

ORLANDO DA SILVA JUNIOR

**CLASSIFICAÇÃO DE IMAGENS EM FLUXOS DE
DADOS UTILIZANDO APRENDIZADO PROFUNDO**

São Paulo
2023

ORLANDO DA SILVA JUNIOR

**CLASSIFICAÇÃO DE IMAGENS EM FLUXOS DE
DADOS UTILIZANDO APRENDIZADO PROFUNDO**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do Título de Doutor em Ciências.

São Paulo
2023

ORLANDO DA SILVA JUNIOR

**CLASSIFICAÇÃO DE IMAGENS EM FLUXOS DE
DADOS UTILIZANDO APRENDIZADO PROFUNDO**

Versão Corrigida

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do Título de Doutor em Ciências.

Área de Concentração:

Engenharia de Computação

Orientador:

Prof. Dr. Jorge Rady de Almeida Junior

São Paulo
2023

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

da Silva Junior, Orlando

Classificação de Imagens em Fluxos de Dados Utilizando Aprendizado Profundo / O. da Silva Junior -- versão corr. -- São Paulo, 2023.

113 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Aprendizado de Máquina 2.Redes Neurais 3.Aprendizado Profundo 4.Processamento de Imagem 5.Fluxos de Dados I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

A meus pais, Orlando e Silvia.

AGRADECIMENTOS

Aos meus pais, Orlando e Silvia, e aos meus irmãos, Jacquelyne e Rodrigo.

Ao meu orientador, Prof. Dr. Jorge Rady de Almeida Junior, pelo acolhimento antes da minha chegada à Universidade de São Paulo (USP), pelo aceite em trabalhar comigo neste projeto de doutorado e por não desistir de mim durante todos esses anos.

Ao Prof. Dr. Pedro Luiz Pizzigatti Corrêa e à Prof.^a Dr.^a Anna Helena Reali Costa, pelas sugestões e críticas ao meu trabalho na qualificação. Ao Prof. Geraldo Francisco Donegá Zafalon, ao Prof. Dr. Ronaldo Cristiano Prati, ao Prof. Dr. Rogério de Oliveira e à Prof.^a Dr.^a Liria Matsumoto Sato, pelas sugestões e críticas ao meu trabalho final.

Aos professores da Escola Politécnica (EP) que em algum momento durante esta jornada tiveram contato comigo. Aos colegas estudantes que me acompanharam na caminhada e se permitiram ouvir o meu trabalho.

Aos amigos, familiares e colegas de profissão que colaboraram de alguma forma para a realização deste trabalho.

Aos gestores diretos e indiretos dos últimos anos que colaboraram para que eu pudesse continuar os estudos e permanecesse na academia após o mestrado. Aos colegas professores que me ensinaram novas formas de aprender e ensinar. Aos colegas pesquisadores, que me inspiram a continuar na academia.

Aos meus eternos alunos de graduação, iniciação científica e pós-graduação, que me mostram todos os dias o valor do ser docente.

Nós precisamos aprender não apenas por erudição, mas também para melhorar nossa habilidade de gerar comportamento efetivo.

Traduzido e adaptado de uma das edições antigas de
Artificial Intelligence: A Modern Approach,
Stuart J. Russell e Peter Norvig.

RESUMO

SILVA JUNIOR, ORLANDO DA. **Classificação de imagens em fluxos de dados utilizando aprendizado profundo**. 2023. 111 f. Tese (Doutorado em Engenharia Elétrica) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2023.

As aplicações modernas têm transformado a vida em sociedade e trazido novos desafios computacionais. Um desses desafios é a produção massiva e não estacionária de dados de imagem, que requerem o desenvolvimento de algoritmos eficientes para processamento em tempo real e técnicas de análise para lidar com a natureza dinâmica desses dados. Nesta tese, a tarefa de classificação de imagens é explorada por meio de técnicas de aprendizagem em fluxos de dados. O objetivo é implementar algoritmos para a classificação de imagens utilizando aprendizado profundo, concentrando-se na investigação de modelos e algoritmos para o processamento e a aprendizagem de imagens em fluxos de dados. Um *framework* completo é proposto para a construção e validação de modelos das redes neurais profundas. Diferentes experimentos são conduzidos para avaliar a capacidade preditiva do *framework* em classificar novas imagens e os resultados são comparados com outros métodos do estado-da-arte. Os resultados mostram que o *framework* proposto supera os métodos comparados na maior parte dos cenários avaliados.

Palavras-Chave – Aprendizado de Máquina, Redes Neurais, Aprendizado Profundo, Processamento de Imagem, Fluxos de Dados.

ABSTRACT

Modern applications have transformed life in society and brought new computational challenges. One of these challenges is the massive and non-stationary production of image data, which requires the development of efficient algorithms for real-time processing and analysis techniques to deal with the dynamic nature of these data. In this thesis, the task of image classification is explored through data stream learning techniques. The aim is to implement algorithms for image classification using deep learning, focusing on investigating models and algorithms for processing and learning images in data streams. A complete framework is proposed for the construction and validation of deep neural network models. Different experiments are conducted to evaluate the predictive capacity of the framework in classifying new images and the results are compared with other state-of-the-art methods. The results show that the proposed framework outperforms the compared methods in most of the evaluated scenarios.

Keywords – Machine Learning, Neural Networks, Deep Learning, Image Processing, Data Streams.

LISTA DE FIGURAS

1	Processo de classificação de uma imagem.	25
2	Processo de aprendizagem em duas fases: treinamento e teste.	27
3	Simulação de fluxos de dados seguindo a abordagem incremental.	32
4	Simulação de fluxos de dados seguindo a abordagem <i>online-offline</i>	32
5	<i>Framework</i> de aprendizagem incremental para cenários <i>online</i>	40
6	Arquitetura do Sistema IMS.	41
7	Pipeline para classificação automática de imagens.	42
8	Criação de <i>buffers</i> no Algoritmo ExStream.	46
9	<i>Framework</i> proposto – arquitetura e fluxo de funcionamento.	51
10	<i>Framework</i> proposto – detalhamento da fase de extração de atributos.	53
11	<i>Framework</i> proposto – detalhamento da fase de aprendizagem.	54
12	Abordagem experimental adotada para a validação do <i>framework</i> proposto.	56
13	Amostra de 25 imagens de alimentos e 25 imagens de não alimentos aleatoriamente selecionadas da base de dados Food-5k.	58
14	Amostra das 50 primeiras imagens contidas na base de dados CIFAR-10.	59
15	Amostra de imagens presentes na base de dados MNIST.	60
16	Desempenho preditivo do <i>framework</i> usando modelos de seleção de atributos sobre a base de dados Food-5k para a tarefa de classificação binária: (a) AM^B e (b) AG^B , conforme o número de atributos selecionados.	71
17	Desempenho preditivo do <i>framework</i> usando modelos de seleção de atributos sobre a base de dados CIFAR-10 para a tarefa de classificação binária: (a) AM^B e (b) AG^B , conforme o número de atributos selecionados.	73
18	Desempenho preditivo do <i>framework</i> usando modelos de seleção de atributos sobre a base de dados MNIST para a tarefa de classificação binária: (a) AM^B e (b) AG^B , conforme o número de atributos selecionados.	74

19	Desempenho preditivo do <i>framework</i> usando modelos de seleção de atributos sobre a base de dados CIFAR-10 para a tarefa de classificação multiclases: (a) AM^B e (b) AG^B , conforme o número de atributos selecionados.	76
20	Desempenho preditivo do <i>framework</i> usando modelos de seleção de atributos sobre a base de dados MNIST para a tarefa de classificação multiclases: (a) AM^M e (b) AG^M , conforme o número de atributos selecionados.	77
21	Desempenho preditivo dos métodos avaliados sobre a base de dados Food-5k: (a) AM^B e (b) AG^B , conforme o tamanho do lote.	80
22	Desempenho preditivo do <i>framework</i> sobre a base de dados CIFAR-10: (a) AM^B e (b) AG^B , conforme o tamanho do lote.	82
23	Desempenho preditivo do <i>framework</i> sobre a base de dados MNIST: (a) AM^B e (b) AG^B , conforme o tamanho do lote.	83
24	Desempenho preditivo do <i>framework</i> usando modelos de seleção de atributos sobre a base de dados CIFAR-10: (a) AM^M e (b) AG^M , conforme o número de atributos selecionados.	86
25	Desempenho preditivo do <i>framework</i> usando modelos de seleção de atributos sobre a base de dados MNIST: (a) AM^M e (b) AG^M , conforme o número de atributos selecionados.	88

LISTA DE TABELAS

1	Bases de dados adotadas para a avaliação experimental do <i>framework</i> proposto.	58
2	Distribuição de classes da base de dados de imagens MNIST.	60
3	Configuração dos ambientes Google Colab e Local utilizados nos experimentos.	61
4	Versões das bibliotecas Python para os ambientes Google Colab e Local.	62
5	Blocos de experimentos realizados para o grupo <i>SA</i>	63
6	Blocos de experimentos realizados para o grupo <i>SA</i>	64
7	Desempenho preditivo dos métodos para a seleção de atributos na classificação binária na base de dados Food-5k.	104
8	Desempenho preditivo dos métodos para a seleção de atributos na classificação binária na base de dados CIFAR-10.	105
9	Desempenho preditivo dos métodos para a seleção de atributos na classificação binária na base de dados MNIST.	106
10	Desempenho preditivo dos métodos para a seleção de atributos na classificação multiclases na base de dados CIFAR-10.	107
11	Desempenho preditivo dos métodos para a seleção de atributos na classificação multiclases na base de dados MNIST.	108
12	Desempenho preditivo dos métodos de aprendizagem na classificação binária na base de dados Food-5k.	109
13	Desempenho preditivo dos métodos de aprendizagem na classificação binária na base de dados CIFAR-10.	110
14	Desempenho preditivo dos métodos de aprendizagem na classificação binária na base de dados MNIST.	111
15	Desempenho preditivo dos métodos de aprendizagem na classificação multiclases na base de dados CIFAR-10.	112

16	Desempenho preditivo dos métodos de aprendizagem na classificação multiclases na base de dados MNIST.	113
----	---	-----

LISTA DE SIGLAS

ADBN	<i>Adaptive Deep Belief Network</i>
AFD	Aprendizado em Fluxos de Dados
AM	Aprendizado de Máquina
APINC	Aprendizado Incremental
CNN	<i>Convolutional Neural Network</i> (Rede Neural Convolucional)
CV	<i>Computer Vision</i> (Visão Computacional)
DFS	<i>Deep Feature Selection</i>
DL	<i>Deep Learning</i> (Aprendizado Profundo)
DSDLA	<i>Deep Streaming Linear Discriminant Analysis</i>
FD	Fluxo de Dados
FSDS	<i>Feature Selection on Data Streams</i>
GPU	<i>Graphics Processing Unit</i> (Unidade de Processamento Gráfico)
HBP	<i>Hedge Backpropagation</i>
IA	Inteligência Artificial
ILDA	<i>Incremental Linear Discriminant Analysis</i>
IMS	<i>Image Mining System</i>
IoT	<i>Internet of Things</i> (Internet das Coisas)
LDA	Análise Discriminante Linear
OGSFS	<i>Online Group Streaming Feature Selection</i>
OFS	<i>Online Feature Selection</i>
DBN	<i>Deep Belief Network</i> (Rede de Crença Profunda)
RNA	Rede Neural Artificial
RNP	Rede Neural Profunda
SGD	<i>Stochastic Gradient Descent</i> (Gradiente Descendente Estocástico)
SVM	<i>Support Vector Machine</i> (Máquina de Vetor de Suporte)
TD	Transformação Digital
VFDT	<i>Very Fast Decision Tree</i>

LISTA DE SÍMBOLOS

AG	Acurácia Geral
AG^B	Acurácia Geral Binária
AG^M	Acurácia Geral Multiclasses
AM	Acurácia Média
AM^B	Acurácia Média Binária
AM^M	Acurácia Média Multiclasses
Q_t^{SA}	Quantidade de Atributos
Q_t^{AP}	Tamanho do Lote de Imagens

SUMÁRIO

1	Introdução	17
1.1	Motivação	18
1.2	Definição do Problema	19
1.3	Hipótese	19
1.4	Objetivos	20
1.5	Questões de Pesquisa	21
1.6	Desafios	21
1.7	Principais Contribuições	22
1.8	Organização da Tese	22
2	Classificação de Imagens	24
2.1	Imagem Digital	24
2.2	Processo de Classificação	24
2.3	Extração de Atributos	25
2.4	Reconhecimento	26
3	Aprendizado em Fluxos de Dados	28
3.1	Fluxos de Dados	28
3.2	Modelos de Fluxos de Dados	28
3.3	Técnicas para Processamento de Dados	29
3.4	Estratégias de Implementação e Simulação	31
3.5	Algoritmos de Classificação	33
4	Trabalhos Relacionados	36
4.1	Extração de Atributos de Imagens em Fluxos de Dados	36

4.2	Classificação de Imagens em Fluxos de Dados	38
4.3	Aprendizado Profundo em Fluxos de Dados	44
5	<i>Framework</i> para a Classificação de Imagens em Fluxos de Dados	49
5.1	Delineamento	49
5.2	Arquitetura do <i>Framework</i> Proposto	50
5.3	Processamento de Imagens como Fluxo de Dados	51
5.4	Mecanismo de Seleção de Atributos	52
5.5	Classificador de Imagens em Fluxos de Dados	53
6	Experimentos	55
6.1	Abordagem Experimental	55
6.2	Bases de Dados	57
6.3	Preparação dos Dados	60
6.4	Infraestrutura	61
6.5	Configuração Experimental	62
6.6	Hiperparâmetros da Rede Neural	65
6.7	Avaliação de Desempenho	66
7	Resultados e Discussão	69
7.1	Seleção de Atributos para Classificação Binária	69
7.2	Seleção de Atributos para Classificação Multiclasses	75
7.3	Aprendizado para Classificação Binária de Imagens	79
7.4	Aprendizado para Classificação Multiclasses de Imagens	85
7.5	Considerações Finais	89
8	Conclusão	92
8.1	Principais Resultados	93
8.2	Contribuições	95

8.3	Limitações	96
8.4	Trabalhos Futuros	97
	Referências	98
	Apêndice A – Resultados dos Experimentos	104

1 INTRODUÇÃO

A Transformação Digital (TD) tem modificado a vida em sociedade e em como as pessoas realizam suas tarefas diárias pessoais e de trabalho. As novas tecnologias originadas dessa transformação, como Big Data, Inteligência Artificial (IA), Internet das Coisas (IoT) e Computação em Nuvem, passaram a impactar na forma como as empresas pensam sua cultura, desenvolvem seus produtos e em como as pessoas interagem entre si. Diferentes segmentos de mercado, como Indústria, Agronegócio, Energia, Transportes, Logística e Cidades Inteligentes, estão se adaptando a essas novas tecnologias para melhorar processos, aumentar a eficiência e fornecer uma melhor experiência aos consumidores (MOHAMMADI et al., 2018).

Embora diversos benefícios possam ser listados, muitos desafios em relação ao uso dessas tecnologias também têm surgido, desde questões técnicas, como: a análise e interpretação de grandes volumes de dados, a transparência no uso de modelos estatísticos, o consumo de energia e a segurança e proteção contra ameaças cibernéticas; até questões humanas e sociais, como: a ética e a privacidade no tratamento de dados pessoais, e a responsabilização sobre o uso de algoritmos inteligentes e a automatização de decisões.

Entre as questões técnicas mais desafiadoras atuais está a produção massiva e não estacionária de dados de imagem gerados pela abundância de sensores, atuadores e comunicações no ambiente de IoT (JAMEEL et al., 2020). Esses dados referem-se ao grande volume de imagens que é gerado de forma contínua e em constante mudança por dispositivos capazes de capturar informações do ambiente ao seu redor.

Dispositivos equipados com câmeras ou sensores de imagem podem capturar imagens em tempo real e transmiti-las para sistemas de processamento e armazenamento de dados, fazendo com que as imagens possam ser geradas em grandes quantidades. Isso cria um fluxo contínuo de dados de imagem que precisa ser processado, armazenado e analisado de maneira eficiente e em tempo real.

Muitas tarefas em Visão Computacional (CV) podem ser desenvolvidas a partir do processamento de imagens em fluxos contínuos, como detecção de objetos, classificação de imagens digitais e reconhecimento facial. A análise desses dados em tempo real pode permitir a detecção de eventos, monitoramento de segurança, tomada de decisões em tempo real e fornecer *insights* valiosos para diferentes públicos.

No entanto, lidar com a produção massiva e não estacionária de dados de imagem também apresenta desafios, como a necessidade de recursos computacionais e de armazenamento escaláveis, além de algoritmos eficientes para processamento em tempo real e técnicas de análise de dados que possam lidar com a natureza dinâmica desses fluxos de dados.

1.1 Motivação

Uma forte motivação para a condução deste trabalho de pesquisa é a demanda por mais análises para a área de aprendizagem de imagens em fluxos de dados (GOMES et al., 2019). Desde o início dos anos 2000, os trabalhos na área sempre utilizaram dados estruturados ou desenvolveram aplicações analíticas. Porém, com o advento e expansão do conceito de Big Data, as novas pesquisas não poderão mais ignorar o uso massivo de dados não estruturados pelas pessoas e empresas.

Ainda que a perda de dados possa acontecer com dados que não forem processados em um fluxo de dados, eles poderão ser salvos e processados posteriormente. No entanto, a importância prática de lidar com a velocidade (*velocity*) associada a Big Data é a rapidez do processo de traduzir a entrada de dados em informações utilizáveis (NAJAFABADI et al., 2015).

Outra motivação é a necessidade de técnicas específicas para analisar dados de dispositivos sensores para IoT (MOHAMMADI et al., 2018). Uma vez que a maior parte desses dados são gerados em contextos de fluxos de dados, as técnicas tradicionais de análise de dados não são capazes de lidar com o formato e a velocidade com que eles chegam. Dessa forma, novas técnicas e algoritmos tornam necessária a condução de pesquisa científica atualizada.

Nesta tese, a tarefa de classificação de imagens é explorada juntamente com a estratégia de Aprendizado em Fluxos de Dados (AFD). Embora qualquer tarefa de Visão Computacional pudesse ser abordada neste trabalho, a classificação de imagens possui muitas vantagens práticas em relação às demais tarefas, como a própria popularidade da

tarefa, a fácil compreensão da tarefa por usuários não técnicos e a abundante quantidade de trabalhos e bases de dados de imagem disponíveis.

1.2 Definição do Problema

O problema de classificar imagens em um fluxo potencialmente infinito de dados consiste em encontrar um classificador \hat{f} , tal que $\hat{f} : X \mapsto \mathbb{R}^C$, onde C é o número de classes observadas em um conjunto não identicamente distribuído de distribuições desconhecidas de imagens $\mathcal{D} = \{D_1, \dots, D_N\}$ sobre $X \times Y$, sendo X e Y variáveis aleatórias de entrada e saída, respectivamente. Em um instante de tempo t , um algoritmo de aprendizagem recebe um mini lote de amostras (x_t^i, y_t^i) da distribuição D_i corrente e observa esse mini lote apenas uma vez para aprender a distribuição e construir o classificador \hat{f} (MAI et al., 2022).

A classificação de imagens é um problema fundamental e desafiador em Visão Computacional, pois requer a extração de características discriminativas das imagens e a capacidade de generalizar essas características para reconhecer uma nova imagem. Esse processo visa identificar padrões em uma imagem digital para adequá-la em uma classe particular. Em geral, é comum empregar técnicas de Aprendizado de Máquina (AM) para o reconhecimento do objeto presente na imagem.

Diferentes técnicas podem ser empregadas para a extração e generalização das características, mas o uso de algoritmos de Aprendizado Profundo (DL) tem prevalecido nesse domínio devido a um desempenho substancialmente melhor em comparação com métodos tradicionais de Visão Computacional (DAVIES, 2017; O'MAHONY et al., 2020).

1.3 Hipótese

O problema da classificação de imagens em fluxos de dados requer que as imagens sejam aprendidas por um modelo de decisão em tempo de processamento constante. Essa característica é importante para prevenir o esquecimento catastrófico, que é um fenômeno que impede o aprendizado de novos conceitos com a retenção de conceitos já observados. Além disso, o fenômeno deteriora o desempenho preditivo do modelo ao longo do tempo.

Hipótese: a combinação de redes neurais profundas e algoritmos para a aprendizagem em fluxos de dados deve permitir a classificação de imagens em fluxos de dados.

Modelos de Redes Neurais Artificiais (RNA) Profundas (RNP) são uma abordagem dominante em Visão Computacional (CV). Elas têm sido utilizadas para aumentar o desempenho preditivo de aplicações de Processamento de Imagens usando Big Data e abundantes recursos computacionais. Mais especificamente, as abordagens baseadas em Redes Neurais Convolucionais (CNN) têm sido consideradas úteis para lidar com dados de alta dimensionalidade devido à capacidade dessas redes em extrair atributos de baixo nível, como pixels, contornos e cores (JAMEEL et al., 2020). Por outro lado, elas não são capazes de aprender rapidamente objetos individuais ou terem seus modelos incrementalmente atualizados (HAYES; KANAN, 2020).

Algoritmos de Aprendizado em Fluxos de Dados (AFD) são capazes de trabalhar com grandes volumes de dados trafegando a altas taxas de velocidade, fornecendo respostas rapidamente atualizadas em qualquer momento e em tempo constante (DOMINGOS; HULTEN, 2001). A pesquisa nessa área deve desenvolver algoritmos e técnicas que possam lidar com fluxos de dados em tempo real e fornecer respostas rápidas para a tomada de decisão.

Portanto, modelos de RNPs devem facilitar a extração de atributos relevantes das imagens, gerando saídas que sejam utilizadas como entradas por algoritmos que aprendem em fluxos contínuos de dados.

