade de dados disponíveis, modelos de ML precisam ser retreinados para se adaptarem às novas informações. Dependendo da quantidade de dados no qual um modelo é treinado, se torna inviável o retreinamento desses modelos para toda nova informação disponível. Isso faz com que o aprendizado contínuo ganhe mais notoriedade nos últimos anos ([REBUFFI et al., 2017](#)).

Para este trabalho, uma tarefa será considerado um problema específico que o modelo deve resolver. No contexto de tráfego urbano, um problema seria a identificação de sinais de trânsito, de veículos ou placas de carro. Esse modelo precisa ser treinado sequencialmente em experiências diferentes. Uma experiência é definido como um passo incremental que define o que o modelo irá ser exposto na etapa em que ele estiver (LOMONACO et al., 2021).

Na literatura, geralmente os dados são introduzidos de três maneiras (LOMONACO; MALTONI, 2017):

- **Novas Instâncias:** novos padrões das classes já disponíveis em experiências anteriores.
- **Novas Classes:** novas classes não ainda disponíveis em experiências passadas.
- **Novas Instâncias e Classes:** novos padrões das classes já disponíveis e novas classes não disponíveis em experiências passadas

Como aprendizado contínuo é uma área de conhecimento recente, existe uma variedade de protocolos utilizados para avaliar esses modelos. Por esse motivo, adota-se a convenção proposta por Ven, Tuytelaars e Tolia (2022a), na qual os cenários de aprendizado contínuo são organizados segundo dois critérios: se a identificação da tarefa está disponível durante o treinamento e se essa identificação precisa ser inferida pelo próprio modelo. A partir desses critérios, os cenários são agrupados em três categorias, apresentadas a seguir.

- **Aprendizado Incremental de Tarefas:** neste cenário os modelos sempre são informados sobre qual tarefa precisa ser feita pelo seu identificador, sendo considerada um dos cenários mais fáceis. Como seu identificador sempre é dado, é possível treinar modelos com modelos específicos para a tarefa.
- **Aprendizado Incremental de Domínio:** neste cenário o identificador da tarefa não é disponibilizado durante os testes. O modelo apenas precisam resolver a tarefa imediata, sem inferir com qual tarefa está trabalhando. Por exemplo, um agente que aprende a identificar veículos no tráfego urbanos em diferentes climas e momentos do dia, como explorado em Verwimp et al. (2023).
- **Aprendizado Incremental de Classes:** neste cenário o modelo precisa ser capaz de resolver e inferir a tarefa que está sendo resolvida, sendo o problema mais comum encontrado no dia a dia.

### 2.3.1 Estratégias de Aprendizado Contínuo

A estratégia de aprendizado contínuo representa como cada experiência será processada, definindo o ciclo de treino e validação em cada etapa, quais perdas e métricas são usadas, como os dados são amostrados, de que modo os parâmetros são atualizados garantindo o

treino de forma consistente entre no conjunto de dados, modelo e memória. Dessa forma, as estratégias são divididas em três categorias: *replay*, regularização e isolamento de parâmetros (LANGE et al., 2021).

#### 2.3.1.1 Estratégia de *Replay*

A estratégia de *replay*, também conhecido como *rehearsal*, procuram mitigar o esquecimento armazenando amostras brutas de experiências anteriores e as reprisando enquanto se aprende uma nova tarefa. Em sua forma explícita, o treino é feito com um *subconjunto limitado* de exemplos antigos misturados aos dados correntes; por construção, o desempenho dessas estratégias é majorado (limitado superiormente) pelo treino conjunto ideal, ou seja, o treino em que todas as todas as imagens são disponibilizadas desde o início em todas as tarefas de uma vez. Entre os representantes mais influentes, estão o iCaRL, *Experience Replay* (ER) e o *Continual Prototype Evolution* (CoPE) (REBUFFI et al., 2017; ROLNICK et al., 2019; LANGE; TUYTELAARS, 2021).

Quando não é possível armazenar amostras antigas, o *pseudo-rehearsal* (PR) propõe aproximar dados passados usando as saídas de modelos anteriores (ROBINS, 1995). Isso foi utilizado de forma efetiva em redes rasas, entretanto, em redes profundas e entradas de alta dimensionalidade entradas aleatórias não cobrem adequadamente o espaço de dados. Com o avanço de modelos gerativos, tornou-se possível criar pseudoexemplares mais realistas, aproximando a distribuição que gerou os dados e permitindo reentrenar o modelo atual sem guardar amostras brutas (GOODFELLOW et al., 2020). Além do PR, outro representante importante é o *Deep Generative Replay* (DGR) (SHIN et al., 2017).

#### 2.3.1.2 Estratégia de Regularização

Para evitar o uso excessivo de memória adicional dos casos de *replay*, foram criados outras estratégias, como a regularização. Essas estratégia buscam prevenir que os parâmetros das CNN se desviem do que já foi aprendido ao serem treinados em novas tarefas. Essas estratégias pode ser dividido em dois grupos principais.

Nas estratégias de regularização focados em destilação de conhecimento (*knowledge distillation*), a ideia é usar as saídas dos modelos treinados com experiências anteriores para ensinar um novo modelo, podendo ser descrito como um modelo professor que ensina um modelo aluno. Entre os mais relevantes estão o *Learn without Forgetting* (LwF) e o *Encoder Based Lifelong Learning* (EBLL) (LI; HOIEM, 2018; RANNEN et al., 2017).