1.4 Objetivos

O objetivo deste trabalho é implementar algoritmos para a classificação de imagens utilizando aprendizado profundo. O trabalho concentra-se na investigação de modelos e algoritmos para o processamento e a aprendizagem de imagens em fluxos de dados. Com o propósito de aprimorar o desenvolvimento de aplicações baseadas em processamento de imagens e visão computacional, os objetivos específicos deste trabalho são:

1. Identificar a literatura recente em Aprendizado Profundo em Fluxos de Dados, incluindo técnicas de processamento de imagens, modelos de seleção de atributos, e algoritmos para a aprendizagem e classificação de imagens;
2. Simular um fluxo de dados de imagens com foco na seleção de atributos para Aprendizado Profundo;
3. Propor um *framework* para a construção e validação de modelos de redes neurais profundas aplicados à classificação de imagens em fluxos de dados a partir das

características intrínsecas ¹ dos fluxos de dados e seus requisitos de processamento computacional ²;

4. Comparar diferentes métodos de classificação de imagens em fluxos de dados baseados em Redes Neurais Convolucionais; e
5. Avaliar o desempenho dos modelos construídos em diferentes tarefas de classificação de imagens em fluxos de dados.

1.5 Questões de Pesquisa

Para cumprir os objetivos especificados anteriormente, foram elaboradas as seguintes perguntas-chave para serem pesquisadas e respondidas:

- Q1: Como extrair atributos relevantes de imagens em fluxos de dados em tempo de processamento constante?
- Q2: Como construir modelos de classificação de imagens em fluxos de dados?
- Q3: Como avaliar modelos de classificação de imagens em fluxos de dados?
- Q4: É possível utilizar modelos de Aprendizado Profundo para classificar imagens em fluxos de dados?

1.6 Desafios

Muitos desafios podem ser encontrados na classificação de imagens em fluxos de dados, sobretudo com a adoção de modelos de redes neurais profundas. A combinação de diferentes áreas com desafios próprios pode originar em novos desafios, tanto de maneira restrita às áreas estudadas quanto de maneira ampla.

Esta tese enfrenta dois desafios principais. O primeiro deles é aprender modelos de redes neurais em tempo de processamento constante (HAYES; KANAN, 2020). À medida que modelos de redes neurais profundas crescem em tamanho e complexidade, as redes

¹São características dos fluxos de dados: sequência de itens contínua e ordenada; tempo real; não é possível controlar a ordem do fluxo; não é viável armazenar todo o fluxo (não se sabe o tamanho); grande volume de dados; alta velocidade de tráfego.

²São requisitos para aprendizagem de fluxos de dados: tempo constante de processamento por registro; quantidade limitada de memória; aprendizagem olhando os dados apenas uma vez; disponibilidade do modelo de aprendizagem a qualquer hora (e atualizado).

também se tornam mais exigentes computacionalmente. O aumento de recursos destinado ao reconhecimento de imagens pode inviabilizar a aprendizagem em fluxos de dados. Para conservar as características dos fluxos como propriedades intrínsecas, os algoritmos de aprendizagem devem atualizar o modelo de decisão em tempo constante.

O segundo desafio é atualizar o modelo de decisão sem esquecer o que foi aprendido (HAYES; KANAN, 2020; WOLFE; KYRILLIDIS, 2022), mitigando o fenômeno do esquecimento catastrófico em redes neurais. Muitas vezes, a atualização incremental pode impactar a aprendizagem do modelo neural e impedir que novos conceitos sejam aprendidos. A mitigação desse fenômeno deve prevenir que ele deteriore o desempenho do modelo de decisão.

Outros desafios existentes que esta tese poderia solucionar são: o impacto do tamanho, dimensão e qualidade das imagens observadas em fluxos de dados; o tempo adicional para a aprendizagem das características das imagens; e a quantidade suficiente de imagens para o treinamento do modelo em um fluxo contínuo de dados. Embora esses desafios possam ser abordados e tratados, eles não serão destacados neste trabalho.

1.7 Principais Contribuições

Os objetivos desta tese e as questões de pesquisa permitiram diferentes contribuições nas áreas de AFD e CV. Destacam-se:

- Uma revisão exaustiva de procedimentos para a extração e seleção de atributos de imagens com foco na aprendizagem em fluxos de dados;
- Uma discussão sobre o uso de modelos de redes neurais profundas em fluxos de dados;
- Um conjunto de análises sobre a eficiência de diferentes métodos de classificação de imagens baseados em redes neurais profundas; e
- A proposição de um *framework* completo para a aprendizagem e classificação binária e multiclases de imagens em fluxos de dados.

1.8 Organização da Tese

Ao longo deste trabalho, diversas características da classificação de imagens em fluxos de dados são discutidas. Os capítulos restantes estão organizados da seguinte forma:

O Capítulo 2 conceitua a imagem digital, apresenta o processo de classificação de imagens e sintetiza o procedimento tradicional para extração de atributos de imagens.

O Capítulo 3 conceitua fluxos de dados, apresenta os principais modelos de fluxos de dados e as técnicas mais comuns para processamento de fluxos de dados utilizadas algoritmos de AM. O capítulo ainda apresenta duas abordagens para a implementação e simulação de fluxos de dados, e a ideia base dos algoritmos de classificação, descrevendo os mais utilizados.

O Capítulo 5 apresenta a proposta de *framework* desta tese para a construção e validação de modelos de RNAs profundas aplicados à classificação de imagens em fluxos de dados. Inicialmente, o delineamento para o desenvolvimento da proposta é apresentado. A arquitetura do *framework* e seus mecanismos são detalhados na seções seguintes.

O Capítulo 4 responde às questões de pesquisa formuladas na Seção 1.5 por meio da revisão de literatura. Os principais e mais recentes trabalhos encontrados são discutidos para responder cada questão. Uma breve discussão ainda analisa as lacunas em relação à proposta desta tese.

O Capítulo 6 fornece os parâmetros experimentais utilizados para o desenvolvimento do *framework* proposto. As seções presentes no capítulo descrevem a abordagem experimental empregada no trabalho, as bases de dados de imagens adotadas para a experimentação e o processo de preparação de dados. Também são descritas a infraestrutura utilizada para os experimentos e a configuração dos hiperparâmetros dos algoritmos comparados, além das métricas de avaliação de desempenho.

O Capítulo 7 apresenta os resultados obtidos a partir do planejamento de experimentos. O capítulo apresenta e discute quatro grupos de experimentos separados em dois blocos experimentais. Cada bloco executa duas tarefas de classificação de imagens.

O Capítulo 8 conclui o presente trabalho, resumindo os principais resultados e apresentando as contribuições e limitações da pesquisa. O capítulo também discute trabalhos futuros correlatos às propostas apresentadas nesta tese.

2 CLASSIFICAÇÃO DE IMAGENS

Este capítulo descreve os conceitos fundamentais em classificação de imagens, objeto de estudo desta tese. A tarefa de classificação de imagens é apresentada a partir de seu problema e processo de solução. A última seção destaca o funcionamento da extração de atributos, que também é estudada nesta tese.

2.1 Imagem Digital

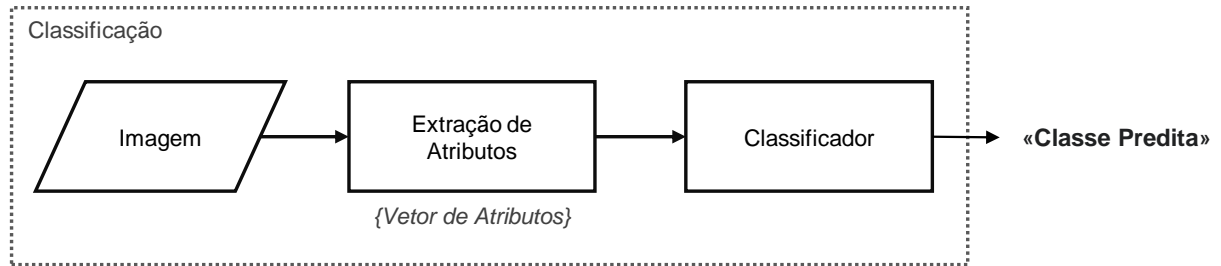
Uma **imagem** (ou imagem digital) é uma representação gráfica de dados processados. As imagens bidimensionais (2D) constituem a resposta de um sensor derivada de um sinal espacial contínuo bidimensional na forma de uma função bidimensional $f(x, y)$, em que x e y são coordenadas espaciais no plano. Os índices x e y indicam a quantidade de linhas e colunas da imagem, e os pixels são referenciados por seus índices (x, y) (GONZALEZ; WOODS, 2018).

2.2 Processo de Classificação

A **Classificação de Imagens Digitais** é uma das principais tarefas em CV, tendo como objetivo compreender o significado de uma imagem digital e categorizá-la dentro de um conjunto limitado de classes. A meta da classificação de imagens é construir um classificador que reconheça e discrimine uma imagem entre várias outras de categorias diferentes.

A Figura 1 ilustra o processo de classificação de uma nova imagem. Um sistema visual que implementa esse processo compreende três aspectos: a imagem a ser classificada; um mecanismo para extração de atributos da imagem, capaz de receber a imagem como entrada; e um classificador, responsável por compreender os atributos extraídos e reconhecer o objeto presente na imagem.

Figura 1: Processo de classificação de uma imagem.



Formalmente, $D_i = \{x_i, y_i\}$ é o conjunto de dados das imagens rotuladas, sendo x o conjunto das imagens observadas, e y o rótulo de classe atribuído às imagens observadas. Uma função desconhecida f aprende uma aproximação \hat{f} da função f que permite estimar o valor y_i para novas observações de x . Na tarefa de classificação de imagens, $y_i = f(x_i) \in \{c_1, \dots, c_m\}$, ou seja, $f(x_i)$ assume um dos valores do conjunto discreto não ordenado correspondente a uma ou mais classes c_i possíveis.

O classificador \hat{f} construído atribui uma imagem a uma das possíveis classes do problema. Se um novo conjunto de imagens $E = \{x_i\}$ é apresentado à função encontrada no processo de aprendizagem, a cada imagem x_i do conjunto será atribuído um rótulo pertencente a uma ou mais classes do problema.

2.3 Extração de Atributos

A **extração de atributos** procura transformar a imagem digital em uma representação numérica que possa ser compreendida pelo classificador. Os atributos da imagem consistem em características significativas que visam auxiliar o classificador na discriminação de classes do problema. Essas características constituem elementos na imagem, que formam regiões individuais e podem ser verificadas por técnicas de aprendizagem para o reconhecimento de objetos na representação gráfica.

Existem muitas abordagens para a extração de atributos de imagens digitais (DAVIES, 2017). Em uma imagem, um objeto é representado como uma coleção de pixels e o propósito do reconhecimento é descrever as propriedades desse grupo de pixels, chamados de descritores. As técnicas de detecção de descritores buscam por atributos que descrevem os limites da forma ou as regiões dos objetos identificados na imagem.

2.4 Reconhecimento

Também existem diferentes abordagens para aprender o classificador \hat{f} . Na classificação, essa etapa é chamada de **reconhecimento**, que busca por atribuir um rótulo identificador ao objeto encontrado na imagem.

Em geral, algoritmos tradicionais de AM são utilizados para o reconhecimento de imagens. A partir de um conjunto de imagens, algoritmos de AM buscam a solução por meio de um processo indutivo, em que conclusões genéricas são inferidas a partir de situações particulares. Para induzir hipóteses, algoritmos de AM utilizam um conjunto prévio de imagens agregando diferentes atributos representativos.

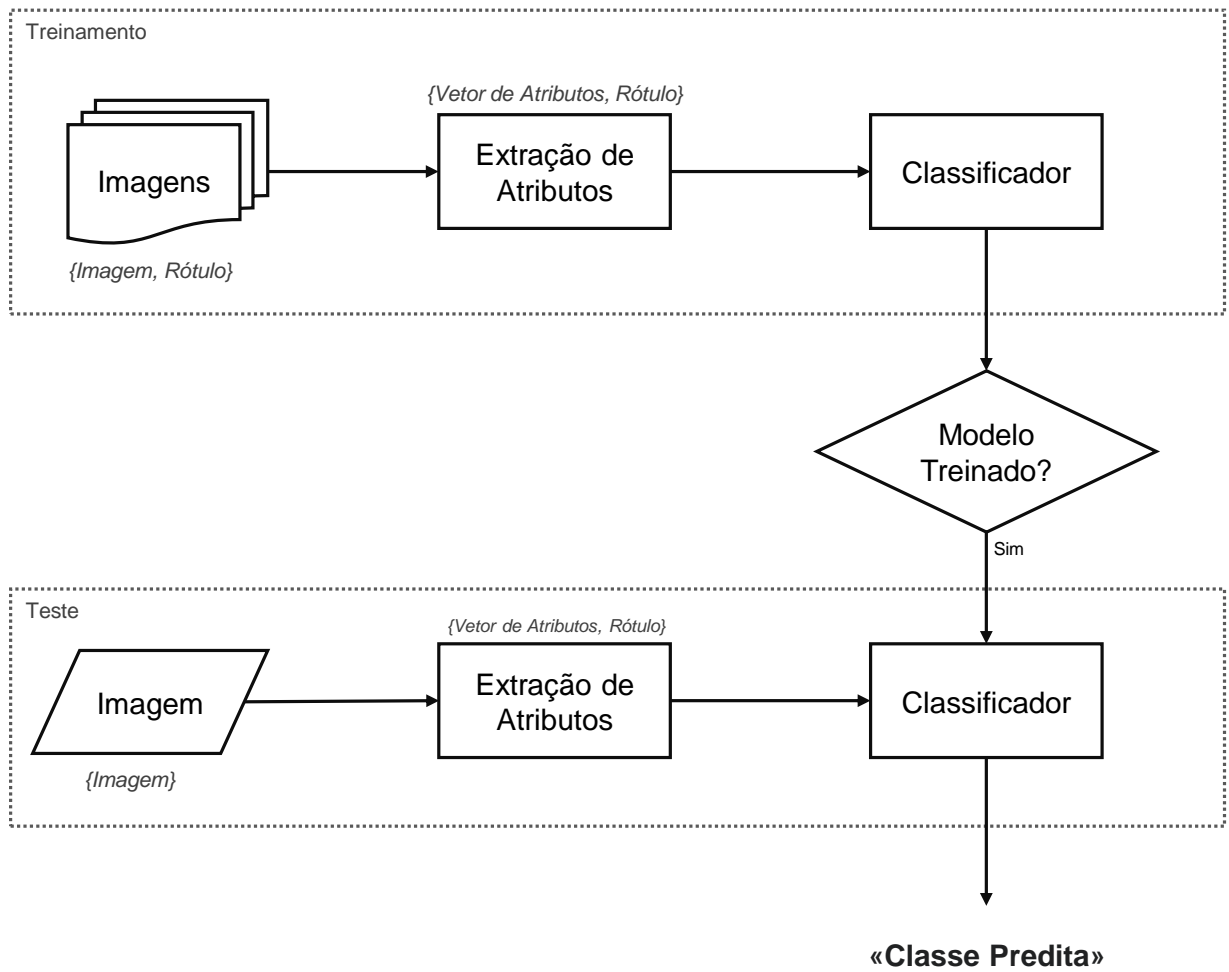
Algoritmos de AM supervisionados são especialmente adotados no reconhecimento de imagens (DAVIES, 2017). Nessa categoria de algoritmos, os conjuntos de dados possuem atributos de saída, cujos valores são obtidos a partir dos demais atributos, chamados de atributos de entrada. No caso das imagens, os atributos podem ser o próprio pixel da imagem ou serem extraídos a partir de diferentes técnicas de processamento de imagens.

Para encontrar a função \hat{f} , AM trabalha em duas fases: treinamento e teste. A fase de treinamento é uma etapa indutiva que consiste no aprendizado de uma hipótese que relaciona os atributos de entrada de um registro com seu respectivo atributo de saída (rótulo). Quando o algoritmo aprende uma hipótese válida também para os dados fora do subconjunto de treinamento, admite-se que essa hipótese possui capacidade de generalização.

A Figura 2 ilustra o processo de aprendizagem. Inicialmente, um vetor de atributos é extraído de um subconjunto de imagens escolhidas para fazer parte da fase de treinamento. Uma técnica de amostragem pode ser aplicada para a seleção dessas imagens. Em seguida, os rótulos derivados das classes do problema são atribuídos às imagens de treinamento para categorizá-las em uma ou mais classes. Com a construção do classificador pelo algoritmo de aprendizagem, uma nova imagem não rotulada poderá ser apresentada ao classificador para que ele encontre uma classe correspondente aos atributos extraídos da imagem.

Na segunda fase, durante a fase de teste, o algoritmo avalia o erro esperado da classificação na fase anterior. Se \hat{f} é uma função aproximada, os valores preditos y podem não ser os valores reais de cada observação x . Uma função de custo $L(y, f(x))$ compõe o processo de aprendizagem a fim de penalizar os erros de predição do algoritmo. Dessa forma, o algoritmo é capaz de escolher o melhor modelo com o menor erro de teste.

Figura 2: Processo de aprendizagem em duas fases: treinamento e teste.



3 APRENDIZADO EM FLUXOS DE DADOS

Este capítulo apresenta os conceitos que fundamentam a utilização de Aprendizado em Fluxos de Dados nesta tese. As seções seguintes exploram os conceitos de fluxos de dados com foco em Aprendizado de Máquina.

3.1 Fluxos de Dados

Um **Fluxo de Dados** (FD) pode ser visto como um conjunto de itens s_1, s_1, \dots que chegam sequencialmente descrevendo um sinal subjacente S , tal que $S : [1 \dots N] \rightarrow R$ seja uma função multidimensional. O fluxo de dados é uma sequência de sinais digitais codificados usados para representar uma informação em transmissão. Essa informação transmitida pode ser representada de várias formas, dando origem a diferentes modelos de fluxos de dados.

3.2 Modelos de Fluxos de Dados

O modelo mais trivial para representar um FD é o **Modelo de Séries Temporais**, onde cada s_i é $S[i]$, ou seja, s_i é o i -ésimo item de dado de S . Nesse modelo, apenas esse item de dado é processado. Recentes propostas têm adaptado esse modelo para codificar séries temporais em imagens, utilizando modelos de redes neurais profundas para aprender atributos relevantes de alto nível, como formas, objetos e cenários (WANG; OATES et al., 2015).

O **Modelo de Caixa Registradora** (GILBERT et al., 2001) é um pouco mais sofisticado e também o mais utilizado na representação de fluxos de dados analíticos. No modelo, os itens que chegam são valores de domínio sem uma ordenação particular, sendo S uma função para agregar o número de itens com valores de domínio iguais. O modelo é análogo ao funcionamento de uma caixa registradora, em que cada compartimento da gaveta é destinado a uma cédula particular de dinheiro. Para cada compartimento ou para toda

a gaveta, é possível calcular diferentes operações, como computar a quantidade de notas distintas ou o valor total presente em determinado compartimento.

No **Modelo Catraca** (MUTHUKRISHNAN et al., 2005; AI et al., 2016), s_i^j é uma tupla (i, w_j) tal que i é o item de dado e w_j é um vetor normal unitário positivo ou negativo para a j -ésima atualização do fluxo de dados. O fluxo de dados S é inicializado como $\vec{0}$ e é caracterizado por atualizações de forma que $s_i \leftarrow s_i + w_j$. Essa característica incremental permite que o modelo seja estudado em situações dinâmicas, tal como a entrada e saída de passageiros de uma estação de metrô, o monitoramento de tráfego em rodovias, o rastreamento de objetos em movimento e a navegação de robôs móveis.

3.3 Técnicas para Processamento de Dados

Modelos específicos para consulta e mineração de dados foram propostos para lidar com as restrições computacionais requeridas para o processamento de fluxos de dados. Em geral, esses modelos utilizam abordagens que analisam de maneira parcial o conjunto de dados completo. Isto é realizado por meio da análise de um subconjunto de dados do conjunto de dados maior ou pela utilização de uma representação dos dados em menor tamanho. O objetivo é reduzir o tempo de processamento e consumo de memória dos algoritmos. Algumas dessas técnicas são apresentadas a seguir.

A **Amostragem** consiste em selecionar um subconjunto de dados do conjunto de dados original a partir de algum parâmetro. Em geral, esse parâmetro é uma função aleatória que decide se determinado item de dado será processado ou não. Aplicado a um FD, esse fluxo torna-se dispensável, pois se passa a trabalhar com instâncias menores, em vez do fluxo inteiro. Para fazer uma análise precisa, a amostragem requer muitos dados (MUTHUKRISHNAN et al., 2005). Embora a adoção dessa abordagem resulte em diversas estatísticas sobre o fluxo de dados, ela pode não ser aplicável onde se exija baixa taxa de erros.

A **Amostragem Aleatória** é uma variação da amostragem, onde todos os itens de dado têm a mesma probabilidade de serem processados. Uma amostragem mais sofisticada e que tem servido de base para outras técnicas é a **Amostragem Reservoir** (VITTER, 1985). A ideia é observar o fluxo de dados que corre e coletar amostras aleatórias de k registros sem reposição. Cada elemento do fluxo de dados tem probabilidade k/n de ser escolhido, onde n corresponde à quantidade de elementos vistos até o momento. A

técnica utiliza espaço constante de memória e realiza todo o processamento em uma única varredura.

A **Amostragem Bernoulli** é outra técnica de amostragem aleatória uniforme (HU; ZHANG, 2012), onde o tamanho da amostra segue a distribuição binomial. Nessa amostragem, cada item de dado tem uma probabilidade q de ser incluído na amostra e uma probabilidade $1-q$ de ser excluído dela, independentemente da quantidade de itens. Embora o processo de coleta seja simples e de baixo custo, a maior desvantagem é a ausência de controle sobre a variabilidade do tamanho da amostra.

Histogramas realizam análises sobre a distribuição dos dados do fluxo com rapidez, embora sem grande precisão (SILVA JUNIOR; LORENA, 2014). Uma versão eficiente dessa técnica é o **Histograma Exponencial**, que constrói intervalos pequenos para dados recentes e intervalos mais longos para dados mais antigos (DATAR et al., 2002). Outra vantagem dessa versão é a possibilidade de utilizá-la em dados multivariados. Histogramas são úteis na aproximação de valores correntes do fluxo passante em funções agregadas, permitindo observar o resultado a qualquer momento com erro limitado.

Sketching é uma técnica de geração de esboços (*sketches*), que são vetores gerados a partir da projeção de variáveis aleatórias sobre o conjunto de atributos. O resultado do esboçamento é um subconjunto de tamanho menor. Essa é uma técnica espacialmente eficiente, mas que apresenta baixa acurácia quando dispensa atributos relevantes. Tem sido utilizada para estimativas do fluxo de dados (MUTHUKRISHNAN et al., 2005).

Janelamento (BAHRI et al., 2021) é uma técnica que analisa os itens de dados mais recentes do fluxo. O algoritmo começa analisando o FD, processando os novos itens que chegam e descartando simultaneamente os itens antigos. Embora a técnica ignore partes do fluxo, ela previne que dados antigos influenciem a análise, mantendo as estatísticas sempre atualizadas. Por essa característica de descarte de conceitos, a técnica de janelamento é muito útil onde há limitação de memória.

Modelos de janelamento podem incluir técnicas de amostragem (NGUYEN et al., 2021). O **Janelamento Baseado em Sequências** adapta a amostragem *Reservoir* ao janelamento e aloca os n primeiros itens de dados em memória, mantendo uma amostra do fluxo, que permanece por um período de tempo suficiente para ser analisado.

Oposta a essa abordagem, a técnica de **Janelamento Deslizante** (DATAR et al., 2002) não aloca janelas de tamanho fixo, mas que variam conforme a duração de um intervalo de tempo t , previamente definido para selecionar todos os itens de dado do período corrente. Em geral, o intervalo de tempo é medido em *timestamps*. A técnica atribui aleatoriamente

uma prioridade a cada item de dado e, em seguida, escolhe aquele item com prioridade mais alta para fazer parte de uma amostra. Assim, são mantidos na memória apenas os *timestamps* mais recentes com alta prioridade. Esse deslizamento é implementado por meio de histogramas, que armazenam as estatísticas em tempo real.

Para processar dados sensíveis ao tempo e discriminá-los de acordo com seu tempo de criação, a técnica de **Janelamento Amortecido** usa uma função que atualiza periodicamente os pesos dos itens dentro da janela. A ideia é atribuir um peso para cada item de dado. A função de amortecimento usa um número real entre 0 e 1, atribuindo valor máximo a novos itens. Ao contrário das demais, essa abordagem não descarta os dados e diminui a influência de itens antigos (YUN et al., 2018).

Microagrupamento (AGGARWAL et al., 2003) é uma abordagem moderna que utiliza a ideia de microgrupos (*micro-clusters*) para armazenar informações estatísticas sobre os dados que passam pelo fluxo de maneira *online*. O objetivo é manter estatísticas em um nível suficientemente alto sobre a granularidade temporal e espacial dos dados que evoluem com o tempo. Os dados são utilizados por um componente *offline*, cuja principal função é fornecer uma rápida análise do fluxo de dados quando solicitada.

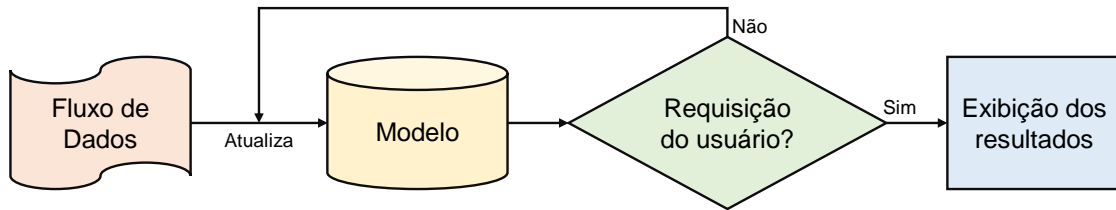
3.4 Estratégias de Implementação e Simulação

Existem duas estratégias (NGUYEN; WOON; NG, 2015) para a implementação de métodos de aprendizagem em um sistema baseado em fluxos de dados. Na **abordagem incremental**, os algoritmos aprendem um modelo de decisão ao longo do tempo, sendo capazes de se adaptar às mudanças que acontecem nos dados de entrada. Esses algoritmos podem aprender informações adicionais de novos dados, não necessitam acessar os dados originais de treinamento, preservam o conhecimento adquirido e se adequam às novas classes que podem surgir com novos dados.

Essa abordagem permite obter os resultados instantaneamente porque o modelo se mantém constantemente atualizado. A Figura 3 ilustra como a abordagem pode ser implementada. O modelo permanece sendo atualizado pelo fluxo de dados de entrada até que um resultado seja requisitado por um usuário. Uma desvantagem dessa implementação é que ela requer mais recursos computacionais.

Na **abordagem *online-offline*** (ou em duas fases) (AGGARWAL et al., 2003), o processo de aprendizado é dividido em duas etapas. Na primeira etapa, o algoritmo coleta, processa e armazena uma pequena porção de dados derivados do fluxo de dados de entrada.

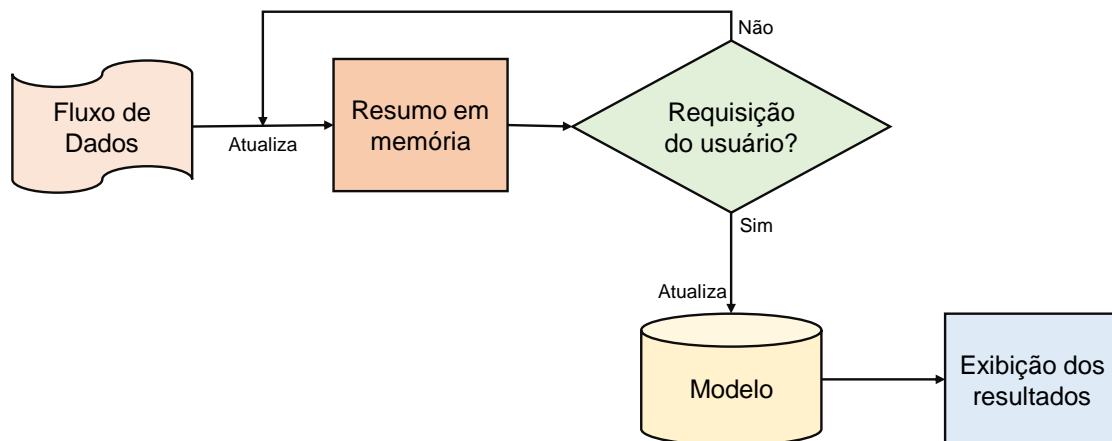
Figura 3: Simulação de fluxos de dados seguindo a abordagem incremental.



Essa porção de dados é continuamente atualizada e armazenada em memória. Na segunda etapa, o algoritmo processa as informações em modo *offline*. Quando um usuário solicita um resultado, o algoritmo interrompe a atualização dos dados, recupera a porção de dados armazenada em memória e atualiza o modelo. O resultado do modelo é exibido ao usuário.

A Figura 4 ilustra como um sistema baseado em fluxos de dados pode implementar a abordagem *online-offline*. A principal vantagem dessa abordagem é que os dados podem ser processados em mais alta velocidade, já que nenhum modelo precisa ser atualizado. Por outro lado, os resultados podem ser menos precisos e o usuário deverá esperar até que eles fiquem disponíveis.

Figura 4: Simulação de fluxos de dados seguindo a abordagem *online-offline*.



Nos últimos anos, a abordagem *online-offline* ganhou mais destaque em relação a outras abordagens por dois motivos. O primeiro é em razão da necessidade de separar técnicas incrementais e aquelas baseadas em *streaming*. O segundo motivo é devido às altas taxas com que os dados de entrada trafegam no fluxo. A manutenção do controle do FD de entrada é uma das características que permite que tarefas usuais de AM, como classificação e agrupamento de dados, possam ser implementadas com menos complexidade.

3.5 Algoritmos de Classificação

Em Aprendizado de Máquina, as técnicas tradicionais podem tratar grandes volumes de dados, mas não são capazes de trabalhar diretamente com dados que trafegam continuamente. A maior parte desses algoritmos é capaz de analisar apenas pequenos conjuntos de dados estáticos. A evolução contínua do volume de dados torna difícil a criação de um modelo de aprendizado preciso e consistente.

O desenvolvimento de novos métodos de aprendizado em fluxos de dados requer que os algoritmos processem cada item de dado em tempo pequeno e constante, utilizem quantidade limitada de memória primária e construam o modelo de aprendizado em uma única varredura. O modelo também deve poder ser fornecido de maneira atualizada em qualquer instante de tempo (DOMINGOS; HULTEN, 2001).

Os algoritmos que adotam esses critérios procuram focar na manutenção do controle do fluxo de dados. Essa estratégia permite que o modelo de aprendizado seja construído rapidamente utilizando pouca informação. Ainda que a quantidade de informações coletadas inicialmente possa não contribuir para a geração de um modelo consistente, o processamento contínuo dos dados em trânsito permitirá que um modelo mais aprimorado seja construído ao longo do tempo.

Uma das formas de executar essa estratégia é implementar técnicas que calculam estatísticas sobre o fluxo passante. Essas estatísticas são utilizadas para análise e processamento dos dados. Muitos algoritmos de aprendizado têm adotado essa estratégia para aprender (DOMINGOS; HULTEN, 2000; AGGARWAL et al., 2003). Dessa forma, os algoritmos podem processar os dados usando os poucos recursos computacionais disponíveis e entregando respostas mais rápidas.

Apesar da imprecisão da resposta, há grande vantagem em utilizar essa abordagem. Como os algoritmos trabalham com uma pequena faixa de erro, à medida que os erros diminuem, o espaço para a alocação de recursos computacionais aumenta, garantindo que as restrições para processamento dos fluxos continuem sendo satisfeitas ao longo de todo o processo de aprendizado.

O sistema *Very Fast Decision Tree (VFDT)* (DOMINGOS; HULTEN, 2000) é baseado em árvores de decisão e continua sendo um dos principais algoritmos para classificação de fluxos de dados. Para construir a árvore, o VFDT utiliza um pequeno conjunto de dados para selecionar o teste de divisão a ser incorporado em um nó. Se a diferença entre os dois melhores testes de divisão não satisfizaz uma medida estatística, o algoritmo

continua examinando o fluxo de dados, extraindo outra amostra e repetindo o processo. Os testes de divisão possuem uma medida que aponta quão bem um atributo é corretamente escolhido. Essa medida consiste em uma função de avaliação que computa o mérito dos testes de divisão, calculados conforme os valores dos atributos, que armazenam estatísticas suficientes para esse cálculo.

O VFDT é um sistema computacionalmente eficiente. A utilização da medida de Hoeffding garante ao sistema sua rapidez, uma vez que é um critério nítido para a tomada de decisão quanto ao crescimento da árvore e as divisões que ocorrem nela durante o processo de aprendizado. O tempo de processamento de um item de dado é menor que o tempo necessário para sua leitura em memória secundária. Além disso, a limitação de memória, necessária aos algoritmos de fluxos de dados, é tratada pela destituição de atributos fracos e pela desativação das folhas menos promissoras, que abrem caminho para novas folhas.

O **Classificador *On-Demand-Stream*** (AGGARWAL et al., 2004) implementa a técnica de microagrupamento para encontrar a melhor janela de tempo que deve ser utilizada para aperfeiçoar a precisão da classificação. O algoritmo isola os processos de classificação dos dados de treinamento e dos dados de teste, executando-os simultaneamente. Em tempo real, esse tratamento classifica o fluxo de teste ao mesmo tempo que os fluxos de treinamento e teste evoluem. Ambos os processos de classificação compartilham a componente *online* do sistema. Na classificação do fluxo de treinamento, os dados de entrada são divididos em duas partes. Na primeira parte, uma pequena porção do fluxo de dados é coletada para ser utilizada na criação dos microgrupos, sendo os dados remanescentes utilizados para coleta de estatísticas sobre os microgrupos e informações da classe.

Ensembles também estão presentes na classificação de fluxos de dados. O algoritmo ***Adaptive Windowing (Adwin) Bagging*** (BIFET et al., 2009) é uma implementação *online* do algoritmo *Bagging* usando o método *Adwin* (BIFET; GAVALDA, 2007) como um detector de mudanças. O método *Adwin* utiliza uma janela de tamanho variável W para armazenar os itens mais recentes que foram processados e decidir em que momento os membros dos *ensembles* devem ser descartados. Utilizando um histograma exponencial, apenas $O(\log W)$ é consumido pela memória e também para o tempo de processamento de cada item de dado. Uma árvore de Hoeffding ainda pode ser utilizada como classificador-base para o *ensemble*.

Mais recentemente, classificadores incrementais baseados em redes neurais têm surgido como uma alternativa para a aprendizagem em fluxos de dados. Uma versão modificada

do Perceptron possui um conjunto de pesos que é atualizado para cada novo item recebido pelo fluxo de dados usando o método do Gradiente Descendente Estocástico (SGD) (BIFET et al., 2010). Essa abordagem é adotada para que o algoritmo realize uma única passada, em vez de múltiplas iterações, para aumentar a acurácia preditiva.

Em geral, modelos são treinados em conjunto para partes sequenciais do fluxo de dados e, em seguida, ponderados com base na acurácia esperada no fluxo de dados de teste. Essa abordagem melhora constantemente o modelo e a acurácia de cada classificador, o que a torna própria para ambientes evolutivos. No entanto, muitos desafios ainda existem nessa categoria de algoritmos em razão da natureza potencialmente infinita dos dados e das limitações de memória e tempo de processamento dos FDs (BESEDIN et al., 2017).

4 TRABALHOS RELACIONADOS

Este capítulo identifica a literatura existente em DL para FDs, e discute os trabalhos relacionados à proposta desta tese. As seções seguintes estão organizadas de acordo com o assunto interrogado por uma ou mais questões de pesquisa apresentadas na Seção 1.5.

4.1 Extração de Atributos de Imagens em Fluxos de Dados

Esta seção visa responder à questão de pesquisa *Q1: Como extrair atributos relevantes de imagens em fluxos de dados em tempo de processamento constante?*. O principal desafio da questão é selecionar atributos significativos para a tarefa de aprendizado sem afetar o desempenho computacional geral do processo de classificação. Embora a literatura específica da área de Processamento de Imagens faça distinção entre os termos *extração* e *seleção*, neste texto eles são utilizados com o mesmo significado.

Existem muitas abordagens para a extração de atributos de imagens digitais. Tradicionalmente, técnicas de Processamento de Imagens e Visão Computacional são utilizadas por especialistas para obter manualmente as características das imagens observadas em um conjunto de dados. Nessa abordagem, um objeto é representado como uma coleção de pixels e o propósito do reconhecimento é descrever as propriedades desse grupo de pixels, chamados de descritores.

As técnicas de detecção de descritores buscam por atributos que descrevem os limites da forma ou as regiões dos objetos identificados na imagem. Embora o uso de descritores continue sendo adotado por aplicações específicas no reconhecimento de objetos, ainda não existe um descritor que seja capaz de caracterizar qualquer tipo de objeto (O'MAHONY et al., 2020).

Nesta tese, as técnicas tradicionais para extração de atributos de imagens não são empregadas, uma vez que os algoritmos implementados atuam em bases de dados com múltiplos objetos e sem aplicação específica. Dessa forma, a adoção de técnicas de Aprendizado

Profundo para a extração de atributos contempla parte da hipótese para a classificação de imagens em fluxos de dados.

O paradigma de Aprendizado Profundo introduziu o conceito de **Aprendizado de Ponta a Ponta** (*End-to-End Learning*) (CASTRO et al., 2018), em que o algoritmo de aprendizagem recebe apenas um conjunto de dados de imagens rotuladas e o modelo de rede neural é treinado nos dados observados. Em seguida, as redes neurais descobrem os padrões subjacentes em classes de imagens e constroem automaticamente os atributos mais descritivos e relevantes com relação a cada classe para cada objeto (O'MAHONY et al., 2020). Alguns modelos específicos para a extração de atributos em fluxos de dados foram propostos e são descritos a seguir.

O modelo **Deep Feature Selection (DFS)** (LI; CHEN; WASSERMAN, 2016) utiliza Aprendizado Profundo para selecionar atributos relevantes na identificação de sequências de DNA em fontes múltiplas de informação em expressões genômicas. O algoritmo propõe a inclusão de uma camada linear esparsa um-para-um entre a camada de entrada e a primeira camada escondida de uma rede Perceptron Multicamadas. Essa camada adicional atua como uma interface, gerando uma saída esparsa onde apenas os atributos com valores não nulos são selecionados. Como o modelo sempre seleciona um único subconjunto de atributos e trabalha em tempo constante, o algoritmo pode ser aplicado à classificação em fluxos de dados.

No modelo **Online Feature Selection (OFS)** (WANG et al., 2014), a tarefa é aprender um classificador online $w^{(t)}$ que pode ser usado para classificar cada novo registro x_i por uma função linear $\text{sign}(w^{(t)' } x_i)$. A seleção de atributos acontece quando o classificador $w^{(t)}$ tem, no máximo, B elementos diferentes de zero com $\|w^{(t)}\|_0 \leq B$, forçando a seleção de B atributos. Embora esse modelo não tenha sido aplicado ainda à classificação de imagens, ele também é capaz processar em tempo constante e real em razão da linearidade das funções de classificação utilizadas.

A proposta do modelo **Scalable and Accurate OnLine Approach (SAOLA)** (YU et al., 2016) é selecionar os atributos de forma escalável lidando com dados de dimensionalidade extremamente alta. O algoritmo emprega a comparação por pares e mantém um modelo parcimonioso¹ ao longo do tempo de maneira online para garantir a escalabilidade.

¹Em Estatística e Aprendizagem de Máquina, um modelo parcimonioso é um modelo de decisão que busca ser construído com a menor quantidade de atributos sem perder a capacidade de representação e generalização.

O modelo *Feature Selection on Data Streams (FSDS)* (HUANG; YOO; KASIVISWANATHAN, 2015) é proposto para selecionar automaticamente os atributos mais relevantes que são observados em um fluxo de dados em apenas uma varredura e com uso limitado de memória. O algoritmo usa a técnica de esboço matricial (LIBERTY, 2013) para manter uma aproximação de posto baixo dos dados que estão sendo observados no momento. Em seguida, o algoritmo aplica um método de regularização de regressão para obter os coeficientes dos atributos. Na análise de regressão, esses coeficientes podem ser interpretados como pesos da reta de regressão e ponderar a importância dos atributos na aprendizagem. Essa abordagem é utilizada na classificação binária de imagens da base de dados MNIST.

O modelo *Online Group Streaming Feature Selection (OGSFS)* (ZHOU; WANG; ZHAO, 2021) assume que nem todos os atributos podem ser apresentados antes da aprendizagem. Conjuntos de dados de imagem costumam apresentar essa propriedade, uma vez as imagens podem ter dimensões diferentes. O algoritmo proposto pode ser dividido em duas partes: seleção online intragrupos e seleção online intergrupos. Na primeira parte, pares de atributos interativos são escolhidos a partir da teoria da informação mútua. Em seguida, uma regressão Elastic Net é utilizada para diferentes grupos, que tende a selecionar mais fortemente atributos correlatos. Ao contrário dos demais modelos, esse algoritmo prioriza a seleção de atributos em grupos.

Discussão: o principal desafio da extração de atributos é o processamento eficiente das imagens observadas. Métodos lineares são capazes de selecionar atributos relevantes das imagens em tempo constante. Modelos de redes neurais profundas são mais inteligentes na seleção de atributos. A combinação de ambas as abordagens pode resultar em uma aprendizagem incremental *online* mais inteligente e menos custosa computacionalmente.

4.2 Classificação de Imagens em Fluxos de Dados

Esta seção visa responder a duas questões de pesquisa. A primeira delas é **Q2: Como construir modelos de classificação de imagens em fluxos de dados?** O principal desafio é entender como sistemas de Aprendizado em Fluxos de Dados processam imagens. Ao contrário da abordagem tradicional, que trabalha com dados estruturados, os algoritmos de aprendizado podem não compreender facilmente dados de imagem. Dessa forma, é desejável investigar como os métodos de classificação tratam os dados de entrada para treinamento do modelo de decisão.

A segunda questão é **Q3: Como avaliar modelos de classificação de imagens em fluxos de dados?**, que apresenta dois desafios. O primeiro desafio é investigar uma metodologia experimental eficiente que observe os requisitos intrínsecos dos fluxos de dados ao lidar com imagens. Essa metodologia deve compreender as diferentes etapas do processo de classificação de imagens.

O segundo desafio corresponde ao processo de avaliação de desempenho. O problema é investigar a existência de diferenças ou particularidades na avaliação da classificação de imagens em fluxos de dados em relação à classificação de imagens tradicional. Também é importante listar as métricas comumente adotadas pela literatura na área.

Inicialmente, são identificadas abordagens tradicionais para a classificação de imagens em fluxos de dados. Uma ideia básica inicial é utilizar métodos multivariados para reduzir o espaço de atributos da aprendizagem. A Análise Discriminante Linear (LDA) é um desses métodos que tem sido bastante explorado. A estratégia é encontrar uma transformação linear que maximize a separação entre as classes enquanto minimiza a dispersão dentro de cada classe. Em geral, essa abordagem é adotada com uma configuração incremental.