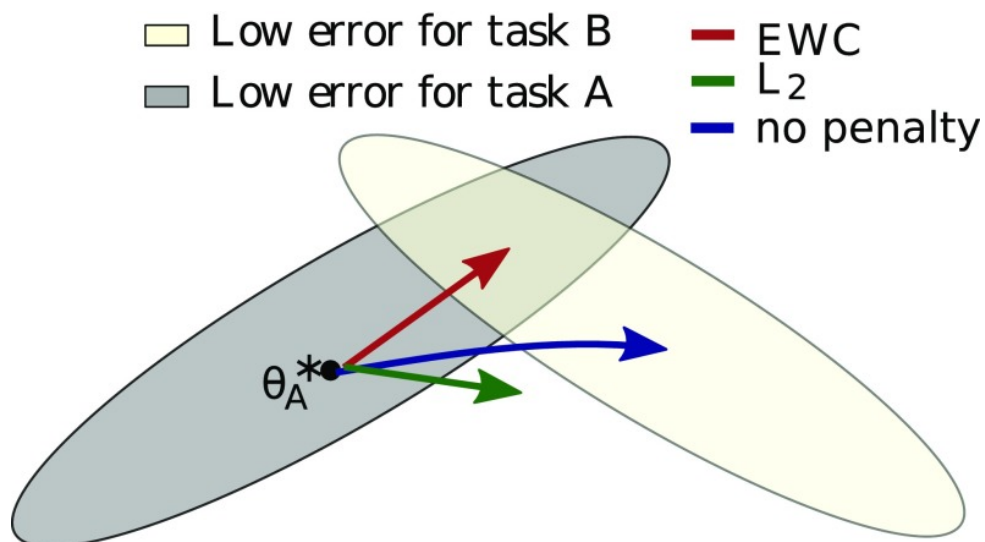
Além dessas, existem as estratégias de regularização na distribuição a priori, nas quais se estima uma distribuição sobre os parâmetros ao final de cada etapa e essa informação é reutilizada ao aprender com novos dados. Em termos práticos, atribui-se uma *importância* a cada parâmetro e penalizam-se alterações significativas justamente naqueles considerados mais relevantes nas tarefas anteriores. O *Elastic Weight Consolidation* (EWC) foi o primeiro a

estabelecer esse conceito (MCRAE, 1993), seguido por métodos igualmente relevantes, como o *Synaptic Intelligence* (SI) e o *Incremental Moment Matching* (IMM) (ZENKE; POOLE; GANGULI, 2017b; LEE et al., 2017).

Como o *Elastic Weight Consolidation* foi utilizado nesta pesquisa, ele será mais desenvolvido no seu funcionamento. A ideia principal do *Elastic Weight Consolidation* é simular o funcionamento de consolidação sináptica da memória, processo esse que faz as sinapses do cérebro mais duráveis. Isso é feito implementando um algoritmo que opera de forma similar nas CNN em que a plasticidade de parâmetros importantes são reduzidas para não perderem seus valores antigos (MCRAE, 1993).

Para aprender uma tarefa significa ajustar os pesos e vieses, representado por  $\theta$ , do modelo para maximizar o desempenho. Diferentes configurações de  $\theta$  podem alcançar desempenhos semelhantes (HECHT-NIELSEN, 1989), o que torna possível existir, para uma nova tarefa  $B$ , uma solução próxima daquela encontrada para a tarefa  $A$ . O EWC utiliza desse fato para treinar em  $B$ , com uma restrição para que os parâmetros permaneçam numa região minimizando o erro para  $A$ , centrada na solução anterior ao redor de  $\theta_A^*$ , como representado na figura 12. Essa restrição é implementada como uma penalização quadrática adicionada à perda atual, com isso, o EWC equilibra estabilidade e plasticidade sem precisar congelar completamente a rede nem armazenar dados antigos.

Figura 12 – Exemplo do funcionamento da estratégia EWC.



Fonte: Adaptado de (MCRAE, 1993)

O EWC adota uma visão *probabilística* do treinamento das CNN. Sob o ponto de vista Bayesiano, após aprender a tarefa  $A$ , a informação sobre os parâmetros fica codificada na posterior  $p(\theta | D_A)$ . Como essa posterior exata é intratável, aplica-se uma aproximação de Laplace (MACKAY, 1992): modela-se  $p(\theta | D_A)$  por uma Gaussiana centrada no ótimo  $\theta_A^*$ , cuja *precisão* é dada pela informação de Fisher. Aprender a tarefa  $B$  passa, então, a ser equivalente

a minimizar a perda de  $B$  somada a um termo quadrático que “ancora” os parâmetros em torno de  $\theta_A^*$  de acordo com a equação 2.4.

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_B(\theta) + \frac{\lambda}{2} \sum_i F_i (\theta_i - \theta_{A,i}^*)^2. \quad (2.4)$$

A perda da tarefa corrente de  $B$  é representada por  $\mathcal{L}_B(\theta)$  calculada sobre seus dados. O vetor  $\theta$  reúne todos os pesos e vieses do modelo, enquanto  $\theta_A^*$  representa a solução encontrada após treinar a tarefa anterior  $A$ . O termo quadrático  $\frac{\lambda}{2} \sum_i F_i (\theta_i - \theta_{A,i}^*)^2$  é a penalização “elástica” que ancora cada parâmetro  $i$  próximo do valor aprendido em  $A$ .  $F_i$  é a importância do parâmetro  $i$ , estimada pela diagonal da informação de Fisher em  $A$ , refletindo a sensibilidade local da perda nessa solução.  $\lambda$  é um hiperparâmetro global que regula o compromisso entre estabilidade de preservar  $A$  e a plasticidade de aprender  $B$ . Valores maiores fortalecem a âncora nos parâmetros críticos, enquanto valores menores permitem maiores ajustes para acomodar a nova tarefa.

Em termos práticos, o EWC atua como um *weight decay adaptativo*: parâmetros cruciais para  $A$  (alto  $F_i$ ) são pouco livres para se mover ao aprender  $B$ , enquanto parâmetros pouco relevantes (baixo  $F_i$ ) podem ajustar-se mais. Em sequências com mais de uma tarefa, somam-se termos análogos para cada tarefa passada.

### 2.3.1.3 Estratégias de Isolamento de Parâmetros

Nessa categoria de estratégias, o modelo dedica parâmetros distintos a cada tarefa para evitar esquecimento. Há dois arranjos principais: arquiteturas expansíveis, que “crescem” novos módulos para tarefas futuras enquanto congelam os parâmetros das tarefas antigas e arquiteturas estáticas, nas quais partes fixas da rede são alocadas por tarefa e as regiões alheias são mascaradas durante o treino. Entre os trabalhos importantes encontram-se *PackNet* e o *Hard Attention to the Task* (HAT) como representantes de arquiteturas estáticas (MALLYA; LAZEBNIK, 2018; SERRA et al., 2018) e o *Expert Gate* para as arquiteturas dinâmicas (ALJUNDI; CHAKRAVARTY; TUYTELAARS, 2017).