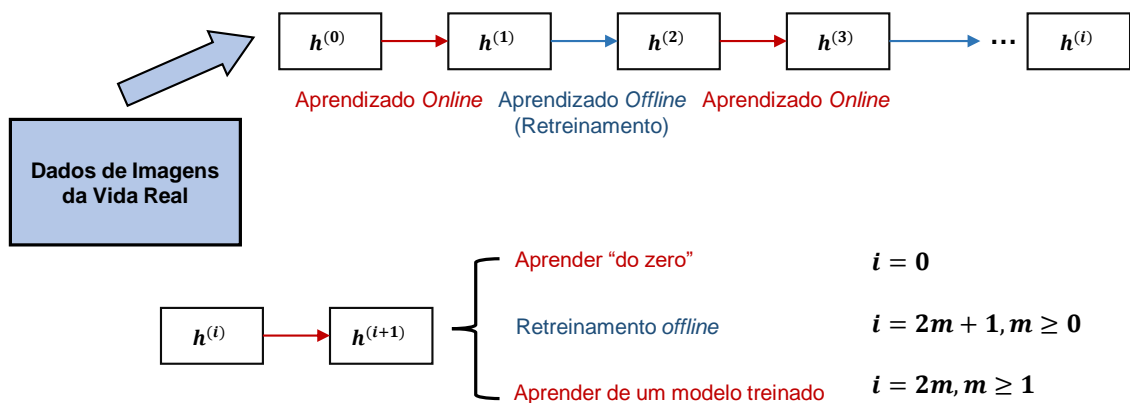
O método ***Incremental Linear Discriminant Analysis (ILDA)*** (PANG; OZAWA; KASABOV, 2005) utiliza essa ideia para extrair atributos com alta relevância e construir um autoespaço discriminante sobre o fluxo de dados. Uma versão atualizada desse método computa a Fatoração QR da matriz de dados e resolve o sistema linear triangular inferior resultante da fatoração (CHU et al., 2015). Consequentemente, o novo método se torna capaz de lidar mais facilmente com novos itens ou blocos de dados, alcançando resultados melhores na classificação. Ambos os métodos são experimentados na tarefa de classificação de imagens e mostram como sistemas não tradicionais lidam com imagens.

O método **iCaRL** (REBUFFI et al., 2017) também emprega uma estratégia incremental para simultaneamente aprender classificadores e representações de atributos. A arquitetura do algoritmo utiliza uma Rede Neural Convolutiva para treinar lotes de classes usando Aprendizado Incremental (APINC). Toda vez que os dados de novas classes estiverem disponíveis, o método chama uma rotina de atualização que ajusta o conhecimento interno da rede a partir das informações adicionais dos novos dados. Na classificação, conjuntos de imagens são selecionados dinamicamente do fluxo de dados. Para cada conjunto, chamado de conjunto exemplar, e para cada classe do conjunto, o método garante que o número total de imagens exemplares não será maior que o parâmetro fixo K correspondente ao tamanho da memória. Para prever o rótulo de uma nova imagem, iCaRL usa a média mais próxima dos exemplos.

Embora o APINC seja capaz de processar fluxos de dados, eles ainda podem processar o item de dado no fluxo passante mais de uma vez, consumindo mais recursos computacionais. Métodos de Aprendizado *Online* podem ser úteis na combinação de novas abordagens incrementais, pois são rápidos para treinar, consomem pouca memória e conseguem lidar com dados que chegam sequencialmente. Modelos *online* são capazes de aprender classes individuais à medida que novas imagens aparecem no sistema.

Uma recente proposta considera a aplicação de APINC em cenários *online* (HE et al., 2020). A Figura 5 ilustra o *framework* dessa proposta, que constrói um modelo evolutivo $h^{(i)}$ capaz de lidar de maneira sequencial com dados de novas classes e novas observações de classes antigas. O *framework* possui três partes: aprender do zero; retreinamento *offline*; e aprender de um modelo treinado, que implementa o APINC. As duas últimas partes podem alternadas para alcançar a aprendizagem *online*.

Figura 5: *Framework* de aprendizagem incremental para cenários *online*.

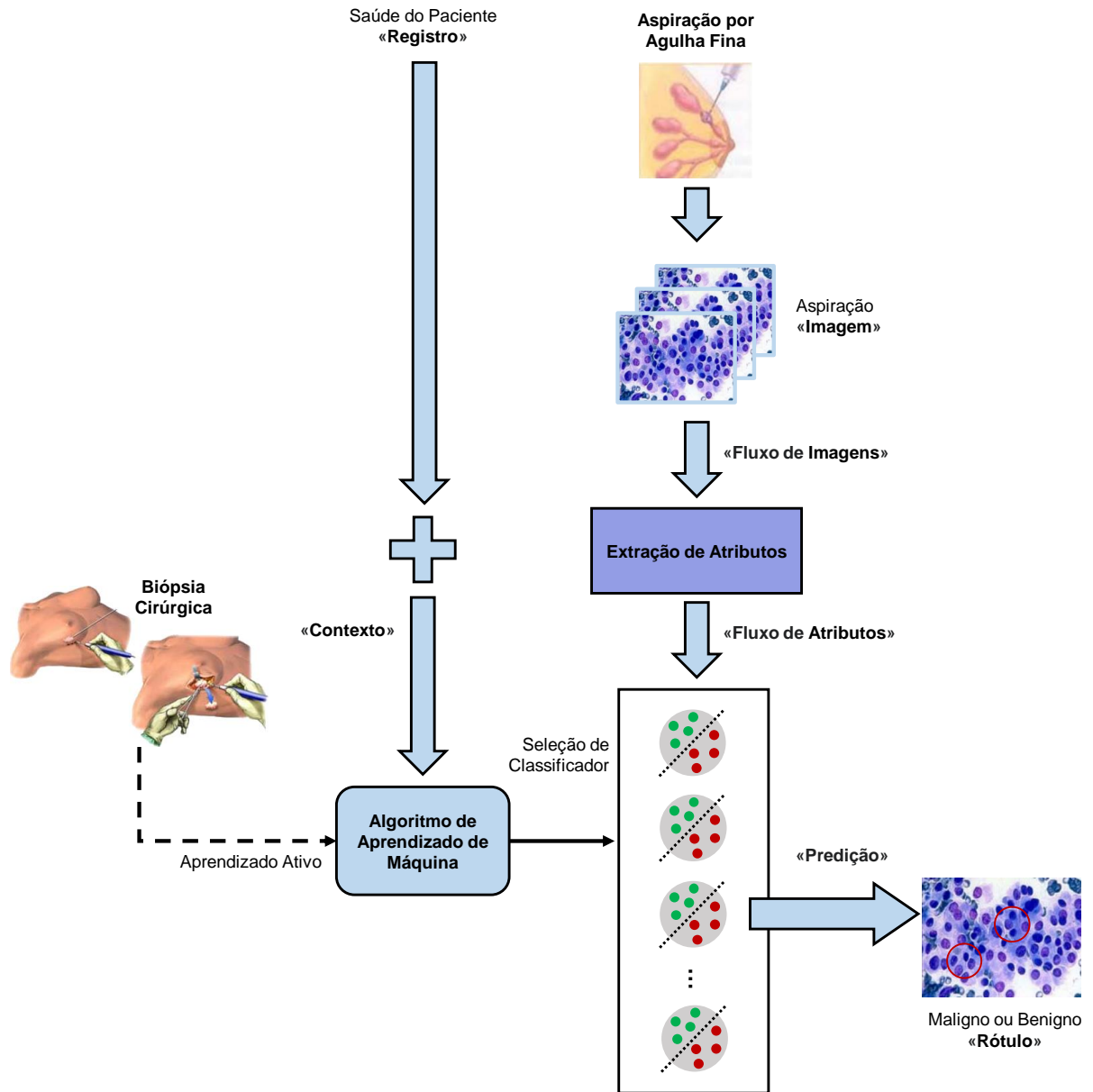


Fonte: Traduzido e Adaptado de (HE et al., 2020).

A metodologia do sistema *Image Mining System (IMS)* (TEKIN; SCHAAR, 2015) considera todo o *pipeline* de classificação de imagens para aprender em fluxos de dados. Inicialmente, o sistema recebe as imagens sequencialmente, com cada imagem sendo acompanhada por um contexto. Esse contexto representa metadados que podem fornecer informações adicionais sobre a imagem. As imagens são pré-processadas e têm seus atributos extraídos. O processamento acontece de maneira *online* e um classificador genérico é escolhido a partir do contexto das imagens para realizar a predição.

A Figura 6 ilustra o funcionamento do sistema IMS. No sistema, o treinamento utiliza informações contextuais para construir atributos mais significativos. A meta é selecionar o melhor conjunto de classificadores para o diagnóstico de câncer de mama. A predição é realizada de maneira direta, sendo baseada no contexto da aplicação.

Figura 6: Arquitetura do Sistema IMS.



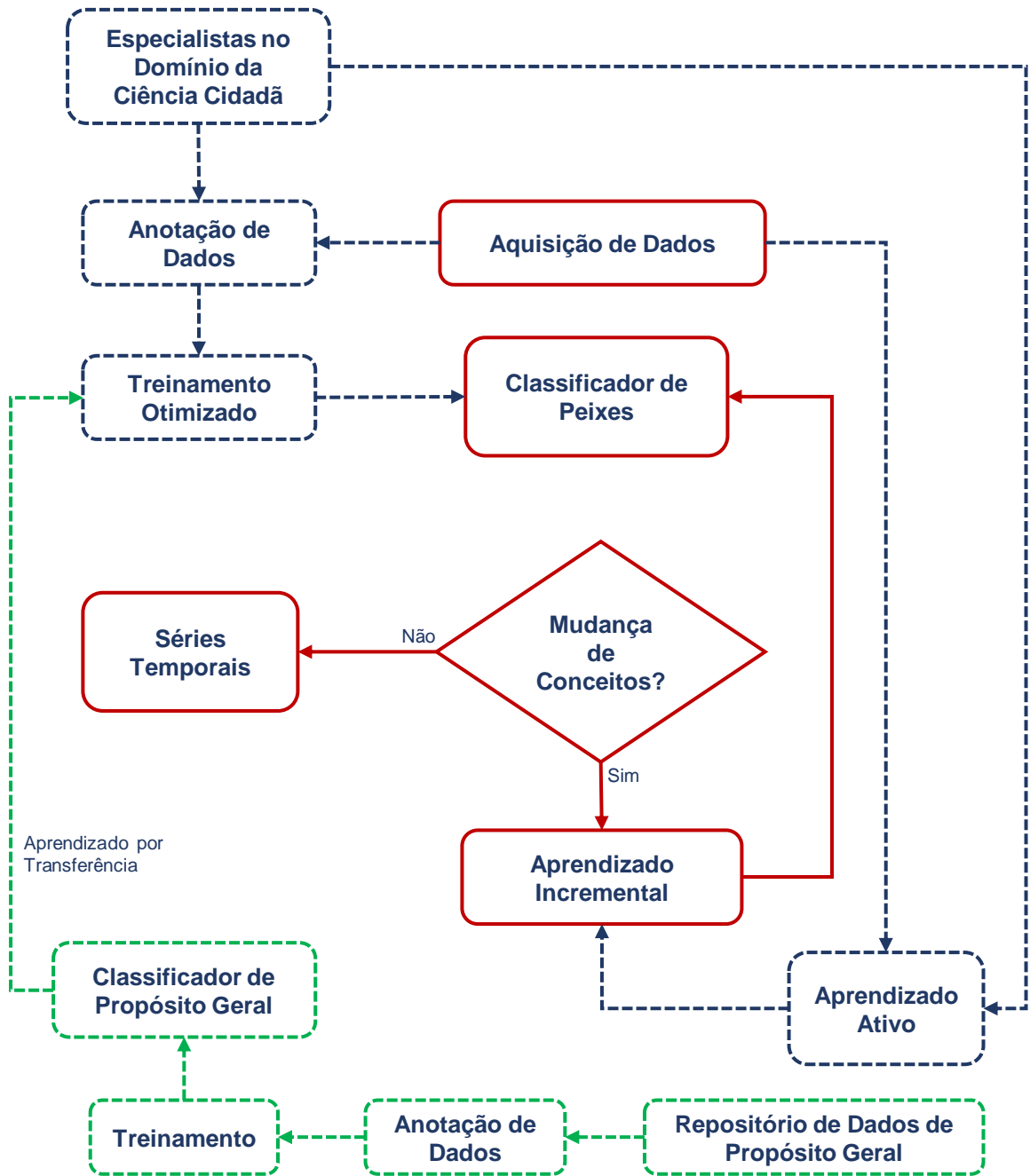
Fonte: Traduzido e Adaptado de (TEKIN; SCHAAAR, 2015).

Assim como o sistema IMS, outros estudos que consideram o APINC têm dividido os conjuntos de imagens em lotes de classes para avaliar o modelo de classificação. Cada lote se torna disponível, sequencialmente, para o treinamento do classificador. Entre cada estágio do treinamento, um teste avalia o modelo com as classes disponíveis no momento. O resultado do classificador corresponde à média de desempenhos computados para cada lote (REBUFFI et al., 2017; CASTRO et al., 2018).

Uma metodologia eficaz que considera aspectos do domínio da aplicação é o *pipeline* apresentado na Figura 7. Esse *pipeline* é aplicado à classificação automática de peixes

(OTTAVIANI et al., 2022) e tem como objetivo avaliar a degradação do desempenho de detecção e classificação de imagens no monitoramento de ecossistemas marinhos. Na figura, as linhas e caixas vermelhas contínuas descrevem o serviço automatizado; as linhas e caixas azuis pontilhadas representam atividades de anotação e treinamento; as linhas e caixas verdes tracejadas representam as atividades de aprendizagem por transferência.

Figura 7: Pipeline para classificação automática de imagens.



Fonte: Traduzido e Adaptado de (OTTAVIANI et al., 2022).

Nessa metodologia, o classificador é obtido de um *fine tuning* do treinamento. A ideia é especializar um classificador previamente treinado em um conjunto de dados de imagem genérico por meio de aprendizado de transferência (*transfer learning*). Para a classificação de peixes, o treinamento utiliza anotações de biólogos especialistas. O APINC é ativado quando o nível médio de confiança de classificação dos organismos contidos nas imagens decai abaixo de um determinado limite. O número de organismos não classificados pode ser estimado usando o nível de confiança para cada classe já fornecida pelo classificador.

A avaliação de modelos evolutivos é um desafio para o Aprendizado em Fluxos de Dados. Nos últimos anos, a **Avaliação Prequencial** com mecanismos de esquecimento tem sido defendida como método para fornecer estimativas de erro mais confiáveis em fluxos de dados (GAMA; SEBASTIAO; RODRIGUES, 2013). A Equação 4.1 apresenta o **Erro Prequencial** E computado no tempo t sobre uma janela deslizante w . O erro e é baseado em uma soma acumulada de uma função de perda L entre a previsão \hat{y} e os valores observados y , apresentada na Equação 4.2.

$$E_w(t) = \frac{1}{w} \sum_{k=1-w+1}^t L(y_k, \hat{y}_k) \quad (4.1)$$

$$L_w^{(t)}(y_k, \hat{y}_k) = \frac{1}{w} \sum_{k=1-w+1}^t e_k \quad (4.2)$$

Algumas propostas tentam adaptar o Erro Prequencial para avaliar dados desbalanceados (BRZEZINSKI; STEFANOWSKI, 2015) e mudança de conceitos (HIDALGO; MACIEL; BARROS, 2019). Mesmo assim, esse tipo de avaliação pode não corresponder às configurações de aprendizagem relacionadas aos aspectos do domínio da aplicação, como o atraso na chegada dos rótulos para predição (GRZENDA; GOMES; BIFET, 2020). Dessa forma, é importante considerar as características do fluxo de dados e o domínio da aplicação ao selecionar métricas de avaliação de desempenho.

No APINC, métricas relacionadas à acurácia do modelo têm se destacado em razão da simplicidade e rapidez de cálculo para avaliação de modelos de classificação de imagens. A Equação 4.3 corresponde à **Acurácia Média** (*average accuracy*), que fornece comparações para todo o processo incremental. Considerando a acurácia como o número de predições corretas, a acurácia $a_{t,k} \in [0, 1]$ de A_t é o resultado da tarefa k depois da tarefa aprendida t , com $k \leq t$, que fornece informações sobre o processo incremental (MASANA et al., 2022).

$$AM_t = \frac{1}{t} \sum_{i=1}^t a_{t,i} \quad (4.3)$$

A **Acurácia Geral** (*overall accuracy*) é uma versão da acurácia que tem sido utilizada de maneira complementar à Acurácia Média e a outras métricas comuns em tarefas de classificação (LI; DING; PIŽURICA, 2019; ROADY et al., 2020). A Acurácia Geral pode ser computada como na Equação 4.4, onde T é o número total de tarefas, α_t é a acurácia do modelo no tempo t e $\alpha_{o,t}$ é a acurácia do modelo *offline* otimizado no tempo t . Essa métrica permite avaliar o desempenho do modelo no fluxo passante e a capacidade de detectar novas entradas.

$$AG = \min \left(1, \frac{1}{T} \sum_{t=1}^T \frac{\alpha_t}{\alpha_{o,t}} \right) \quad (4.4)$$

Discussão: uma abordagem experimental moderna e eficiente deve considerar mecanismos *online* e *offline* para a aprendizagem e classificação das imagens. O Aprendizado Incremental é um fator importante nesse processo. Adaptações da acurácia para configurações incrementais são capazes de avaliar com eficácia modelos de classificação de imagens em fluxos de dados.

4.3 Aprendizado Profundo em Fluxos de Dados

Esta seção responde à questão de pesquisa **Q4: É possível utilizar modelos de Aprendizado Profundo para classificar imagens em fluxos de dados?**. De maneira antecipada, a resposta é positiva a essa questão. Com a ascensão dos modelos de Aprendizado Profundo nos últimos anos, a maior parte trabalhos em Visão Computacional passaram a ser conduzidos com aquela abordagem. De igual modo, a classificação de imagens em fluxos de dados acompanhou a tendência de trabalhar com CNNs. Dessa forma, apresenta-se a seguir um histórico de trabalhos relacionados ao Aprendizado Profundo em fluxos de dados, enfatizando aplicações em Visão Computacional.

Métodos de Aprendizado Profundo costumam trabalhar em modo *offline*, onde cada minilote é amostrado de um conjunto de dados estático e treinado ao longo de várias épocas. Para detectar as mudanças na distribuição de dados, esse tipo de treinamento requer um novo treinamento completo da rede, que pode ser ineficiente e tornar inviável a classificação quando os dados anteriores não estão disponíveis.

Em geral, o Aprendizado Profundo assume que a chegada de novos dados constitui uma tarefa individual, sendo cada tarefa estacionária. Assim, um modelo pode ser treinado de maneira *offline*, constituindo várias épocas sobre a tarefa atual com repetição aleatória. No entanto, essa configuração requer o armazenamento de todos os dados da tarefa de treinamento em execução, o que pode não ser viável devido às limitações de recursos computacionais (MAI et al., 2022).

O uso de algoritmos baseados em RNAs não é recente no Aprendizado em Fluxos de Dados. No entanto, o uso contínuo desses modelos revelou novos desafios ao treinamento, como a necessidade de atualização do modelo com novos dados e a manutenção do desempenho preditivo em tempo real. Embora essa estratégia possa ser simples em outros métodos, a atualização incremental de uma RNA em um fluxo de dados não estacionário resulta no fenômeno do **esquecimento catastrófico**.

O método ***Adaptive Deep Belief Network (ADBN)*** (CALANDRA et al., 2012) é uma proposta inicial para empregar o Aprendizado Profundo em FDs por meio do Aprendizado Online. A ideia é explorar as propriedades generativas do modelo para retreinar incrementalmente uma Rede de Crença Profunda (DBF) sempre que novos dados forem coletados. Como essa estratégia não armazena dados antigos, o consumo de memória é constante. Enquanto outras estratégias utilizam mecanismos explícitos para esquecer o passado, o método ADBN descarta conceitos antigos ao mesmo passo que incorpora novos dados.

O método ***Hedge Backpropagation (HBP)*** (SAHOO et al., 2018) é uma adaptação do popular algoritmo *Backpropagation* usando o algoritmo *Hedge* (FREUND; SCHAPIRE, 1997) para atualizar de maneira *online* os parâmetros de uma RNA densa com muitas camadas. A estratégia é fazer previsões com várias saídas de diferentes camadas ocultas da rede a fim de permitir o compartilhamento de conhecimento entre as redes mais profundas e mais rasas. Experimentos foram conduzidos para avaliar a capacidade da rede com diferentes profundidades. A abordagem identificou automaticamente como e quando modificar a capacidade de rede efetiva de maneira orientada a dados, com base na complexidade de dados observada.

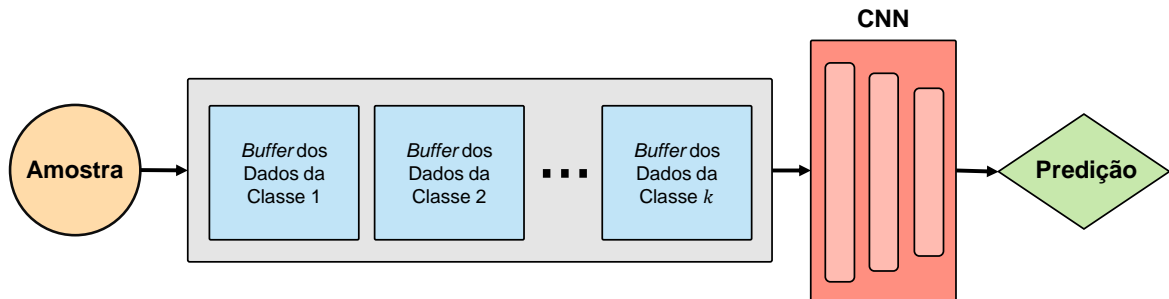
Os métodos ADBN e HBP adaptam algoritmos existentes para trabalhar de maneira incremental e *online*. No entanto, nenhuma das abordagens lida diretamente com o problema do esquecimento catastrófico. Ambos os métodos utilizam mecanismos para esquecer os dados passados e aprender novos conceitos. Embora essa estratégia seja eficiente na utilização de recursos computacionais, ela pode não ser eficaz em lidar com todas

as características intrínsecas dos fluxos de dados ou com características específicas, como o desvio de conceitos.

Uma estratégia que tem se mostrado eficaz é associar um valor de probabilidade para cada elemento do fluxo de dados (DUDA et al., 2020). A soma cumulativa C das probabilidades associadas permite monitorar as mudanças nos valores da função de perda e detectar o desvio quando a soma ultrapassa um valor limiar λ_C . O sobreajuste (*overfitting*) do modelo também pode ser detectado com a inclusão de um conjunto de dados de avaliação. Além disso, a substituição do hiperparâmetro de épocas pela construção incremental aleatória de minilotes permitiu que o treinamento da rede fosse mais eficiente.

No *framework* **Exemplar Streaming (ExStream)** (HAYES; CAHILL; KANAN, 2019), a estratégia é misturar dados já observados com dados recentes durante o treinamento para evitar o esquecimento catastrófico. A Figura 8 ilustra o funcionamento do algoritmo, que se baseia no agrupamento particional para criar conjuntos de *buffers* de dados por classe. Cada *buffer* também armazena estatísticas sobre os dados. Em seguida, esses dados são usados para treinar uma CNN e realizar previsões.

Figura 8: Criação de *buffers* no Algoritmo ExStream.



Fonte: Traduzido e Adaptado de (HAYES; CAHILL; KANAN, 2019).

Se um *buffer* específico fica cheio assim que um novo item de dado (x_t, y_t) chega ao fluxo, os dois grupos mais próximos no *buffer* para a classe y_t são buscados e mesclados usando a distância euclidiana $d_w(y_t)$. A Equação 4.5 define o cálculo da distância, onde w_i e w_j são os grupos mais próximos, e c_i e c_j são as estatísticas associadas aos grupos.

$$d_{w(y_t)_i} = \frac{c_i w_i + c_j w_j}{c_i + c_j} \quad (4.5)$$

O *framework* **Move-to-Data (MTD)** (POURSANIDIS et al., 2020; GEBRESLASSIE; BENOIS-PINEAU; ZEMMARI, 2022) é uma versão mais rápida do algoritmo ExStream. As duas abordagens atualizam os pesos da camada densa de classificação usando

Aprendizado em Fluxos de Dados enquanto congela a extração de atributos da parte convolucional. No entanto, enquanto ExStream atualiza os pesos de todas as camadas densas, MTD atualiza os pesos movendo, para um único neurônio, um vetor de pesos em direção ao vetor de dados.

O algoritmo *Deep Streaming Linear Discriminant Analysis (DSLDA)* (HAYES; KANAN, 2020) adapta o método ILDA para treinar o decodificador linear da Equação 4.6, sendo $F(\cdot)$ as J camadas iniciais da rede convolucional e $G(\cdot)$ as camadas densas. W é a matriz de pesos e z é o vetor de vieses que serão atualizados pela rede.

$$F(G(X_t)) = Wz_t + b \quad (4.6)$$

No DSLDA, a rede é treinada no primeiro lote de classes e congelada posteriormente. Durante o treinamento, um vetor de médias de cada classe e uma matriz compartilhada de covariâncias são atualizados. A previsão é feita atribuindo o rótulo ao gaussiano mais próximo no espaço de atributos definido pelos vetores médios da classe e pela matriz de covariância. O método evita o esquecimento catastrófico porque as médias de cada classe são independentes. Embora a matriz de covariância possa mudar com o tempo e seja sensível à ordenação de classes, as alterações na matriz resultam em, no máximo, um esquecimento gradual.

O algoritmo *REplay using Memory INDEXing (REMIND)* (HAYES et al., 2020) baseia-se numa representação inicial que é apenas parcialmente atualizada posteriormente. A abordagem utiliza uma técnica de quantização vetorial para armazenar representações intermediárias comprimidas de imagens, que são mais compactas que imagens. Os vetores armazenados são reconstruídos e reproduzidos para consolidação da memória. A abordagem REMIND é projetada para lidar com dados bidimensionais e de alta dimensão dentro da estrutura de classificação de imagens.

Considerando que o foco deste trabalho é a classificação de imagens, a prática comum da área é adotar CNNs como método de aprendizado. Por outro lado, diferentes tipos de RNAs podem contribuir para a execução da tarefa. Algumas abordagens incluem ainda a utilização de Redes Neurais Autocodificadoras (ASHFAHANI, 2019; DUDA; WANG, 2020), sobretudo para a detecção de mudança de conceitos na distribuição do fluxo de dados (JAWORSKI; RUTKOWSKI; ANGELOV, 2020).

Discussão: com o avanço das Redes Neurais Profundas, a tarefa de classificação de imagens passou a ter resolução trivial. O desafio principal é a classificação de imagens de maneira

incremental e *online*. Para que isso possa ser resolvido, é necessário que o mecanismo de aprendizagem da rede retenha apenas as imagens que formulem novos conceitos ou modifiquem conceitos existentes. No entanto, para evitar o esquecimento catastrófico, o processo deve ignorar as imagens excedentes antes que elas sejam apresentadas à rede. A extração prévia de atributos das imagens ou a combinação de métodos lineares com modelos profundos devem permitir a atualização incremental do modelo de decisão sem que o desempenho seja totalmente deteriorado.

5 *FRAMEWORK* PARA A CLASSIFICAÇÃO DE IMAGENS EM FLUXOS DE DADOS

Este capítulo apresenta o *framework* proposto nesta tese para a classificação de imagens em fluxos de dados. O *framework* consiste em um método computacional organizado em etapas para a utilização de modelos de Aprendizado Profundo no processo de classificação de imagens. As seções seguintes descrevem o desenvolvimento do *framework* e as etapas presentes no método.

5.1 Delineamento

Inicialmente, este trabalho observou o estado-da-arte e os desafios da aplicação da classificação de imagens em fluxos de dados. Com a evolução do Aprendizado Profundo nos últimos anos, CNNs foram consideradas como o principal solução aos desafios encontrados. No entanto, novos desafios relacionados ao uso dessa classe de modelos em fluxos de dados também foram encontrados. Dessa forma, um *framework* foi elaborado para sistematizar os elementos que compõem uma solução a partir dos desafios encontrados.

O *framework* proposto consiste em um método computacional organizado em etapas, que se baseiam no processo de classificação de imagens. Os seguintes aspectos iniciais foram considerados para o desenvolvimento das etapas:

1. **Processamento de imagens:** um conjunto potencialmente infinito de imagens devem ser processadas e, possivelmente, transformadas para serem utilizadas em um processo *online* de mineração de imagens;
2. **Extração de atributos:** as imagens que chegarem à etapa de treinamento deverão, antecipadamente, ter seus atributos mais relevantes selecionados e incluídos em um conjunto de treinamento;
3. **Aprendizagem:** o mecanismo de aprendizagem deve treinar o conjunto de dados observando as características intrínsecas dos fluxos de dados; e

4. **Classificação:** a classificação de uma nova imagem, pertencente ao fluxo contínuo de imagens, é rotulada.

Essas etapas formam a estrutura básica de funcionamento do *framework*. As seções seguintes detalham a construção do método e o mecanismo de funcionamento de cada etapa. Experimentos envolvendo a proposta são detalhados no Capítulo 6.

5.2 Arquitetura do *Framework* Proposto

Uma representação gráfica do *framework* é apresentada na Figura 9, que ilustra a arquitetura de implementação e o fluxo de funcionamento do método. As etapas são representadas nas caixas com bordas pontilhadas. Cada objeto interno às caixas representa uma tarefa. As linhas contínuas com setas indicam a transição entre as tarefas. As linhas tracejadas indicam a tarefa subjacente realizada pelos objetos de origem e/ou destino.

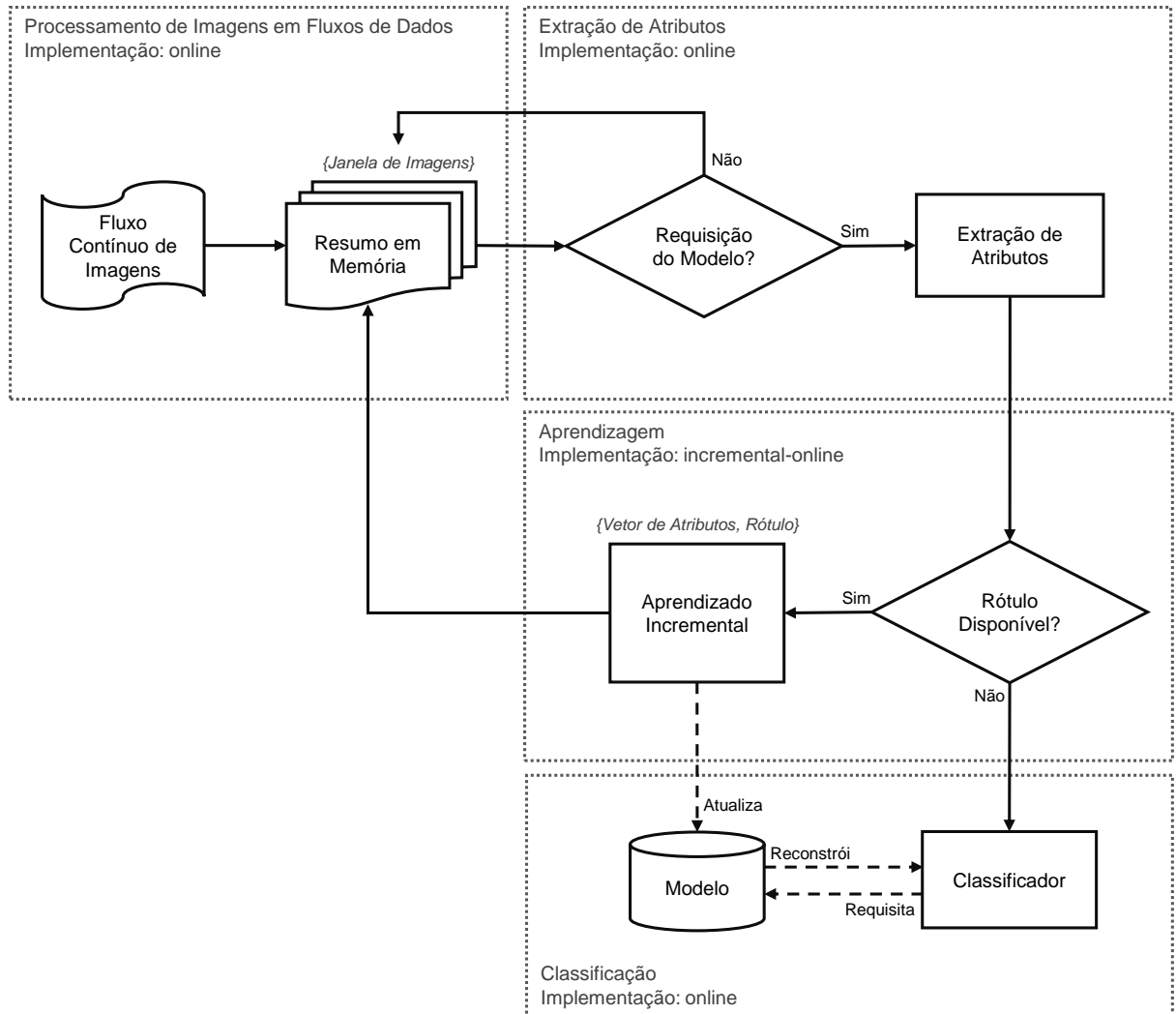
A primeira etapa do método é o **Processamento de Imagens em Fluxos de Dados**, que determina como o sistema irá coletar as imagens continuamente. O processo de coleta é definido pela estratégia de implementação do fluxo de dados adotado para a captura de imagens continuamente. Um simulador é utilizado para representar o fluxo contínuo de imagens.

A segunda etapa corresponde à **Extração de Atributos**, que visa à seleção das características mais relevantes das imagens passantes no fluxo de dados para a etapa de aprendizagem. Para que um conjunto de imagens passe por essa etapa, o modelo deverá requisitar acesso aos dados de entrada. A saída da etapa é um vetor de atributos extraído das imagens de entrada. Em alguns casos, a saída também poderá ser o conjunto de rótulos para treinamento.

Se os rótulos estiverem disponíveis, o método seguirá para a etapa de **Aprendizagem**, responsável por treinar o modelo neural. Após o treinamento, o modelo será atualizado e o método deverá selecionar novas imagens do fluxo de entrada. A requisição do modelo na etapa anterior também poderá acionar a construção do classificador.

Na etapa de **Classificação**, um classificador existente poderá requisitar ao modelo construído incrementalmente e armazenado de maneira *online*. Essa requisição não atualizará o modelo, mas irá recuperar a versão mais recente do modelo construído pelo modelo neural.

Figura 9: *Framework* proposto – arquitetura e fluxo de funcionamento.



Resumidamente, o *framework* proposto soluciona os principais desafios relacionados às características intrínsecas dos fluxos de dados para classificação de imagens. A solução emprega classes de modelos neurais em uma implementação incremental e *online*.

5.3 Processamento de Imagens como Fluxo de Dados

Nesta etapa, as imagens são recebidas pelo método para serem processadas como um fluxo contínuo de imagens. As imagens são tratadas como itens de dados de entrada no fluxo e, em seguida, são processadas para serem utilizadas no processo de aprendizagem. Embora o *framework* não descreva como é realizada a captura de imagens, um simulador de fluxos de dados pode ser implementado para inserir as imagens no processo. Esta tese

adota a estratégia do simulador para processar as imagens de entrada. Qualquer modelo de fluxo de dados pode ser implementado para a realização dessa tarefa.

Para resumir as imagens em memória, implementou-se a técnica de Janelamento Amortecido (YUN et al., 2018). No *framework*, essa técnica foi modificada para manter a distribuição de probabilidades dos itens de dado que chegam ao fluxo, mas sem computar os *timesteps*. Considerando que o simulador pode não fornecer o intervalo de tempo ao processo, a ideia principal da janela é observar a maior probabilidade dos itens no fluxo e resumir em memória um conjunto limitado de imagens. A adoção dessa estratégia também visou o não descarte dos conceitos processados, evitando que o esquecimento catastrófico aconteça na etapa de aprendizagem.

5.4 Mecanismo de Seleção de Atributos

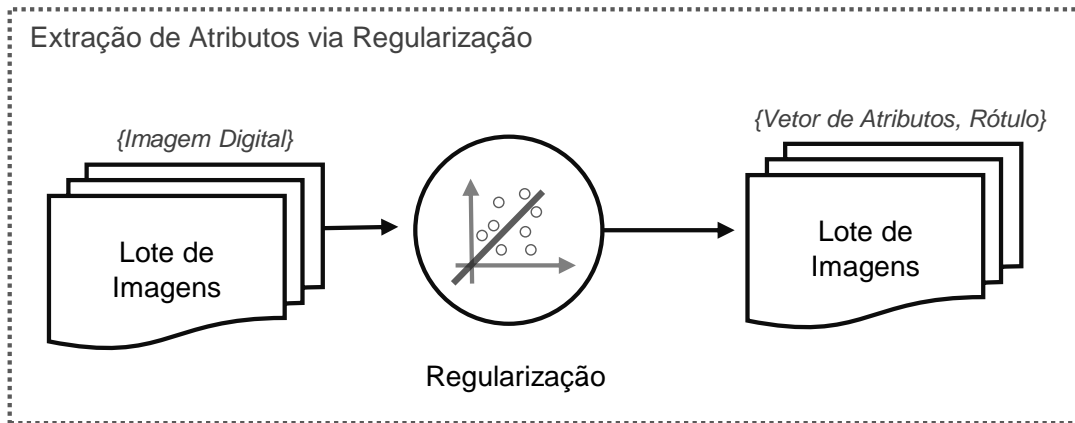
Para a etapa de extração de atributos, diferentes estratégias foram investigadas para a seleção das características mais significativas das imagens em um fluxo de dados. A Seção 4.1 descreve os resultados dessa investigação, que conclui sobre a combinação de métodos lineares com Aprendizado Profundo para a eficiência de uma solução computacional menos custosa.

Neste trabalho, optou-se por adotar um mecanismo de seleção de atributos semelhante aos modelos *Deep Feature Selection* (LI; CHEN; WASSERMAN, 2016) e *Online Feature Selection* (WANG et al., 2014). Um método de Regressão Linear com regularização ElasticNet foi implementado junto à Rede Neural Convolutiva, na camada de entrada, para reduzir em tempo linear o espaço de atributos. A implementação é mais simples do que os modelos estudados e demonstrou resultados equivalentes.

A Figura 10 ilustra o funcionamento do mecanismo de seleção de atributos implementado para o *framework* proposto. Um lote de imagens digitais é recebido e, imediatamente, transformado em um vetor de atributos por meio da aplicação de um método de Regressão Linear regularizado. A regularização adotada pelo *framework* é o método Elastic Net, mantendo o parâmetro de penalização $0 < \alpha < 1$.

Uma vez que as imagens são recebidas em lote, o mecanismo não trabalha de maneira incremental, lidando imagem por imagem. No entanto, a implementação *online* permite que o modelo de decisão da regressão seja construído e atualizado rapidamente em cada nova requisição do modelo de classificação.

Figura 10: *Framework* proposto – detalhamento da fase de extração de atributos.



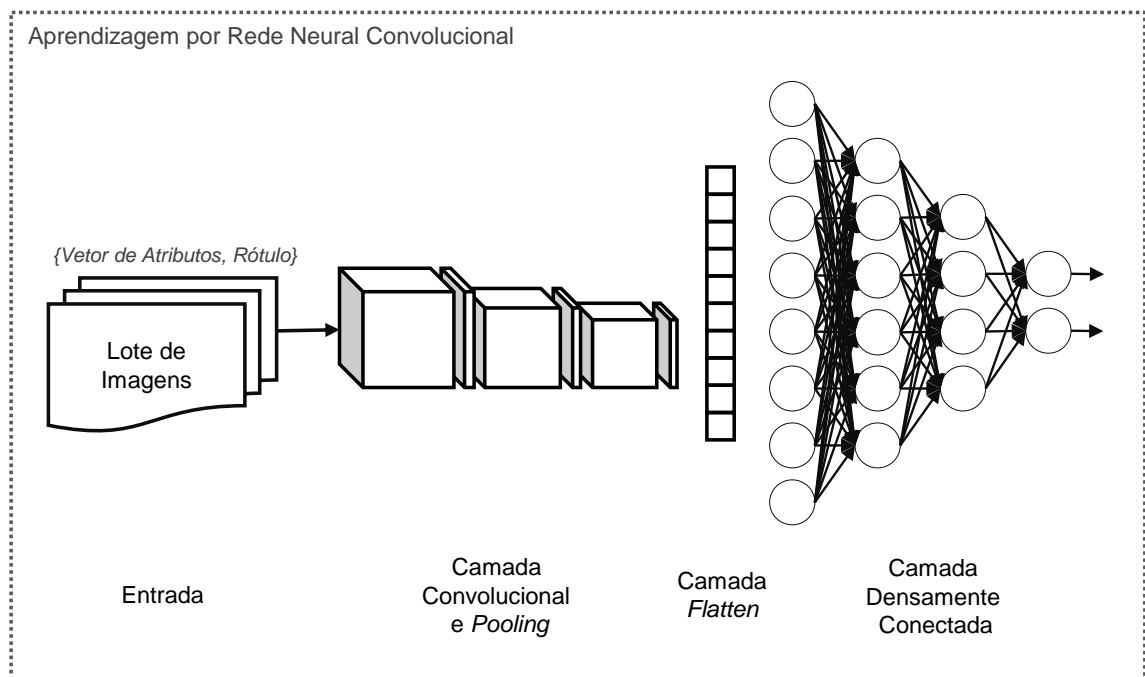
5.5 Classificador de Imagens em Fluxos de Dados

A Figura 11 detalha o funcionamento da fase de aprendizagem. Um lote de imagens no formato de vetor de atributos mais o rótulo são recebidos como entrada pela CNN. Em seguida, blocos de camadas convolucional e *pooling* contribuem para a aprendizagem das características das imagens. Como a Regressão Linear da etapa anterior é implementada junto com a CNN, o método Elastic Net não modifica as características visuais das imagens observadas, mas apenas reduz o espaço de atributos. Por fim, a camada *flatten* transforma a saída do último bloco em um vetor unidimensional, que é a entrada da do conjunto de camadas densamente conectadas. A saída é a resposta da rede para a predição da classe da imagem.

O processo de aprendizagem é semelhante ao algoritmo *Deep Streaming Linear Discriminant Analysis* (HAYES; KANAN, 2020). Para que os dados no bloco de camadas convolucional sejam processados incrementalmente, o rótulo deve estar disponível. Além disso, a atualização incremental do modelo acontece apenas quando novos conceitos são observados pela rede neural. Dessa forma, conceitos antigos são mantidos no modelo sem que ele precise ser atualizado. Essa estratégia contribui para que evitar o esquecimento catastrófico e deteriorar o desempenho do modelo de decisão.

Caso o rótulo não esteja disponível para a aprendizagem, o *framework* direciona o vetor de atributos para o classificador prever os rótulos das imagens do lote. Esse processamento acontece na etapa de classificação. Durante essa etapa, o modelo é atualizado pelo processo de aprendizagem, que reconstrói o classificador.

Figura 11: *Framework* proposto – detalhamento da fase de aprendizagem.



6 EXPERIMENTOS

Este capítulo descreve o planejamento experimental conduzido nesta tese para avaliar o desempenho dos modelos construídos para classificação de imagens. As seções seguintes descrevem a abordagem experimental empregada no trabalho, as bases de dados adotadas para a experimentação e o processo de preparação de dados. Também são descritas a infraestrutura utilizada para os experimentos e a configuração de hiperparâmetros dos algoritmos comparados, além das métricas de avaliação de desempenho.