### 2.3.2 Métricas de Avaliação

Para avaliar o desempenho das estratégias de aprendizado contínuo, adota-se a métrica proposta por Chaudhry et al. (2018), que quantifica o esquecimento como a diferença entre o melhor desempenho já alcançado em uma tarefa ao longo do treinamento e o desempenho atual dessa mesma tarefa após a aprendizagem de novas tarefas, representado pela equação 2.5.

$$f_{k,j} = \max_{l \in \{1, \dots, k-1\}} a_{l,j} - a_{k,j}, \quad \forall j < k \quad (2.5)$$

Nessa equação, para uma tarefa  $j$  previamente aprendida, calcula-se, após o treinamento até a tarefa  $k$ , a diferença entre a melhor precisão já obtida para essa tarefa em qualquer etapa anterior, representada por  $\max a_{l,j}$ , e a precisão atual  $a_{k,j}$  após o aprendizado das novas tarefas.

Um valor maior de  $f_{k,j}$  indica maior perda de desempenho portanto maior esquecimento, enquanto valores próximos de zero sugerem boa retenção do conhecimento previamente adquirido.

## 3 Material e Métodos

Este capítulo tem como objetivo abordar os métodos e materiais empregados neste trabalho, descrevendo a experimentação e avaliação nas tarefas de detecção de objetos e CIL.

### 3.1 Sumário da Metodologia

A metodologia adotada neste trabalho pode ser resumida em quatro etapas principais. Inicialmente, realizou-se a coleta de dados, utilizando conjuntos de imagens disponibilizados pela plataforma Roboflow, já anotados para tarefas de detecção de objetos. Em seguida, procedeu-se ao treinamento do modelo, definindo a arquitetura de detecção Fast R-CNN e os hiperparâmetros utilizados. Na terceira etapa, o modelo foi treinado sob o regime de aprendizado contínuo incremental de classes, por meio da incorporação de estratégias para mitigar o esquecimento catastrófico. Por fim, efetuou-se a avaliação de desempenho, calculando métricas de detecção de objetos e métricas de aprendizado contínuo para comparar o comportamento do modelo ao longo das diferentes experiências.

- Coleta de dados: As imagens utilizadas neste trabalho foram obtidas pelo Roboflow.
- Treinamento de Modelos: O modelo e os hiperparâmetros escolhidos para detecção de objetos.
- Treinamento de Modelos com CIL: As estratégias de aprendizado incremental implementadas no treinamento do modelo.
- Avaliação de Desempenho: Como foram calculadas as métricas de detecção de objetos e aprendizado contínuo.

### 3.2 Base de Dados

Inicialmente, foi utilizado um conjunto de dados referente ao tráfego urbano, assim foi selecionado o *Automotive*, disponibilizado na plataforma *Roboflow* representado na figura 13. O conjunto contém 4743 imagens para treinamento e 303 para validação, todas com anotações de detecção de objetos. Para este trabalho foram consideradas cinco classes: pedestres, carros, motos, bicicletas e ônibus. As imagens apresentam ampla variabilidade de iluminação (diferentes horários do dia) e de cenário (ruas e rodovias), além de mudanças de ângulo, escala e oclusões. Essa heterogeneidade aliada ao pequeno número de amostras por classes torna o problema desafiador.

Posteriormente, utilizou-se um conjunto com menos imagens e classes, o *Rock, Paper, Scissors* (ROBOFLOW, 2025) representado na figura 14, também disponibilizado na plataforma

Figura 13 – Exemplo do conjunto *Automotive*



Fonte: Autoria Própria

*Roboflow*. Apesar de ser menor com apenas 2196 imagens para treinamento e 604 para validação, a quantidade de classes é menor, o que facilita o aprendizado e tempo de treinamento. Na prática, esse cenário resultou em métricas superiores nos experimentos deste trabalho.

Figura 14 – Exemplo do conjunto *Rock, Papers, Scissors*



Fonte: Autoria Própria

Todos os conjuntos de dados foram utilizados com o formato COCO (LIN et al., 2014), isso permite que seja utilizado as ferramentas do *pycocotools*, interface de programação de aplicações (API) para facilitar o uso desses *datasets*, que serão utilizados pelo Avalanche para o cálculo das métricas avaliativas.

### 3.3 Modelo e Hiperparâmetros

Nos experimentos deste trabalho empregou-se o *Fast RCNN* como algoritmo de detecção de objetos, utilizando uma *ResNet-101* como *backbone* para extração de características. Além disso, o *backbone* foi pré-treinado no dataset COCO (LIN et al., 2014), opção disponível

no *pytorch* para melhorar sua eficiência, diminuindo os efeitos de um conjunto com número limitado de imagens.

Na tarefa de detecção, *abackbone* é responsável por extrair mapas de características da imagem, convertendo pixels em representações úteis para as etapas seguintes. O *anchor generator* define uma malha de caixas âncora em múltiplas escalas que servem como pontos de partida para prever as *bounding boxes*. Já o *Rol pooler*, recorta e reamostra cada região de interesse sobre os mapas da *backbone* para um tamanho fixo (como  $7 \times 7$ ), padronizando a entrada da cabeça de detecção.

Tabela 1 – Valores dos Hiperparâmetros do FastRCNN

Hiperparâmetros	Variáveis	Valores
Backbone	Out channels	2048
Anchor Generator	Sizes	32, 64, 128, 256, 512
Anchor Generator	Aspect Ratios	0,5, 1,0, 2,0
Rol Pooler	Featmap names	0, 1, 2, 3
Rol Pooler	Output size	7
Rol Pooler	Sampling ratio	2

Fonte: Elaborado pelo autor.

Como otimizador, foi adotado o *Stochastic Gradient Descent* (SGD), amplamente usada na literatura relacionada a aprendizado profundo (SUTSKEVER et al., 2013), com *momentum* e *weight decay*. Aliado a isso, foi utilizado o *plugin* do Avalanche, uma biblioteca de aprendizado contínuo que será desenvolvido posteriormente, com *learning rate scheduler* responsável por fazer com que a taxa de aprendizado seja dinâmica, diminuindo o valor graditivamente com cada iteração. Esse ajuste foi feito para melhorar o desempenho do FastRCNN com a quantidade de imagens presente, representado na tabela 2.

Tabela 2 – Valores do Otimizador

Hiperparâmetros	Variáveis	Valores
SDG	<i>Weight decay</i>	$5 \times 10^{-4}$
SDG	<i>Momentum</i>	0,9
SDG	<i>Learning Rate</i>	$1 \times 10^{-3}$
Learning rate scheduler	<i>Start factor</i>	$5 \times 10^{-3}$
Learning rate scheduler	<i>Step granularity</i>	Iteration

Fonte: Elaborado pelo autor.