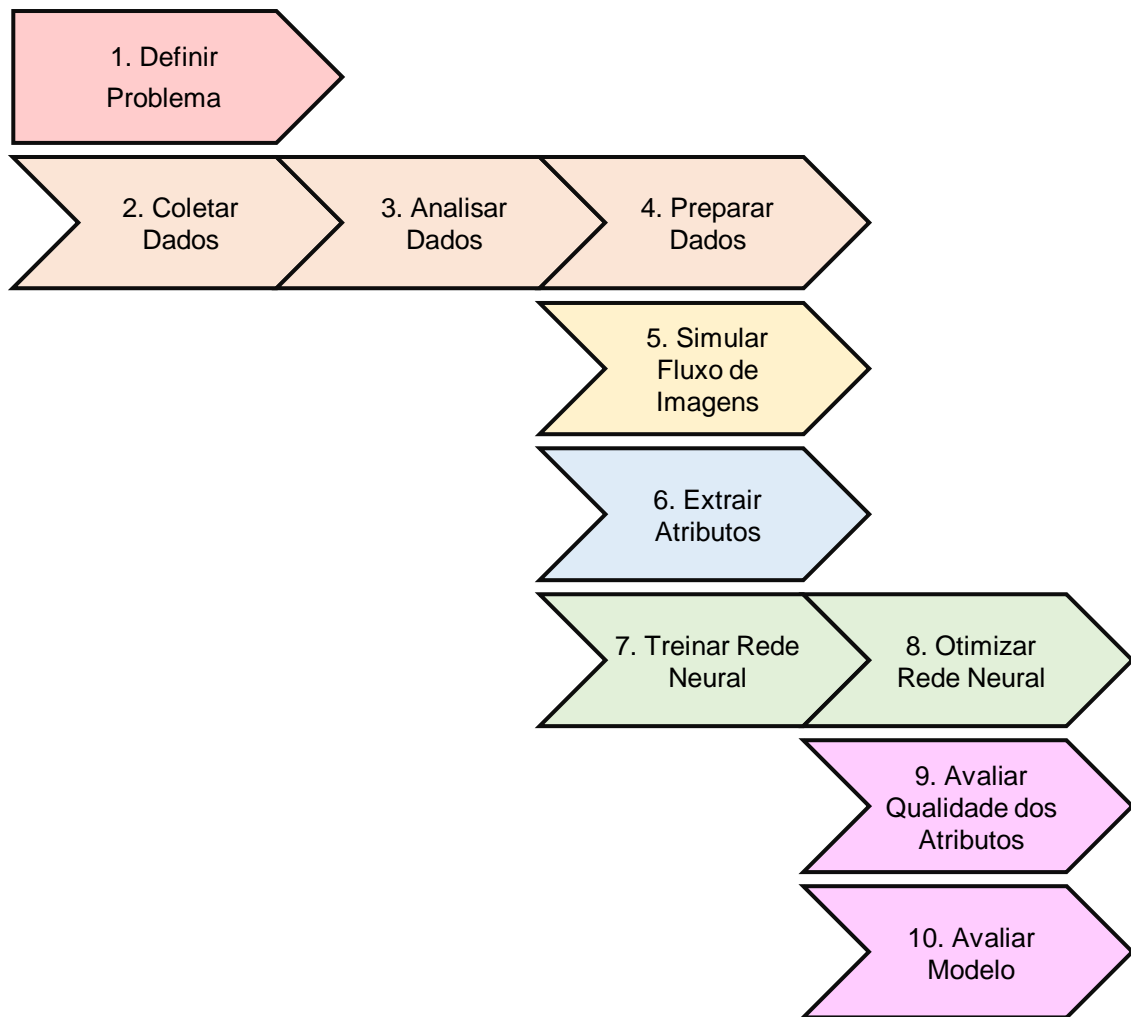
6.1 Abordagem Experimental

Para avaliar a qualidade dos modelos de classificação de imagens, esta tese enfatizou a validação do *framework* proposto no Capítulo 5. Dessa forma, para verificar se a proposta é superior aos métodos encontrados na literatura, planejou-se uma abordagem experimental que permitisse a comparação dos resultados entre os métodos.

A abordagem experimental é ilustrada na Figura 12, que mostra o processo de avaliação em dez etapas, sendo detalhadas a seguir:

1. **Definir problema:** o problema da classificação de imagens é definido em termos de predição para um fluxo contínuo de dados, tendo como objeto a imagem digital. A definição do problema é descrita na Seção 1.2;
2. **Coletar dados:** os dados utilizados para a experimentação e comparação junto ao *framework* proposto são detalhados na Seção 6.2. Optou-se por escolher bases de dados reais, em vez de sintéticas, a fim de evitar vieses na comparação com outros métodos;
3. **Analisar dados:** a análise dos dados das imagens consistiu na exploração visual das características das imagens. Também são observados possíveis ruídos e inconsistências que não permitissem a validação completa do *framework*;

Figura 12: Abordagem experimental adotada para a validação do *framework* proposto.



4. **Preparar dados:** esta etapa organizou os dados de modo a separá-los em conjuntos disjuntos de treinamento e teste das bases de dados de imagens. Os conjuntos de treinamento foram destinados ao treinamento da rede neural. Os conjunto de teste avaliaram a qualidade do modelo neural construído. Mais detalhes sobre a análise e preparação dos dados de imagens são apresentados na Seção 6.3.
5. **Simular fluxo de imagens:** para permitir que os dados fossem processados pela rede neural de maneira incremental e *online*, foi desenvolvido um mecanismo de ingestão de dados baseado na técnica de Janelamento para consolidar o lote de imagens em memória. Detalhes do funcionamento do fluxo estão descritos na Seção 5.3;
6. **Extrair atributos:** o mecanismo de extração e seleção de atributos relevantes das imagens de entrada no fluxo estão descritos em detalhes na Seção 5.4;

7. **Treinar rede neural:** as características extraídas na etapa anterior foram reunidas em um conjunto de dados juntamente com o rótulo original da imagem para formar o vetor de atributos utilizados no treinamento da Rede Neural Convolutacional;
8. **Otimizar rede neural:** após o treinamento com os hiperparâmetros *default* da rede neural, um novo experimento foi conduzido utilizando a busca em grade (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019) para encontrar os melhores hiperparâmetros, que são descritos na Seção 6.6;
9. **Avaliar qualidade dos atributos:** a qualidade dos atributos foi avaliada de acordo a capacidade das características extraídas das imagens em classificá-las corretamente, conforme as métricas de avaliação de desempenho preditivo adotadas;
10. **Avaliar modelo:** a avaliação da qualidade preditiva do modelo neural de classificação de imagens foi realizada a partir de métricas comuns da área.

Embora a abordagem detalhe sequencialmente as etapas, elas não necessariamente foram aplicadas de maneira seriada. Por exemplo, a simulação do fluxo de imagens não é indispensável para a obtenção dos resultados preditivos, sobretudo se interligada com a extração de atributos. Dessa forma, para computar os resultados e torná-los mais facilmente publicáveis neste trabalho, diferentes experimentos foram realizados para validar o *framework*, sem a necessidade da construção de um sistema computacional de ponta a ponta.

6.2 Bases de Dados

O bloco de etapas subsequentes à definição do problema é relacionado à utilização das imagens digitais. Nesta tese, três bases de dados de imagens de foram escolhidas para avaliar o *framework* proposto. As imagens presentes nessas bases de dados são comuns na área de Visão Computacional para a tarefa de classificação de imagens. As características das bases de dados avaliadas são descritas na Tabela 1.

A base de dados **Food-5k**¹ (SINGLA; YUAN; EBRAHIMI, 2016) contém 2.500 imagens de alimentos e 2.500 imagens de não alimentos. A Figura 13 ilustra uma amostra retirada de ambas as categorias. A base de dados original está dividida em três conjuntos de imagens: treinamento, avaliação e teste. As imagens possuem dimensões e tamanhos

¹<https://www.epfl.ch/labs/mmspg/downloads/food-image-datasets/>

Tabela 1: Bases de dados adotadas para a avaliação experimental do *framework* proposto.

Nome	Coloração	Classes	Imagens	Treinamento	Teste
Food-5k	Colorida RGB	2	5.000	4.000	1.000
CIFAR-10	Colorida RGB	10	60.000	50.000	10.000
MNIST	Nível de Cinza	10	70.000	60.000	10.000

variados. Para permitir que todas as bases de dados sejam avaliadas igualmente, o conjunto de avaliação foi mesclado com o conjunto de treinamento.

Figura 13: Amostra de 25 imagens de alimentos e 25 imagens de não alimentos aleatoriamente selecionadas da base de dados Food-5k.



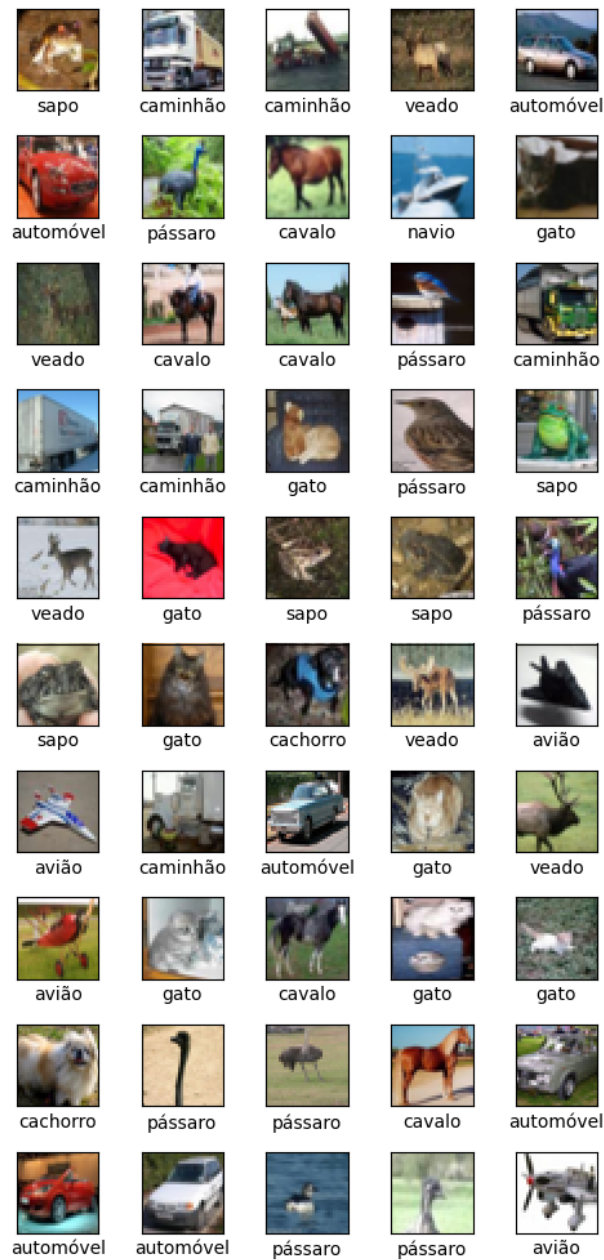
A base de dados **CIFAR-10**² (KRIZHEVSKY; HINTON et al., 2009) é uma coleção de imagens coloridas comumente usadas em Aprendizado de Máquina para treinar modelos diversos a classificar imagens de categorias diferentes. O conjunto de dados possui 60 mil imagens categorizadas em 10 classes diferentes: avião, automóvel, pássaro, gato, veado, cachorro, sapo, cavalo, navio e caminhão. Cada classe possui exatamente 6 mil imagens no formato de 32x32 pixels. A Figura 14 ilustra um subconjunto das imagens presentes na base de dados.

A base de dados **MNIST**³ (LECUN et al., 2010) é um conjunto de dados de imagens de dígitos manuscritos. Cada imagem é representada por uma matriz de 28x28 pixels,

²<https://www.cs.toronto.edu/~kriz/cifar.html>

³<https://yann.lecun.com/exdb/mnist/>

Figura 14: Amostra das 50 primeiras imagens contidas na base de dados CIFAR-10.



sendo cada pixel representado com um valor inteiro entre 0 e 255 na escala de cinza. A Figura 15 apresenta um subconjunto das imagens contidas na base de dados. O conjunto de dados possui 10 classes. A distribuição de classes é apresentada na Tabela 2 e mostra que aproximadamente 85% das imagens por classe são usadas para treinamento.

Figura 15: Amostra de imagens presentes na base de dados MNIST.

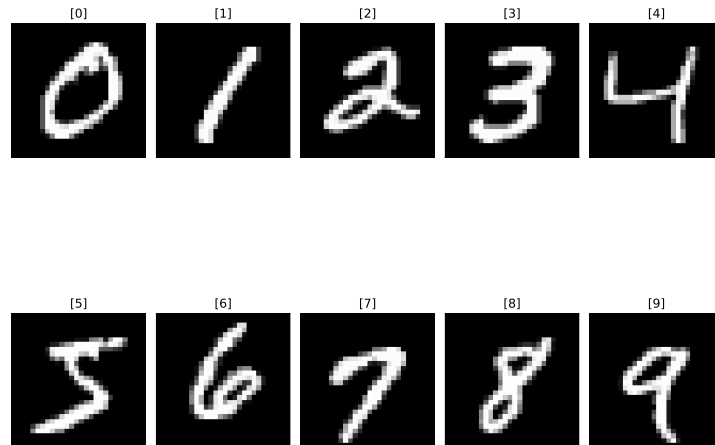


Tabela 2: Distribuição de classes da base de dados de imagens MNIST.

Classe	Imagens	Treinamento	Teste	% Treinamento
0	6903	5923	980	85,80%
1	7877	6742	1135	85,59%
2	6990	5958	1032	85,24%
3	7141	6131	1010	85,86%
4	6824	5842	982	85,61%
5	6313	5421	892	85,87%
6	6876	5918	958	86,07%
7	7293	6265	1028	85,90%
8	6825	5851	974	85,73%
9	6958	5949	1009	85,50%

6.3 Preparação dos Dados

O *framework* proposto observou um total de 135 mil imagens distintas. Para suportar esse volume de imagens de maneira eficiente e em diferentes e repetidos experimentos, os conjuntos de imagens foram pré-processados.

Para a base de dados Food-5k, os diretórios de imagens foram reorganizados para facilitar a leitura pelo fluxo de dados. Como os experimentos não esperam pela etapa de validação, as imagens originais de validação foram movidas para o diretório de treinamento. Além disso, como as imagens possuem diferentes dimensões, todas foram redimensiona-

das para uma única dimensão de 32x32 pixels. Nenhuma imagem tinha dimensão menor que essa. Dessa forma, um único vetor totalmente preenchido com 1024 características contribuiu para ser formatado como entrada da Rede Neural Convolutacional.

As bases de dados CIFAR-10 e MNIST não precisaram passar por nenhum pré-processamento sobre a imagem original. Da mesma forma que na base de dados Food-5k, as imagens em CIFAR-10 e MNIST foram organizadas em conjuntos de treinamento e teste para facilitar o processamento dos fluxos de dados na rede neural. Em todas as bases, os pixels das imagens foram normalizados na escala entre 0 e 1.

Durante a experimentação, diversas cópias das imagens foram criadas para representar os lotes processados pela simulação do fluxo de imagens. Arquivos de texto CSV também foram criados para armazenar em disco os vetores de atributos e o rótulo das imagens, quando disponível. Essa estratégia de implementação permitiu avaliar a aprendizagem independentemente da simulação, que imita um fluxo de dados real.

6.4 Infraestrutura

Os experimentos realizados neste trabalho foram executados em dois ambientes diferentes: localmente e no Google Colab⁴. O ambiente local consiste em um notebook pessoal da marca Acer. O ambiente Google Colab também foi utilizado por disponibilizar melhores recursos computacionais e uma plataforma em nuvem controlada de rápida codificação. A Tabela 3 apresenta a configuração de recursos computacionais dos ambientes utilizados nos experimentos.

Tabela 3: Configuração dos ambientes Google Colab e Local utilizados nos experimentos.

	Colab	Local
Processadores	2	1
CPU	Intel Xeon	Intel Core i5-10210U
Clock CPU	2.20GHz	1.60GHz
Espaço Total em Disco	108 GB	256 GB
Memória RAM	13 GB	20 GB
Sistema Operacional	Ubuntu 22.04	Windows 10

⁴<https://colab.research.google.com/>

Em ambos os ambientes, diversas bibliotecas construídas em Python foram instaladas e configuradas para permitir que o *framework* proposto fosse desenvolvido. Além disso, a maior parte dos métodos e algoritmos comparados têm suas implementações também nessa linguagem. A Tabela 4 apresenta as bibliotecas Python e suas versões utilizadas nos experimentos deste trabalho para os ambientes Google Colab e ambiente local.

Tabela 4: Versões das bibliotecas Python para os ambientes Google Colab e Local.

Ferramenta	Colab	Local
Python	3.10.12	3.9.4
SciPy	1.10.1	1.10.1
Numpy	1.23.5	1.24.3
Pandas	1.5.3	2.0.1
Matplotlib	3.7.1	3.7.1
scikit-learn	1.2.2	1.2.2
TensorFlow	2.12.0	2.11.0
Keras	2.12.0	2.11.0
Torch	2.0.1+cu118	2.0.1+cu118
Theano	-	1.0.5
OpenCV	4.8.0	4.7.0
scikit-multiflow	0.6.dev0	-
river	0.18.0	0.14.0

6.5 Configuração Experimental

A avaliação do *framework* foi realizada para dois grupos de experimentos. O primeiro grupo (*SA*) teve como objetivo a avaliar a qualidade da seleção de atributos para a classificação das imagens. A estratégia foi substituir o mecanismo de seleção de atributos do *framework* (*FW*), mencionado na Seção 5.4, por diferentes modelos de seleção de atributos relatados na Seção 4.1. A avaliação dos experimentos desse bloco reporta métricas de classificação de imagens, e não a importância dos atributos para o modelo.

A Tabela 5 apresenta o conjunto de blocos de experimentos realizados para o grupo *SA*. Cada bloco realiza 5 experimentos, considerando a quantidade Q_t^{SA} de atributos selecionados, onde $Q_t^{SA} = \{100, 200, 300, 400, 500\}$. Os tamanhos Q_t^{SA} foram definidos a

partir da dimensão máxima das imagens analisadas e dos modelos comparados. No total, 100 experimentos foram realizados para o grupo *SA*.

Tabela 5: Blocos de experimentos realizados para o grupo *SA*.

Grupo	Classificação	Classes	Base de Dados	Método
<i>SA</i> – 1	Binária	2	Food-5k	DFS+FW
<i>SA</i> – 2	Binária	2	Food-5k	OFS+FW
<i>SA</i> – 3	Binária	2	Food-5k	FSDS+FW
<i>SA</i> – 4	Binária	2	Food-5k	FW
<i>SA</i> – 5	Binária	2	CIFAR-10	DFS+FW
<i>SA</i> – 6	Binária	2	CIFAR-10	OFS+FW
<i>SA</i> – 7	Binária	2	CIFAR-10	FSDS+FW
<i>SA</i> – 8	Binária	2	CIFAR-10	FW
<i>SA</i> – 9	Binária	2	MNIST	DFS+FW
<i>SA</i> – 10	Binária	2	MNIST	OFS+FW
<i>SA</i> – 11	Binária	2	MNIST	FSDS+FW
<i>SA</i> – 12	Binária	2	MNIST	FW
<i>SA</i> – 13	Multiclasses	10	CIFAR-10	DFS+FW
<i>SA</i> – 14	Multiclasses	10	CIFAR-10	OFS+FW
<i>SA</i> – 15	Multiclasses	10	CIFAR-10	FSDS+FW
<i>SA</i> – 16	Multiclasses	10	CIFAR-10	FW
<i>SA</i> – 17	Multiclasses	10	MNIST	DFS+FW
<i>SA</i> – 18	Multiclasses	10	MNIST	OFS+FW
<i>SA</i> – 19	Multiclasses	10	MNIST	FSDS+FW
<i>SA</i> – 20	Multiclasses	10	MNIST	FW

O segundo grupo de experimentos (*AP*) teve como objetivo avaliar o modelo de decisão implementado pelo *framework* originalmente proposto. Nessa implementação, o mecanismo de seleção de atributos não é modificado. Dessa forma, os resultados preditivos do *framework* são comparados com os métodos citados na Seção 4.2 e Seção 4.3.

A Tabela 6 apresenta o conjunto de blocos de experimentos realizados para o grupo *AP*. Cada bloco realiza cinco experimentos, considerando a quantidade Q_t^{AP} de lotes de imagens de entrada para a Rede Neural Convolutiva, onde $Q_t^{AP} = \{100, 200, 300, 400, 500\}$.

Os tamanhos dos lotes Q_t^{AP} foram definidos a partir do mínimo múltiplo comum entre os volumes de imagens das bases de dados analisadas. No total, 100 experimentos foram realizados para o grupo AP .

Tabela 6: Blocos de experimentos realizados para o grupo SA .

Grupo	Classificação	Classes	Base de Dados	Método
$AP - 1$	Binária	2	Food-5k	ILDA
$AP - 2$	Binária	2	Food-5k	ExStream
$AP - 3$	Binária	2	Food-5k	DSLDA
$AP - 4$	Binária	2	Food-5k	FW
$AP - 5$	Binária	2	CIFAR-10	ILDA
$AP - 6$	Binária	2	CIFAR-10	ExStream
$AP - 7$	Binária	2	CIFAR-10	DSLDA
$AP - 8$	Binária	2	CIFAR-10	FW
$AP - 9$	Binária	2	MNIST	ILDA
$AP - 10$	Binária	2	MNIST	ExStream
$AP - 11$	Binária	2	MNIST	DSLDA
$AP - 12$	Binária	2	MNIST	FW
$AP - 13$	Multiclasses	10	CIFAR-10	ILDA
$AP - 14$	Multiclasses	10	CIFAR-10	ExStream
$AP - 15$	Multiclasses	10	CIFAR-10	DSLDA
$AP - 16$	Multiclasses	10	CIFAR-10	FW
$AP - 17$	Multiclasses	10	MNIST	ILDA
$AP - 18$	Multiclasses	10	MNIST	ExStream
$AP - 19$	Multiclasses	10	MNIST	DSLDA
$AP - 20$	Multiclasses	10	MNIST	FW

Os experimentos dos grupos SA e AP foram conduzidos para avaliar o *framework* proposto tanto para a classificação binária quanto para a classificação multiclasses. Como a base de dados Food-5k possui apenas duas classes, nenhum filtro foi executado sobre ela. No entanto, para as bases de dados CIFAR-10 e MNIST, foi necessário realizar o filtro sobre dois rótulos para os experimentos de classificação binária. Em CIFAR-10, as classes selecionadas foram $\{cachorro, gato\}$. Para MNIST, as classes foram $\{0, 1\}$.