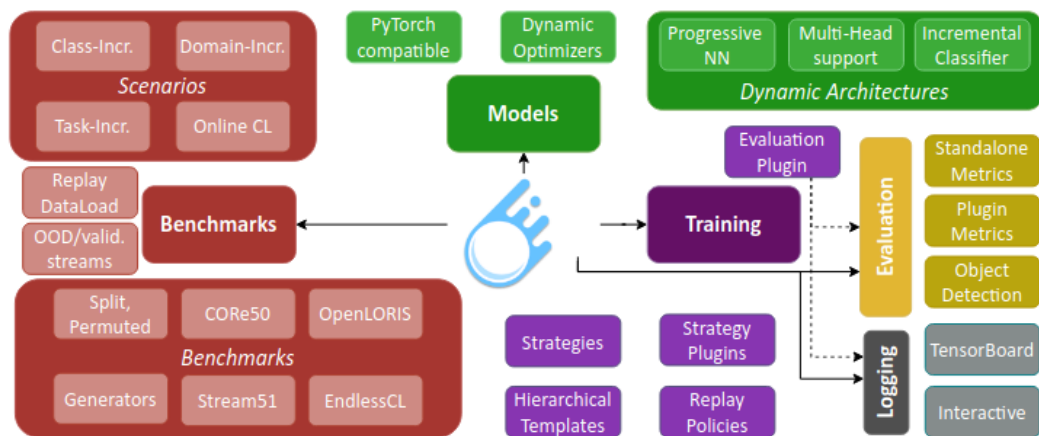
### 3.4 Aprendizado Contínuo

Neste trabalho, para o treinamento do modelo, foi determinado o Avalanche como um orquestrador e uso de dois métodos para definir a estratégia CIL: *Replay* e *EWC*

### 3.4.1 Avalanche

O Avalanche é um *framework* que foi feito voltado para problemas de classificação (LOMONACO et al., 2021). Entretanto, mesmo com menos suporte, ele oferece abstrações que viabilizam a extensão para *detecção de objetos*. Portanto, foram utilizados 3 abstrações importantes para a realização deste experimento. Os *benchmarks* organizam o CIL em fluxos, experiências e permitem definir como os dados são particionados e apresentados ao modelo ao longo do tempo. Os *plugins*, que injetam métodos durante o ciclo de treino, métodos esses que podem ser tanto na parte de otimização do modelo quanto as próprias técnicas de CIL, sem acoplar a lógica ao modelo. Por fim, as estratégias que encapsulam o *loop* de treinamento/validação em aprendizado contínuo, coordenando modelo, perdas, métricas e plugins. A figura 15 é uma ilustração das funcionalidades da biblioteca.

Figura 15 – Ilustração da biblioteca Avalanche



Fonte: Adaptado de Lomonaco et al. (2021)

### 3.4.2 Replay

O *Replay* é um *plugin* oferecido pelo Avalanche que busca mitigar a perda de desempenho ao longo do tempo por meio de uma memória externa de experiências anteriores. A ideia central é que durante o treinamento em um novo lote de dados o modelo não vê apenas exemplos recentes, mas também uma amostra de exemplos antigos guardados em um *buffer* de *replay*. Ao reintroduzir periodicamente esses exemplos passados no gradiente, o treinamento preserva conhecimento já adquirido enquanto incorpora informações novas.

O *Replay* do Avalanche, mantém uma *memória externa* com imagens e anotações selecionados aleatoriamente ao final de cada experiência. Antes de treinar uma experiência, é feito uma chamada para substituir o *dataloader* da experiência atual com um mistura do que está na memória externa com as imagens da experiência atual, buscando um balanceamento por experiência. Após encerrar essa experiência, novos padrões da experiência recém-treinada

são adicionados ao *buffer* e caso a capacidade seja excedida, há remoção do excedente, sempre tentando seguir a política padrão do *plugin* de amostragem aleatória e balanceada entre experiências.

Na estratégia implementada, foi definido um total de 200 imagens limite para serem usadas no *Replay*.

### 3.4.3 Elastic Weight Consolidation

O EWC é uma estratégia de aprendizado contínuo que busca preservar conhecimentos prévios sem impedir a aprendizagem de informações novas. A ideia central é identificar quais parâmetros do modelo foram mais “importantes” para tarefas já aprendidas e, durante o treinamento em novas experiências, penalizar alterações bruscas nesses parâmetros. Assim, o modelo tende a manter desempenho no que já sabe enquanto se adapta ao que surge.

O *plugin* de EWC nativo do Avalanche não funciona para detecção de objetos, como o Avalanche é um *framework* voltado a classificação e não a detecção de objetos, é esperado que haja incompatibilidade em certos métodos. Neste caso, uma tentativa de criar o EWC foi feito utilizando-se das abstrações presentes no *framework* para criação de *plugins* e a codificação com base no artigo (AICH, 2021) e sua implementação seguiria a pipeline utilizada na competição *CLVision2022* (PELLEGRINI et al., 2022), demonstrado na figura 16.

Ao integrar o *plugin* no ciclo de treino seria calculado a importância de cada parâmetro com base em métricas derivadas do gradiente, como a informação de Fisher em sua forma diagonal ao final de cada experiência. Em seguida, durante o treinamento de uma época, adiciona-se um termo de regularização à função de perda que “puxa” os pesos atuais na direção dos valores anteriores considerados críticos. Em termos práticos, isso resulta em um compromisso controlado entre estabilidade e plasticidade.

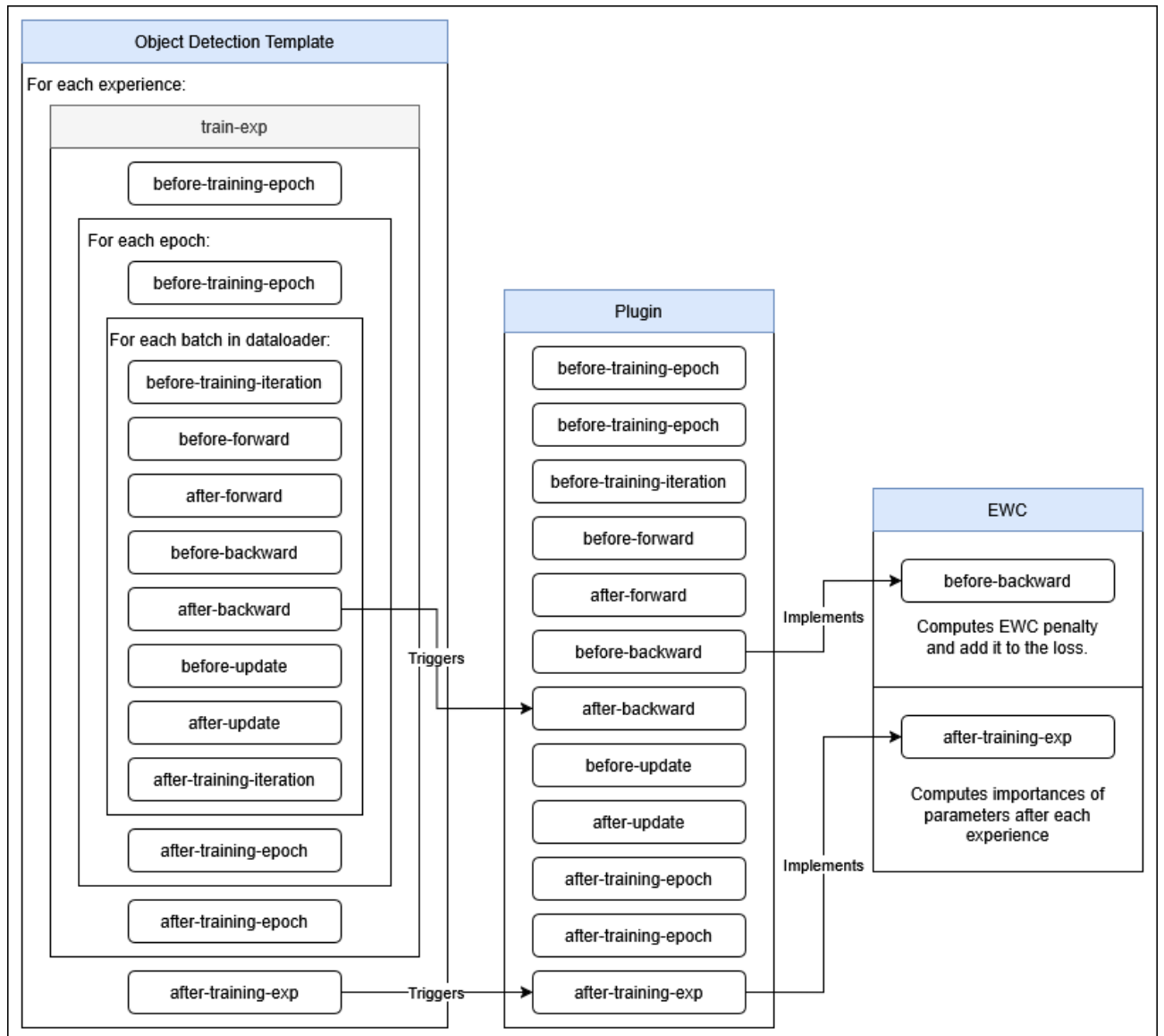
## 3.5 Métricas avaliativas

### 3.5.1 Detecção de Objetos

A avaliação dos modelos será realizada com o auxílio do *pycocotools*, que implementa as métricas no estilo COCO. Foi adotado a interseção sobre união (IOU) como critério de sobreposição entre a caixa predita e a caixa de referência. Para duas caixas  $A$  (predição) e  $B$  (real), a IOU é dada por:

$$\text{IOU} = \frac{|A \cap B|}{|A \cup B|}. \quad (3.1)$$

No processo de avaliação, uma detecção é considerada verdadeira quando a classe está correta e a sobreposição atende a um limiar mínimo de IOU; caso contrário, conta como *falso positivo*. Anotações não pareadas são contabilizadas como *falsos negativos*.

Figura 16 – Representação da *pipeline* do EWC durante o treinamento

Fonte: Adaptado de <<https://github.com/cl-challenge/clvision-challenge-2022/>>. Acesso em: 27 out de 2025.

Nesse projeto, os resultados seguem as três configurações usuais do COCO:  $AP@0,50$ ,  $AP@0,75$  e  $AP@[0,50:0,95]$ , que corresponde à média da AP em dez limiares igualmente espaçados de 0,50 a 0,95 (passo 0,05). Nesse trabalho, será adotado o *Medium Average Precision* (mAP) que é a média calculado com a média de todas as detecções AP feitas na validação.

Além dessa métrica será usado o AR foi utilizado como ilustrado na equação 2.3, mas para o escopo desse trabalho, *Medium Average Recall* (mAR) foi utilizando, que é a média entre as detecção AR feitas na validação.

### 3.5.2 Aprendizado Contínuo

A avaliação dos métodos de CIL foi conduzida com as implementações do Avalanche. Para quantificar o esquecimento, adotou-se a equação 2.5 em que a cada nova classe incorporada, calcula-se, para cada classe já vista, a diferença entre a sua melhor precisão observada até então e a precisão obtida após o treinamento da experiência corrente. O índice reportado corresponde à média dessas diferenças sobre todas as classes anteriores. Portanto, quanto maior o valor, maior o esquecimento.

## 3.6 Ferramentas e Tecnologias

Neste trabalho, adotou-se a linguagem *Python* pela ampla disponibilidade de bibliotecas e *frameworks* voltados a visão computacional e aprendizado profundo, principalmente com o *PyTorch* e o *Avalanche* para este projeto. O *PyTorch* foi empregado na definição das arquiteturas, na modelagem, no treinamento e avaliação dos modelos, já o *Avalanche* foi utilizado para orquestrar o *pipeline* de aprendizado contínuo incremental (CIL), oferecendo abstrações de cenários, estratégias e *plugins*, o que facilitou a implementação e a comparação das abordagens.

O ambiente de desenvolvimento escolhido foi o *Visual Studio*, utilizado pela facilidade de organização do projeto e documentação. Por fim, a visualização dos resultados e a geração de figuras foram realizadas com a biblioteca *matplotlib*.