6.6 Hiperparâmetros da Rede Neural

Considerando a alta quantidade de experimentos e os limitados recursos computacionais, optou-se por construir uma nova RNP, mais simples e direcionada aos desafios desta tese, em vez de utilizar modelos pré-treinados, como VGG (SIMONYAN; ZISSERMAN, 2014), ResNet (HE et al., 2016) e Inception (SZEGEDY et al., 2015). A adaptação desses modelos para a aprendizagem incremental e *online* demandaria um novo treinamento sobre as arquiteturas, tornando inviável a realização da tarefa com a infraestrutura disponível.

Outra desvantagem dos modelos pré-treinados é o viés das arquiteturas sobre a base de dados ImageNet (RUSSAKOVSKY et al., 2015). Esses modelos podem destacar atributos pouco relevantes para as bases de dados de imagens em estudo. Dessa forma, a construção de uma nova RNP com menos blocos de camadas permitiu que o treinamento dos modelos fosse realizado com os recursos disponíveis. Os atributos realmente relevantes também puderam ser extraídos com a nova arquitetura.

Os detalhes da arquitetura da RNP utilizada para o *framework* são detalhados a seguir:

- **Camada de Entrada**

- Número de camadas: 1
- Dimensão da entrada: (28, 28) ou (32, 32)
- Canais: 1 ou 3

- **Camada de Convolação**

- Número de camadas: 4
- Número de filtros: 32, 64 e 128
- Dimensão dos filtros: (3, 3)
- Função de ativação: ReLU
- Regularização: Dropout com taxa de 20%
- *Padding*: *same*
- *Strides*: (1, 1)

- **Camada de *Pooling***

- Número de camadas: 4, uma para cada camada convolucional
- Tipo: *Max Pooling*
- Dimensão dos filtros: (2, 2)
- **Camada *Flatten***
- **Camada Densamente Conectada**
 - Número de camadas: 4
 - Unidades nas camadas intermediárias: 1024, 512 e 128
 - Função de ativação das camadas intermediárias: ReLU
 - Unidades na última camada: 2 ou 10
 - Função de ativação da última camada: Sigmoid (binário) ou Softmax (multi-classes)

Para o treinamento da RNP, os seguintes hiperparâmetros foram configurados:

- **Função de perda:** Entropia cruzada binária ou categórica (multiclasses)
- **Otimização:** Adam
- **Tamanho do lote:** {100, 200, 300, 400, 500}
- **Épocas:** 50
- **Taxa de aprendizado:** 10^{-3}
- **Métrica de avaliação:** acurácia (ver Seção 6.7)

6.7 Avaliação de Desempenho

A avaliação do *framework* proposto consistiu na comparação entre os resultados obtidos por meio dos experimentos listados na Tabela 5 e Tabela 6. A comparação entre os experimentos computou as duas principais métricas encontradas na literatura para a classificação de imagens em configurações incrementais: Acurácia Média (Equação 4.3) e Acurácia Geral (Equação 4.4).

No entanto, em razão das configurações dos experimentos selecionados, as métricas precisaram ser adaptadas. A equação de cada métrica foi ajustada para computar mais

facilmente os acertos em cada classe. Para consolidar os resultados, as métricas passaram a computar o valor médio dos lotes de imagem treinados.

Por fim, cada métrica foi expandida para analisar unicamente a tarefa em análise: classificação binária ou classificação multiclases. Os problemas de classificação binária comparam o resultado entre as classes. Nos problemas multiclases, utilizou-se a estratégia um-contra-todos (JAMES et al., 2023), criando um classificador para cada classe. As novas métricas são apresentadas a seguir.

A **Acurácia Média Binária** AM^B do modelo \hat{f} para a tarefa de classificação binária B corresponde à taxa de acerto α nas duas classes, c_1 e c_2 , em cada lote L com K imagens, ponderada pela quantidade N de imagens em cada classe. Esse cálculo é expresso na Equação 6.1.

$$AM_B(\hat{f}) = \frac{1}{L \times K} \sum_{l=1}^L \left(\frac{1}{N_{c_1}} \sum_{k=1}^K \alpha_{c_1}^{(k)} + \frac{1}{N_{c_2}} \sum_{k=1}^K \alpha_{c_2}^{(k)} \right) \quad (6.1)$$

A **Acurácia Geral Binária** AG^B do modelo \hat{f} para a tarefa de classificação binária B corresponde à taxa de acerto nas classes c_1 e c_2 , sendo calculada pela média aritmética não ponderada de acertos de cada classe em cada lote L incrementalmente observado em N imagens. A métrica é calculada conforme a Equação 6.2.

$$AG_B(\hat{f}) = \frac{1}{L \times K} \sum_{l=1}^L \left(\sum_{k=1}^K (\alpha_{c_1}^{(k)} + \alpha_{c_2}^{(k)}) \right) \quad (6.2)$$

A **Acurácia Média Multiclases** AM^M do modelo \hat{f} para a tarefa de classificação multiclases M corresponde à taxa de acerto α computada nas classes C em cada lote L com K imagens, ponderada pela quantidade N_c de imagens em cada classe. A Equação 6.3 apresenta o cálculo dessa métrica.

$$AM_M(\hat{f}) = \frac{1}{L \times K} \sum_{l=1}^L \sum_{i=1}^K \left(\frac{1}{N_{c_j}} \sum_{j=1}^C \alpha_{c_j}^{(k)} \right) \quad (6.3)$$

A **Acurácia Geral Multiclases** AG^M do modelo \hat{f} para a tarefa de classificação multiclases M corresponde à taxa de acerto α computada pela média aritmética não ponderada de acertos de cada classe em cada em cada lote L incrementalmente observado em K imagens. Esse cálculo é expresso na Equação 6.4.

$$AG_M(\hat{f}) = \frac{1}{L \times K \times C} \sum_{l=1}^L \sum_{i=1}^K \sum_{j=1}^C \alpha_{c_j}^{(k)} \quad (6.4)$$

Os valores obtidos para cada métrica foram normalizados entre 0 e 1. Quanto mais próximo de 1, maior o poder preditivo do método. Valores próximos de 0 indicam baixo desempenho do modelo. Admite-se ainda que, para valores menores que 0.5, o modelo não classifica imagens em fluxos de dados. Ou seja, a hipótese desta tese é rejeitada. Os resultados são reportados no Capítulo 7 com três casas decimais.

Discussão: os experimentos relatados neste capítulo foram planejados a partir da abordagem experimental descrita na Seção 6.1. Bases de dados clássicas em Visão Computacional foram utilizadas para aumentar a reprodutibilidade dos experimentos descritos. Dois grupos de experimentos analisaram a qualidade da seleção de atributos para a classificação de imagens e o desempenho do modelo neural adotado pelo *framework* proposto. O desempenho geral do modelo é comparado, para ambos os grupos experimentais, com métodos estudados na literatura, utilizando-se adaptações da métrica da acurácia para tarefas de classificação. A análise dos resultados deve enfatizar o desempenho da proposta sobre os métodos comparados.

7 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados dos experimentos realizados conforme o planejamento delineado no Capítulo 6. Os experimentos avaliaram o *framework* proposto em duas tarefas em classificação de imagens: classificação binária e classificação multiclases. As tarefas foram avaliadas em três diferentes bases de dados de imagens.

A avaliação completa do *framework* considerou dois grupos de experimentos. O primeiro grupo tinha como meta avaliar a qualidade das predições a partir da quantidade de atributos selecionados para a tarefa. Diferentes modelos de seleção de atributos foram combinados com o mecanismo de aprendizagem do *framework* para permitir a comparação entre os modelos. A Seção 7.1 e a Seção 7.2 discutem os resultados desse grupo de experimentos.

O segundo grupo tinha como meta avaliar a qualidade das predições a partir do tamanho dos lotes de imagens selecionados como entrada no fluxo de dados. No *framework* proposto, o tamanho do lote é fundamental para o funcionamento do mecanismo de aprendizagem. Também neste grupo, diferentes métodos foram selecionados para permitir a comparação com os resultados gerados pela proposta. A Seção 7.3 e a Seção 7.4 mostram os resultados obtidos e discutem a proposta em relação aos objetivos desta tese.

Por fim, a Seção 7.5 apresenta as considerações finais sobre os experimentos realizados e os resultados obtidos. Em todas as seções, os resultados são discutidos em termos da métrica acurácia e suas versões. Detalhes dos resultados podem ser consultados no Apêndice A.

7.1 Seleção de Atributos para Classificação Binária

Os primeiros experimentos avaliaram o desempenho dos modelos comparados na seleção de atributos para a tarefa de classificação binária. Os experimentos consideraram a

capacidade do classificador presente no *framework* em rotular novas imagens a partir dos atributos escolhidos pelos modelos comparados.

Inicialmente, os experimentos avaliaram os modelos DFS (LI; CHEN; WASSERMAN, 2016), OFS (WANG et al., 2014) e FSDS (HUANG; YOO; KASIVISWANATHAN, 2015) juntamente com o *framework*. O mecanismo de extração de atributos do *framework* foi substituído por um dos modelos e os resultados preditivos foram comparados com o mecanismo padrão, que é a Regressão Elastic Net. Os experimentos foram realizados para diferentes quantidades de atributos nas bases de dados Food-5k, CIFAR-10 e MNIST.

Esta seção descreve os resultados dos experimentos para a classificação binária a partir das métricas AM^B e AG^B . Todos os experimentos relatados foram avaliados observando a quantidade de atributos selecionados pelo modelo de seleção de atributos. Os tamanhos comparados são $Q_t^{SA} = \{100, 200, 300, 400, 500\}$.

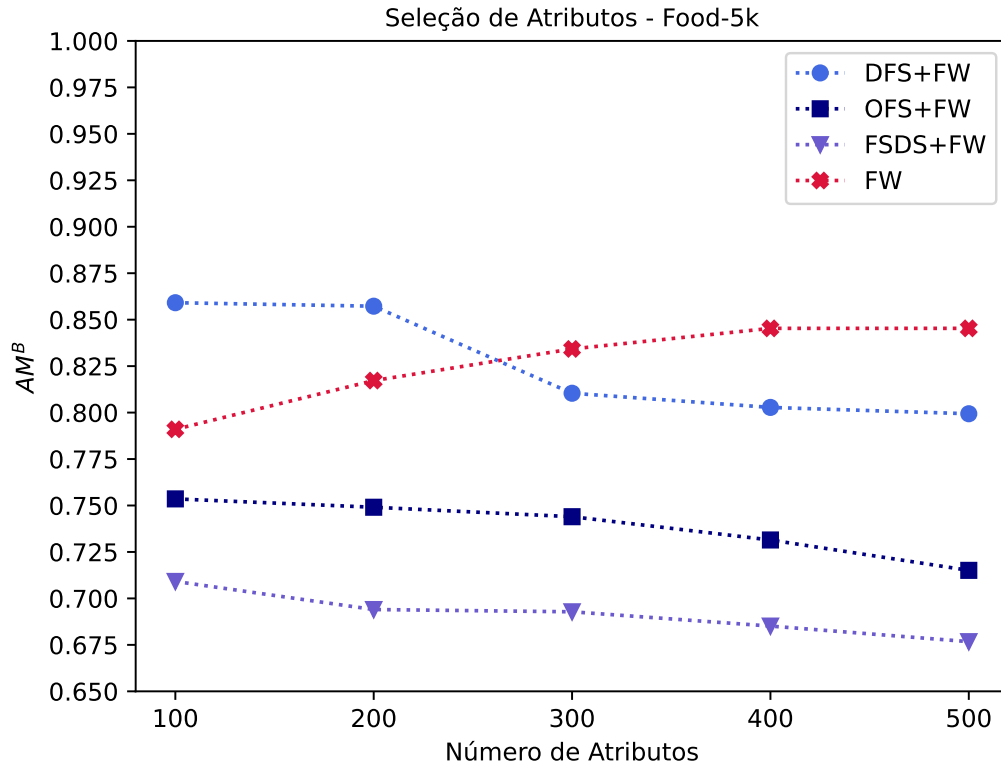
A Figura 16 compara os métodos avaliados para a base de imagens Food-5k. Na Figura 16a, a métrica AM^B mostra a taxa de acerto ponderada em cada uma das classes $\{\text{alimento}, \text{não alimento}\}$ por lote para cada método avaliado. Para $Q_t^{SA} = 100$, o modelo DFS apresenta os melhores resultados com o *framework*, obtendo o melhor desempenho dos experimentos. No entanto, à medida que o valor de Q_t^{SA} cresce, AM^B diminui. Esse fenômeno acontece com todos os métodos, exceto com a proposta original.

Quando $Q_t^{AP} = 100$, DFS+FW obtém $AM^B = 0.859$. Em $Q_t^{SA} = 500$, o modelo decaiu para $AM^B = 0.799$. As combinações OFS+FW e FSDS+FW também reportam uma pequena queda na acurácia média. Por outro lado, FW apresenta $AM^B = 0.791$ em $Q_t^{SA} = 100$, sendo inferior ao uso do modelo DFS. No entanto, em $Q_t^{SA} = 500$, obtém $AM^B = 0.845$.

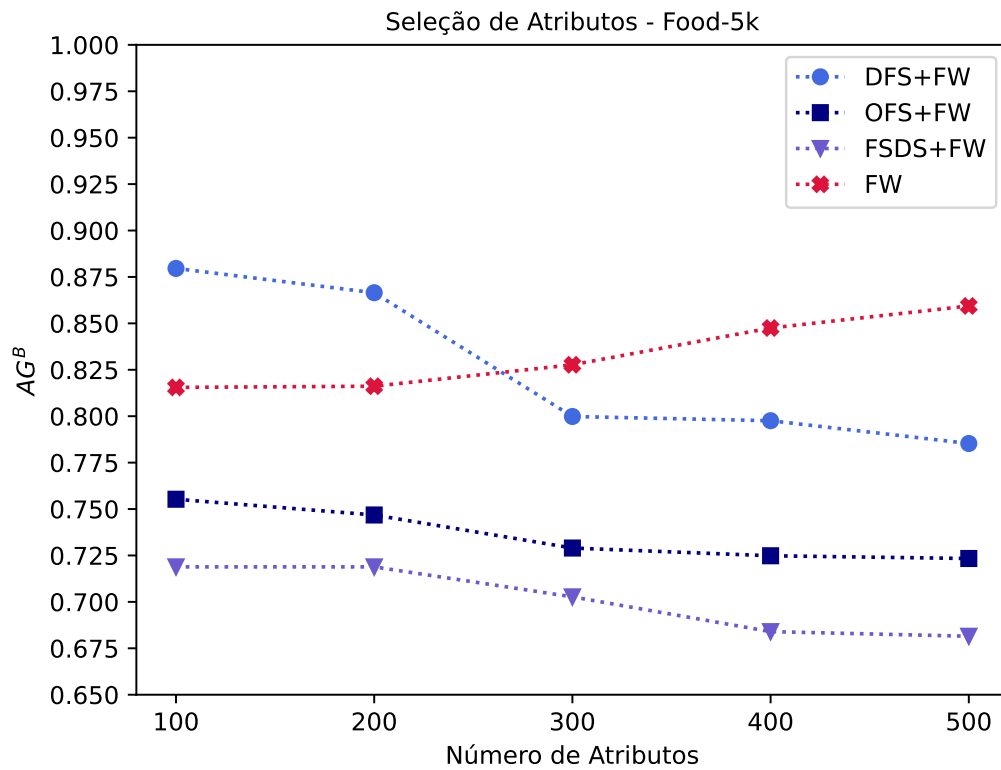
A Figura 16b apresenta um comportamento semelhante, mas menos suave. Para FW, o crescimento é mais discreto, embora com taxas maiores. Em $Q_t^{SA} = 100$, FW obtém $AG^B = 0.816$. Em $Q_t^{SA} = 500$, FW cresce para $AG^B = 0.859$. DFS+FW segue caindo à medida que Q_t^{SA} aumenta, mas ainda obtém os melhores resultados para $Q_t^{SA} = \{100, 200\}$. Para essa mesma métrica, OFS+FW apresenta um crescimento leve, mas maior estabilidade a partir de $Q_t^{SA} \geq 300$. No caso da combinação FSDS+FW, o gráfico apresenta instabilidade do método de seleção de atributos.

A Figura 17 compara os modelos na base de imagens CIFAR-10. Na Figura 17a, para $Q_t^{SA} = 100$, DFS+FW obtém $AM^B = 0.849$, e FW obtém $AM^B = 0.781$. Em $Q_t^{SA} = 400$, DFS+FW obtém $AM^B = 0.820$, e FW obtém $AM^B = 0.799$. Essas pequenas diferenças em $Q_t^{SA} = \{100, 400\}$ entre os modelos confirmam a competitividade da RNA. Embora

Figura 16: Desempenho preditivo do *framework* usando modelos de seleção de atributos sobre a base de dados Food-5k para a tarefa de classificação binária: (a) AM^B e (b) AG^B , conforme o número de atributos selecionados.



(a)



(b)

o modelo usado por DFS seja mais sofisticado, a Regressão Elastic Net do *framework* é mais competitiva na presença de mais imagens para treinamento. Em $Q_t^{SA} = 500$, FW alcança o melhor desempenho do experimento, obtendo $AM^B = 0.859$.

Na Figura 17b, FW supera o modelo DFS na disputa anterior em $Q_t^{SA} = \{100, 400\}$. Em $Q_t^{SA} = 100$, FW obtém $AG^B = 0.856$, enquanto DFS+FW obtém $AG^B = 0.848$. Em $Q_t^{SA} = 400$, FW obtém $AG^B = 0.849$, enquanto DFS+FW obtém $AG^B = 0.821$. Finalmente, quando $Q_t^{SA} = 500$, FW obtém $AM^B = 0.856$, o mesmo desempenho para a seleção de apenas 100 atributos. Em ambos os gráficos, FW aparece como um método mais eficiente na seleção de conjuntos de atributos maiores.

A Figura 18 compara a aplicação dos modelos na base de imagens MNIST. Também para essa base de imagens, a principal disputa do *framework* proposto é com o modelo DFS. Na Figura 18a, a combinação DFS+FW supera em grande vantagem FW para $Q_t^{SA} \leq 300$. No entanto, FW alcança o melhor desempenho do experimento em $Q_t^{SA} = 500$, obtendo $AM^B = 0.863$. Os experimentos nas bases de imagens anteriores mostram que FW é mais eficiente na seleção de maiores quantidades de atributos.

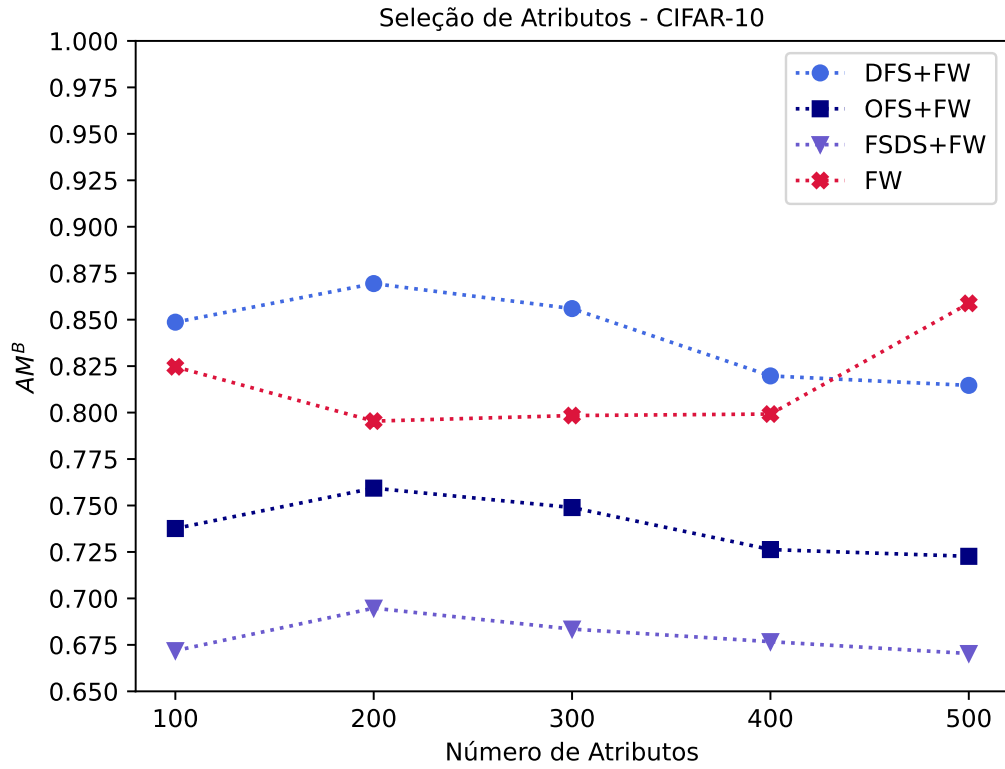
Na Figura 18b, a pequena diferença anteriormente relatada entre DFS e FW apresenta um novo resultado. Em $Q_t^{SA} = 100$, FW obtém $AG^B = 0.856$, enquanto DFS+FW obtém $AG^B = 0.848$. O modelo DFS continua com vantagem para o intervalo $Q_t^{SA} = \{100, 200, 300\}$. No entanto, para $Q_t^{SA} = \{400, 500\}$, o modelo passa a decrescer em desempenho. Nesse mesmo intervalo, FW cresce, chegando a $AG^B = 0.856$ em $Q_t^{SA} = 500$.

Nas base de imagens analisadas, os modelos DFS e FW competem em diferentes cenários de maneira mais acirrada do que os demais modelos. Embora DFS e FW tenham apresentado melhores resultados na três bases de imagens, eles são mais instáveis na seleção de atributos que OFS e FSDS.