Tabela 3 – Ferramentas e Tecnologias utilizada

<b>Categoria</b>	<b>Ferramenta</b>	<b>Objetivo</b>
Ambiente	<i>Visual Studio Code</i>	Editor e ambiente de desenvolvimento.
Linguagem	<i>Python</i>	Linguagem para codificação do projeto.
Visão Computacional	<i>PyTorch</i>	<i>Framework</i> de aprendizado profundo utilizada para modelagem e treinamento.
API	<i>Pycocotools</i>	API para interação com conjuntos de dados no formato COCO.
Orquestrador	<i>Avalanche</i>	<i>Framework</i> para orquestrar o <i>pipeline</i> de CIL (estratégias, <i>plugins</i> , métricas).
Visualização	<i>matplotlib</i>	Biblioteca para geração de gráficos e visualizações dos resultados.

Fonte: Elaborado pelo autor.

## 4 Resultados

Neste capítulo, são apresentados os resultados obtidos com os conjuntos de dados *Automotive* e *Rock, Paper and Scissors*. Inicialmente, foi analisado o desempenho de detecção de objetos sem técnicas adicionais. Em seguida, é reportado os resultados do treinamento com aprendizado contínuo no regime incremental de classes, no qual as classes são introduzidas sequencialmente e treinadas isoladamente, para cada nova classe, executa-se uma experiência. Esse arranjo permite comparar a linha de base com a evolução do modelo à medida que incorpora novas classes, mantendo o mesmo protocolo de avaliação.

### 4.1 Desempenho da Detecção de Objetos

Os resultados com o Fast RCNN no conjunto *Automotive* são apresentados na tabela 4. Observa-se desempenho baixo: o AP@0,50 médio entre classes alcança apenas 0,234 e o mAP@0,50:0,95 fica em 0,116. Nessa condição, mesmo a aplicação de técnicas de aprendizado contínuo o seu impacto seria difícil de ser analisado, pois a linha de base do detector já é limitada dado as múltiplas classes, variação de iluminação, cenário e maior complexidade do domínio.

Tabela 4 – Avaliação no conjunto *Automotive*.

Métrica	IOU	Resultado
<i>Average Precision</i> (AP)	0.50	0.234
<i>Average Precision</i> (AP)	0.75	0.107
<i>mean Average Precision</i> (mAP)	0.50:0.95	0.116
<i>mean Average Recall</i> (mAR)	0.50:0.95	0.151

Fonte: Autoria própria.

Entretanto, o conjunto *Rock, Paper and Scissor* representado na tabela 5, o RCNN atinge um AP@0,50 de 0,665 e mAP@0,50:0,95 de 0,315, valores compatíveis com um cenário mais controlado por causa de uma menor variação de fundo e iluminação, causando imagens mais homogêneas e com menos classes, apresentando três classes, diferente das cinco do *Automotive*. Essa diferença de domínio explica, em parte, a melhora na precisão e na cobertura.

Dessa forma, os resultados reforçam a sensibilidade do desempenho do detector à complexidade do domínio da tarefa, principalmente aplicado a tráfego urbano quando não se tem muitas imagens, fator crucial ao interpretar ganhos eventuais com técnicas de aprendizado contínuo.

Tabela 5 – Avaliação no conjunto *Rock, Paper and Scissor*.

<b>Métrica</b>	<b>IOU</b>	<b>Resultado</b>
<i>Average Precision</i> (AP)	0.50	0.665
<i>Average Precision</i> (AP)	0.75	0.344
<i>mean Average Precision</i> (mAP)	0.50:0.95	0.315
<i>mean Average Recall</i> (mAR)	0.50:0.95	0.437

Fonte: Autoria própria.

#### 4.1.1 Desempenho das técnicas de Aprendizado Contínuo

Após definir uma comparação base, foi aplicado os métodos de *Replay* e EWC ao treinamento do FastRCNN no conjunto de dados *Rock, Papers, Scissors*. Inicialmente, já é esperado uma queda no desempenho base do algoritmo pois além do treinamento padrão do modelo, a cada experiência ele precisará mitigar o esquecimento catastrófico.

Entretanto, neste projeto não houve resultados satisfatórios quando aplicados as técnicas. Durante o treinamento, a partir da segunda experiência de todos os testes feitos, houve esquecimento catastrófico nas classes já aprendidas e prejuízo nos treinamentos posteriores.

Uma das hipóteses desse desempenho pode ser explicado pois, ao treinar a primeira classe, o modelo está aprendendo a classificar as demais classes como *background* (ou uma área que não precisa ser detectada). Isso pesa no aprendizado dos demais parâmetros. Além disso, o fato das classes iniciais sofrerem esquecimento catastrófico, chegando a um valor de zero em vários casos, indica que há um erro na implementação desses métodos.

Embora na literatura há ocorrências de perda de conhecimento significativas no uso de EWC (VEN; TUYTELAARS; TOLIAS, 2022b), não é uma justificava para o resultado, em sua maioria, chegar em zero ou um valor próximo.

## 5 Conclusão

Neste trabalho investigou-se o uso de técnicas de *aprendizado contínuo* em algoritmos de detecção de objetos, com aplicação no contexto de tráfego urbano. Foi utilizado do Fast RCNN em dois conjuntos de dados: *Automotive*, voltado ao cenário urbano, mais dinâmico e com maior variabilidade visual e *Rock, Paper and Scissors*, composto por imagens mais homogêneas, portanto mais favorável ao treinamento. Em seguida, os modelos foram treinados no cenário incremental de classes, no qual, em vez de apresentar todas as classes e imagens de uma só vez, as classes são introduzidas e aprendidas sequencialmente e isoladamente.

Os resultados experimentais não atingiram os objetivos propostos. A hipótese para possíveis motivos são a implementação, uma vez que o esquecimento catastrófico persistiu mesmo com a aplicação dos métodos *Replay* e EWC.

A aplicação de aprendizado contínuo à detecção de objetos ainda é recente e gera problemas desafiadores. Por exemplo, ainda não foi definido um estado da arte, a escassez de bibliotecas dedicadas ao aprendizado contínuo de detecção de objetos e literatura menos consolidada do que problemas mais consolidados de aprendizado contínuo, como a classificação. Além disso, particularidades da detecção dificultam a transposição direta de estratégias consagradas em classificação.

Portanto, dado este cenário, para trabalhos futuros pode se criar conjuntos de dados dedicados a aprendizado contínuo, análogos ao [Lomonaco e Maltoni \(2017\)](#), porém voltado a situações reais, como na área da saúde, tráfego urbano ou agricultura. Isso pode ser uma das causas de existirem poucos artigos relacionados a aplicações reais, sendo a maioria dos experimentos voltados aos conjuntos de dados COCO e VOC que possuem um foco mais generalista de treino ([LIN et al., 2014](#); [EVERINGHAM et al., 2010](#)).

## Referências

- AGGARWAL, C. *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, 2018. (Computer Science). ISBN 9783319944630. Disponível em: <<https://books.google.com.br/books?id=achqDwAAQBAJ>>.
- AICH, A. Elastic weight consolidation (ewc): Nuts and bolts. *ArXiv*, abs/2105.04093, 2021. Disponível em: <<https://api.semanticscholar.org/CorpusID:234336029>>.
- ALJUNDI, R.; CHAKRAVARTY, P.; TUYTELAARS, T. Expert gate: Lifelong learning with a network of experts. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 7120–7129.
- BOUKTIF, S.; CHENIKI, A.; OUNI, A.; EL-SAYED, H. Deep reinforcement learning for traffic signal control with consistent state and reward design approach. *Knowledge-Based Systems*, v. 267, p. 110440, 2023. ISSN 0950-7051. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950705123001909>>.
- BURKOV, A. *Machine Learning Engineering*. True Positive Incorporated, 2020. ISBN 9781999579579. Disponível em: <<https://books.google.com.br/books?id=HeXizQEACAAJ>>.
- CHAUDHRY, A.; DOKANIA, P. K.; AJANTHAN, T.; TORR, P. H. S. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: FERRARI, V.; HEBERT, M.; SMINCHISESCU, C.; WEISS, Y. (Ed.). *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018. p. 556–572. ISBN 978-3-030-01252-6.
- CHEN, Z.; LIU, B. Continual learning and catastrophic forgetting. In: *Lifelong Machine Learning*. [S.l.]: Springer, 2018. p. 55–75.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.l.: s.n.], 2005. v. 1, p. 886–893 vol. 1.
- 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.
- EVERINGHAM, M.; GOOL, L. V.; WILLIAMS, C. K. I.; WINN, J.; ZISSERMAN, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, v. 88, n. 2, p. 303–338, 2010. Disponível em: <<https://doi.org/10.1007/s11263-009-0275-4>>.
- GIRSHICK, R. Fast r-cnn. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1440–1448.
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 580–587.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial networks. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 63, n. 11, p. 139–144, out. 2020. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/3422622>>.

HAMET, P.; TREMBLAY, J. Artificial intelligence in medicine. *Metabolism*, v. 69, p. S36–S40, 2017. ISSN 0026-0495. Insights Into the Future of Medicine: Technologies, Concepts, and Integration. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S002604951730015X>>.

HAYKIN, S. *Neural Networks and Learning Machines*. Prentice Hall, 2009. (Neural networks and learning machines, v. 10). ISBN 9780131471399. Disponível em: <[https://books.google.com.br/books?id=K7P36IKzl\\_QC](https://books.google.com.br/books?id=K7P36IKzl_QC)>.

HECHT-NIELSEN. Theory of the backpropagation neural network. In: *International 1989 Joint Conference on Neural Networks*. [S.l.: s.n.], 1989. p. 593–605 vol.1.

HUANG, J.; KOROTEEV, D. D. Artificial intelligence for planning of energy and waste management. *Sustainable Energy Technologies and Assessments*, v. 47, p. 101426, 2021. ISSN 2213-1388. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2213138821004367>>.

LANGE, M. D.; ALJUNDI, R.; MASANA, M.; PARISOT, S.; JIA, X.; LEONARDIS, A.; SLABAUGH, G.; TUYTELAARS, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 44, n. 7, p. 3366–3385, 2021.

LANGE, M. D.; TUYTELAARS, T. Continual prototype evolution: Learning online from non-stationary data streams. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2021. p. 8230–8239.

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.

LEE, S.-W.; KIM, J.-H.; JUN, J.; HA, J.-W.; ZHANG, B.-T. Overcoming catastrophic forgetting by incremental moment matching. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 4655–4665. ISBN 9781510860964.

LI, Z.; HOIEM, D. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 40, n. 12, p. 2935–2947, 2018.

LI, Z.; LIU, F.; YANG, W.; PENG, S.; ZHOU, J. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, IEEE, v. 33, n. 12, p. 6999–7019, 2021.

LIAKOS, K. G.; BUSATO, P.; MOSHOV, D.; PEARSON, S.; BOCHTIS, D. Machine learning in agriculture: A review. *Sensors*, Mdpi, v. 18, n. 8, p. 2674, 2018.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014. p. 740–755. ISBN 978-3-319-10602-1.

LOMONACO, V.; MALTONI, D. Core50: a new dataset and benchmark for continuous object recognition. In: LEVINE, S.; VANHOUCHE, V.; GOLDBERG, K. (Ed.). *Proceedings of the 1st Annual Conference on Robot Learning*. PMLR, 2017. (Proceedings of Machine Learning Research, v. 78), p. 17–26. Disponível em: <<https://proceedings.mlr.press/v78/lomonaco17a.html>>.

LOMONACO, V.; PELLEGRINI, L.; COSSU, A.; CARTA, A.; GRAFFIETI, G.; HAYES, T. L.; LANGE, M. D.; MASANA, M.; POMPONI, J.; VEN, G. M. van de; MUNDT, M.; SHE, Q.; COOPER, K.; FOREST, J.; BELOUADAH, E.; CALDERARA, S.; PARISI, G. I.; CUZZOLIN, F.; TOLIAS, A. S.; SCARDAPANE, S.; ANTIGA, L.; AHMAD, S.; POPESCU, A.; KANAN, C.; WEIJER, J. van de; TUYTELAARS, T.; BACCIU, D.; MALTONI, D. Avalanche: An end-to-end library for continual learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2021. p. 3600–3610.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, v. 60, n. 2, p. 91–110, 2004. Disponível em: <<https://doi.org/10.1023/B:VISI.0000029664.99615.94>>.

MACKAY, D. J. C. A practical bayesian framework for backpropagation networks. *Neural Computation*, v. 4, n. 3, p. 448–472, 1992.

MALLYA, A.; LAZEBNIK, S. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning . In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2018. p. 7765–7773. Disponível em: <<https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00810>>.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, n. 2, p. 115–133, 1943.

MCRAE, K. Catastrophic interference is eliminated in pretrained networks. 03 1993.

NAQA, I. E.; MURPHY, M. J. What is machine learning? In: \_\_\_\_\_. *Machine Learning in Radiation Oncology: Theory and Applications*. Cham: Springer International Publishing, 2015. p. 3–11. ISBN 978-3-319-18305-3. Disponível em: <[https://doi.org/10.1007/978-3-319-18305-3\\_1](https://doi.org/10.1007/978-3-319-18305-3_1)>.

PADILLA, R.; NETTO, S. L.; SILVA, E. A. D. A survey on performance metrics for object-detection algorithms. In: IEEE. *2020 international conference on systems, signals and image processing (IWSSIP)*. [S.l.], 2020. p. 237–242.

PELLEGRINI, L.; ZHU, C.; XIAO, F.; YAN, Z.; CARTA, A.; LANGE, M. D.; LOMONACO, V.; SUMBALY, R.; RODRIGUEZ, P.; VAZQUEZ, D. *3rd Continual Learning Workshop Challenge on Egocentric Category and Instance Level Object Understanding*. 2022. Disponível em: <<https://arxiv.org/abs/2212.06833>>.

PRINCE, S. J. D. *Computer Vision: Models, Learning, and Inference*. [S.l.]: Cambridge University Press, 2012.

RANNEN, A.; ALJUNDI, R.; BLASCHKO, M. B.; TUYTELAARS, T. Encoder Based Lifelong Learning . In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2017. p. 1329–1337. ISSN 2380-7504. Disponível em: <<https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.148>>.

REBUFFI, S.-A.; KOLESNIKOV, A.; SPERL, G.; LAMPERT, C. H. icarl: Incremental classifier and representation learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017.

ROBINS, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, Taylor & Francis, v. 7, n. 2, p. 123–146, 1995.

ROBOFLOW. Open Source Dataset, *rock-paper-scissors Dataset*. Roboflow, 2025. <<https://universe.roboflow.com/roboflow-58fyf/rock-paper-scissors-sxsw>>. Disponível em: <<https://universe.roboflow.com/roboflow-58fyf/rock-paper-scissors-sxsw>>.

ROLNICK, D.; AHUJA, A.; SCHWARZ, J.; LILICRAP, T. P.; WAYNE, G. Experience replay for continual learning. In: \_\_\_\_\_. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

SALTON, G.; MCGILL, M. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983. (International student edition). ISBN 9780070544840. Disponível em: <<https://books.google.com.br/books?id=7f5TAAAAMAAJ>>.

SERRA, J.; SURIS, D.; MIRON, M.; KARATZOGLOU, A. Overcoming catastrophic forgetting with hard attention to the task. In: DY, J.; KRAUSE, A. (Ed.). *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018. (Proceedings of Machine Learning Research, v. 80), p. 4548–4557. Disponível em: <<https://proceedings.mlr.press/v80/serra18a.html>>.

SHIN, H.; LEE, J. K.; KIM, J.; KIM, J. Continual learning with deep generative replay. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 2994–3003. ISBN 9781510860964.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. In: *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. [S.l.: s.n.], 2018. p. 1–6.

SHOTTON, J.; WINN, J.; ROTHER, C.; CRIMINISI, A. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vision*, Kluwer Academic Publishers, USA, v. 81, n. 1, p. 2–23, jan. 2009. ISSN 0920-5691. Disponível em: <<https://doi.org/10.1007/s11263-007-0109-1>>.

SUTSKEVER, I.; MARTENS, J.; DAHL, G.; HINTON, G. On the importance of initialization and momentum in deep learning. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. [S.l.]: JMLR.org, 2013. (ICML'13), p. III–1139–III–1147.

THRUN, S. *Lifelong Learning: A Case Study*. Pittsburgh, PA, 1995.

UIJLINGS, J. R. R.; SANDE, K. E. A. van de; GEVERS, T.; SMEULDERS, A. W. M. Selective search for object recognition. *International Journal of Computer Vision*, v. 104, n. 2, p. 154–171, 2013. Disponível em: <<https://doi.org/10.1007/s11263-013-0620-5>>.

VEN, G. M. Van de; TUYTELAARS, T.; TOLIAS, A. S. Three types of incremental learning. *Nature Machine Intelligence*, Nature Publishing Group UK London, v. 4, n. 12, p. 1185–1197, 2022.

- VEN, G. M. van de; TUYTELAARS, T.; TOLIAS, A. S. Three types of incremental learning. *Nature Machine Intelligence*, v. 4, n. 12, p. 1185–1197, 2022. Disponível em: <<https://doi.org/10.1038/s42256-022-00568-3>>.
- VERWIMP, E.; YANG, K.; PARISOT, S.; HONG, L.; MCDONAGH, S.; PÉREZ-PELLITERO, E.; LANGE, M. D.; TUYTELAARS, T. Clad: A realistic continual learning benchmark for autonomous driving. *Neural Networks*, Elsevier, v. 161, p. 659–669, 2023.
- YANG, L.; LUO, Z.; ZHANG, S.; TENG, F.; LI, T. Continual learning for smart city: A survey. *arXiv preprint arXiv:2404.00983*, 2024.
- YAO, H.; WU, F.; KE, J.; TANG, X.; JIA, Y.; LU, S.; GONG, P.; YE, J.; CHUXING, D.; LI, Z. Deep multi-view spatial-temporal network for taxi demand prediction. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. [S.l.]: AAAI Press, 2018. (AAAI'18/IAAI'18/EAAI'18). ISBN 978-1-57735-800-8.
- ZENKE, F.; POOLE, B.; GANGULI, S. Continual learning through synaptic intelligence. In: PMLR. *International conference on machine learning*. [S.l.], 2017. p. 3987–3995.
- ZENKE, F.; POOLE, B.; GANGULI, S. Continual learning through synaptic intelligence. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. [S.l.]: JMLR.org, 2017. (ICML'17), p. 3987–3995.
- ZHANG, J.; ZHENG, Y.; QI, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2017. (AAAI'17), p. 1655–1661.
- ZHAO, X.; WANG, L.; ZHANG, Y.; HAN, X.; DEVECI, M.; PARMAR, M. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, Springer, v. 57, n. 4, p. 99, 2024.
- ZHAO, Z.-Q.; ZHENG, P.; XU, S.-T.; WU, X. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, v. 30, n. 11, p. 3212–3232, 2019.