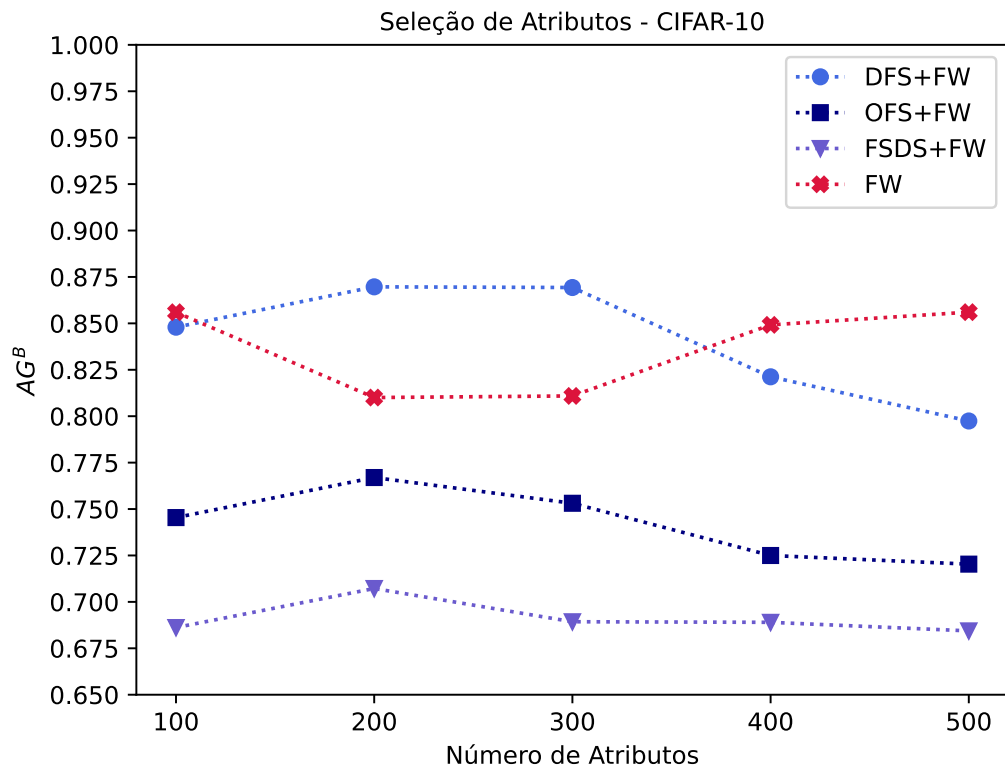
Mesmo com experimentos prévios em CIFAR-10, o modelo FSDS não apresentou resultados significativos na classificação binária realizada em combinação com o *framework* proposto. Apesar dos modelos serem parecidos, FSDS não soube lidar com as janelas de imagens que foram apresentadas em modo *online*, descartando conceitos significativos para a seleção de atributos.

No caso do modelo OFS, os resultados mostram que a proposta de seleção *online* de atributos é bastante vantajosa para problemas de classificação em fluxos de dados. Essa abordagem é defendida nesta tese. No entanto, o desafio do modelo foi entender as estruturas de dados não estruturados sendo processadas pelo fluxo de entrada. Uma

Figura 17: Desempenho preditivo do *framework* usando modelos de seleção de atributos sobre a base de dados CIFAR-10 para a tarefa de classificação binária: (a) AM^B e (b) AG^B , conforme o número de atributos selecionados.

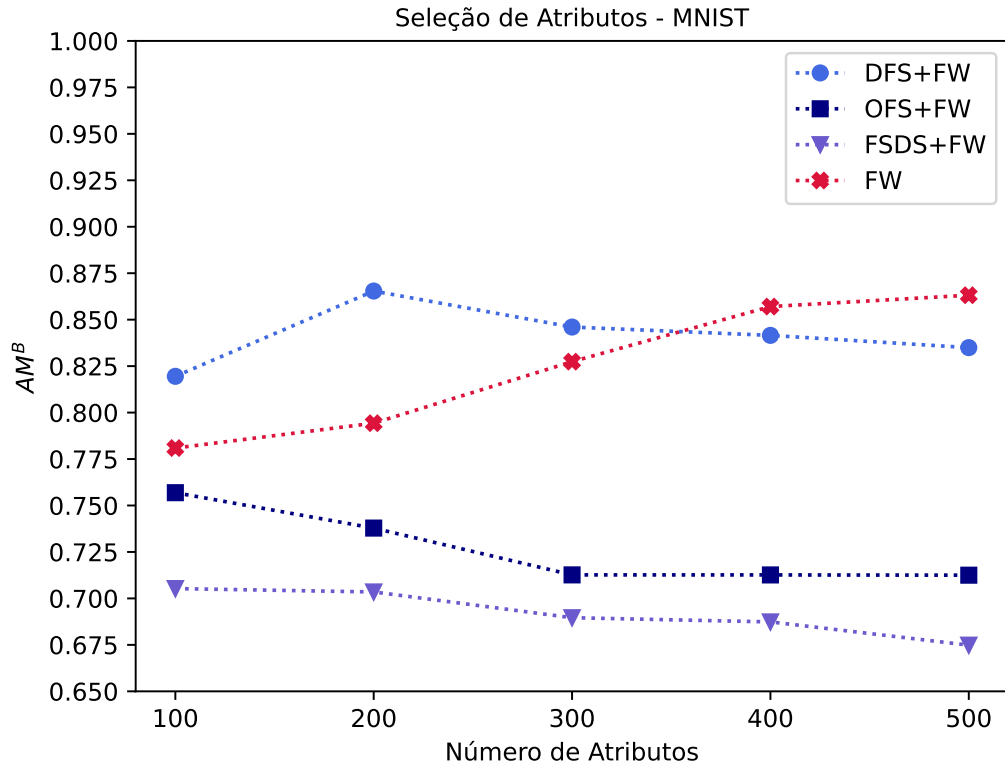


(a)

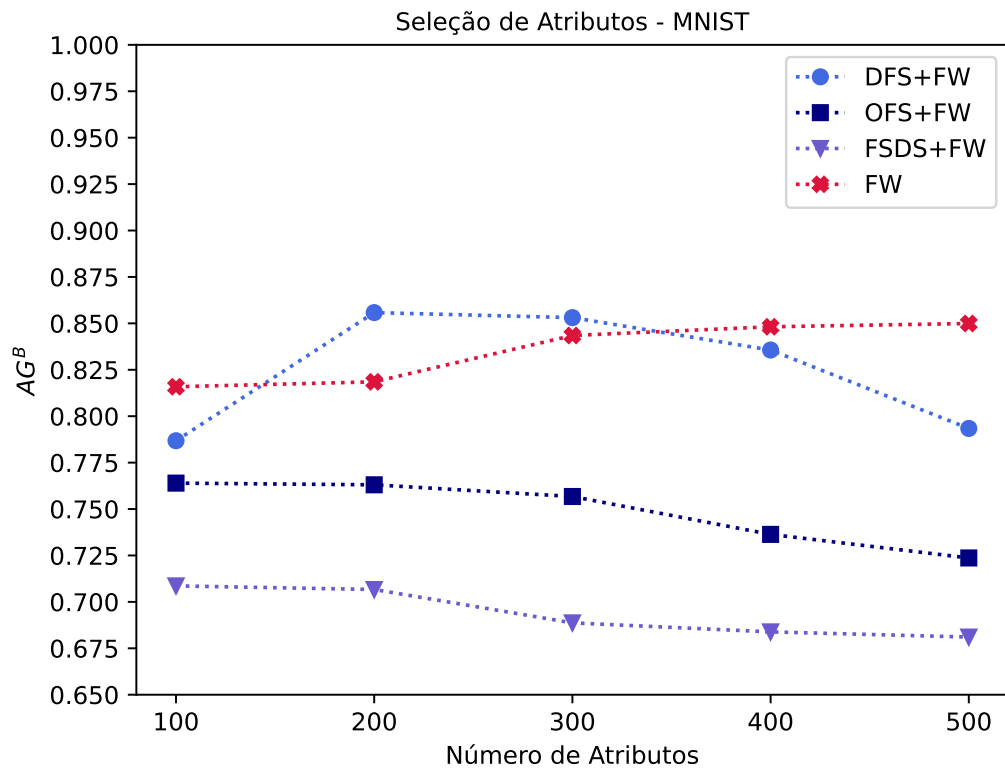


(b)

Figura 18: Desempenho preditivo do *framework* usando modelos de seleção de atributos sobre a base de dados MNIST para a tarefa de classificação binária: (a) AM^B e (b) AG^B , conforme o número de atributos selecionados.



(a)



(b)

heurística ou adaptação ao processamento de imagens poderia melhorar os resultados dessa abordagem.

Discussão: o mecanismo de seleção de atributos adotado pelo *framework* proposto consiste em um método de regressão Elastic Net. A adoção desse método para a classificação binária mostrou como a aprendizagem incremental lidou favoravelmente com a seleção e redução de atributos, destacando coeficientes relevantes sobre o lote de imagens.

7.2 Seleção de Atributos para Classificação Multiclasses

O segundo conjunto de experimentos do primeiro grupo avaliou a seleção de atributos do *framework* proposto nas tarefas de classificação multiclasses. Nos experimentos, as métricas AM^M e AG^M foram computadas para as dez classes das bases de imagens CIFAR-10 e MNIST.

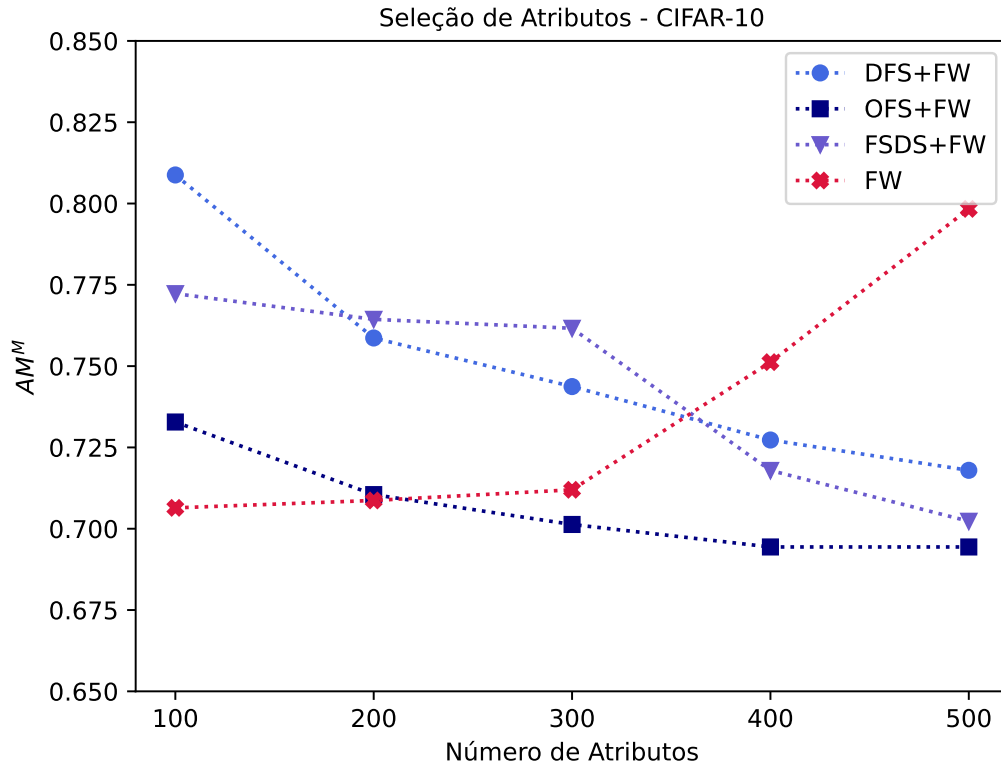
Os modelos DFS (LI; CHEN; WASSERMAN, 2016), OFS (WANG et al., 2014) e FSDS (HUANG; YOO; KASIVISWANATHAN, 2015) também foram avaliados juntamente com o *framework* para a classificação multiclasses. Todos os experimentos relatados foram avaliados observando a quantidade de atributos selecionados pelo modelo de seleção de atributos, sendo os tamanhos comparados em $Q_t^{SA} = \{100, 200, 300, 400, 500\}$.

A Figura 19 apresenta os resultados obtidos nas duas métricas comparadas para os diferentes modelos na base de imagens CIFAR-10. De maneira geral, o *framework* proposto mostra dificuldades para a classificação multiclasses de imagens a partir da seleção de poucos atributos.

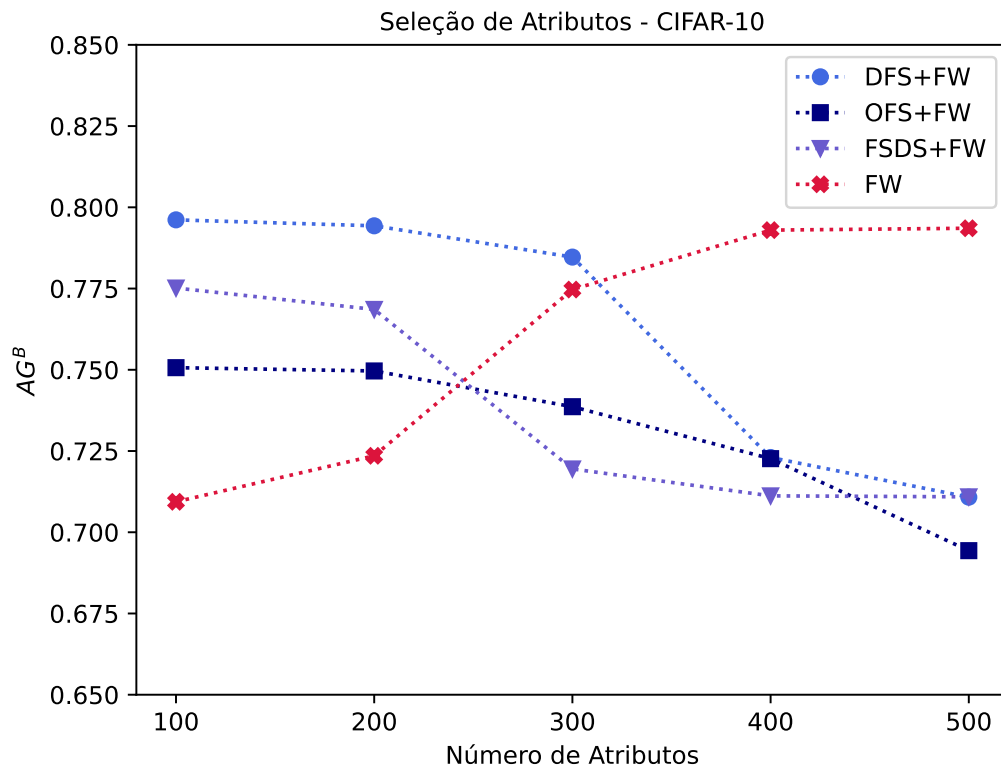
Na Figura 19a, FW apresenta o menor desempenho em $Q_t^{SA} = 100$, obtendo $AM^M = 0.706$. Para o mesmo valor de Q_t^{SA} , a combinação DFS+FW obtém $AM^M = 0.809$, sendo 14,6% melhor que a proposta. No entanto, o desempenho de FW cresce à medida que cresce também a quantidade de atributos selecionados para a RNA. Já em $Q_t^{SA} = 400$, FW supera os demais modelos. Em $Q_t^{SA} = 500$, FW obtém $AM^M = 0.798$, aproximando-se do primeiro resultado de DFS+FW.

Na Figura 19b, os resultados mostram a importância da experimentação para a classificação multiclasses. Enquanto FSDS+FW mostra excelente desempenho na acurácia média para $Q_t^{SA} \leq 300$, o modelo FSDS não apresenta resultados equivalentes na acurácia geral. Em $Q_t^{SA} = 300$, o modelo obtém o melhor resultado em AM^M . Porém, obtém apenas $AG^M = 0.719$, sendo o pior modelo para o valor de Q_t^{SA} informado.

Figura 19: Desempenho preditivo do *framework* usando modelos de seleção de atributos sobre a base de dados CIFAR-10 para a tarefa de classificação multiclasse: (a) AM^M e (b) AG^B , conforme o número de atributos selecionados.

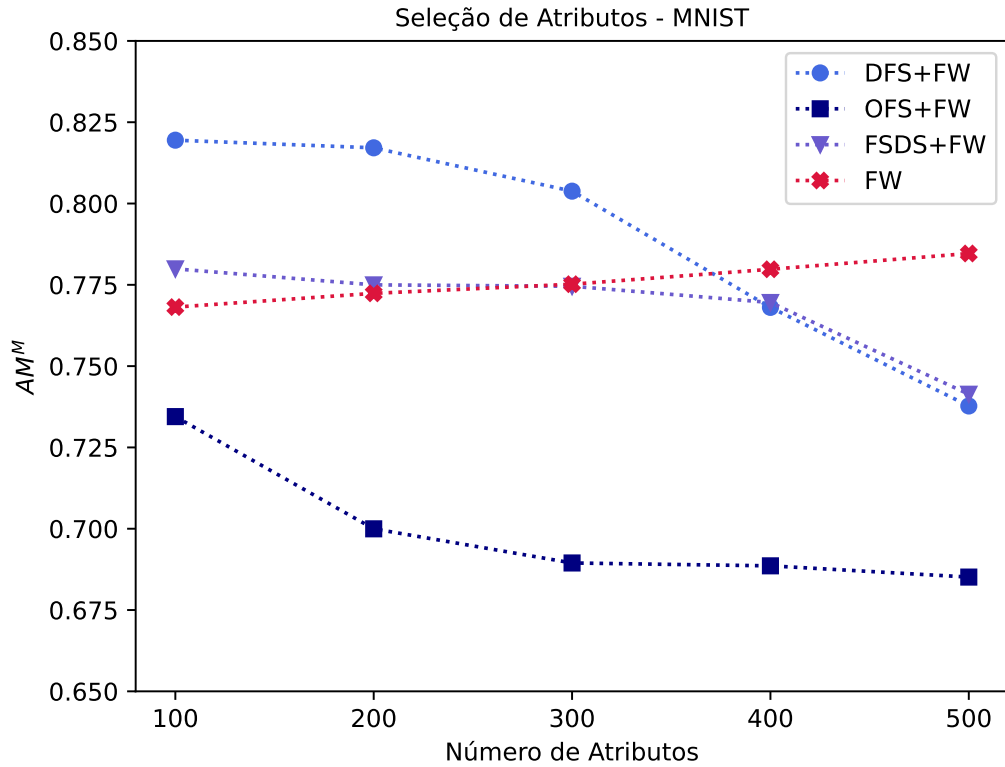


(a)

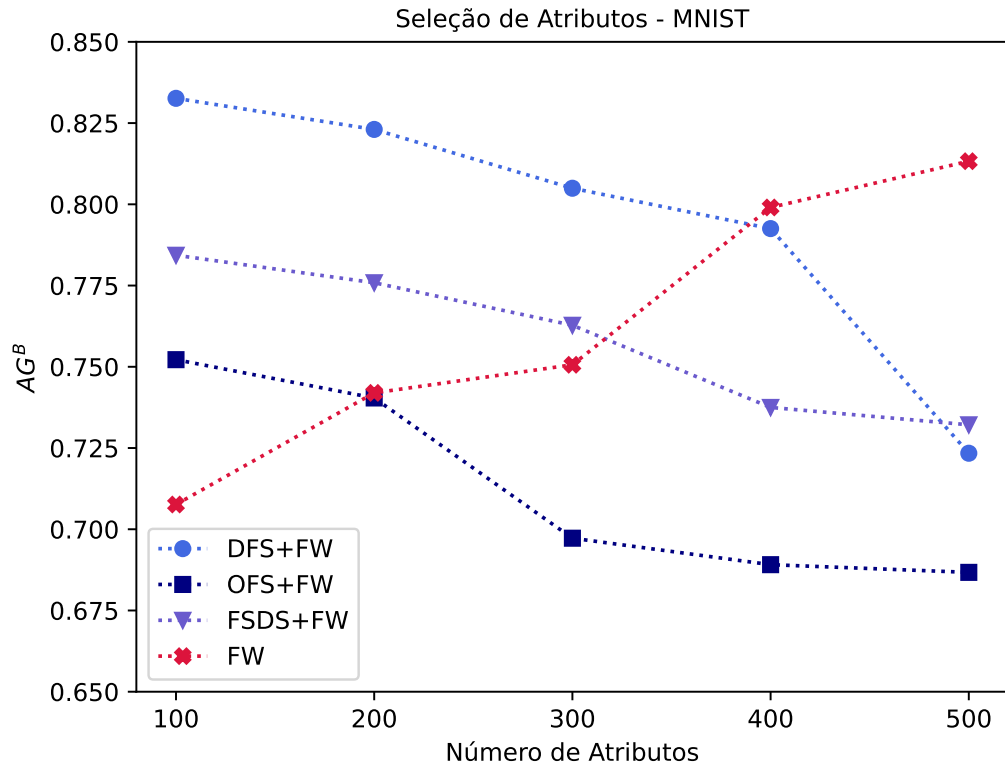


(b)

Figura 20: Desempenho preditivo do *framework* usando modelos de seleção de atributos sobre a base de dados MNIST para a tarefa de classificação multiclases: (a) AM^M e (b) AG^M , conforme o número de atributos selecionados.



(a)



(b)

A proposta FW apresenta resultados individuais melhores para as distintas classes em AG^M quando $Q_t^{SA} = \{200, 300\}$. Em $Q_t^{SA} = 200$, obtém $AG^M = 0.724$. Em $Q_t^{SA} = 400$, o desempenho cresce 9,5%, alcançando $AG^M = 0.793$. Finalmente, em $Q_t^{SA} = 500$, obtém $AG^M = 0.794$, atingindo o melhor resultado do experimento para a base CIFAR-10.

A Figura 20 apresenta os resultados obtidos na base de imagens MNIST. Para essa base de imagens, o *framework* proposto apresenta melhores resultados para classificação multiclases em relação à base CIFAR-10. Tanto para AM^M quanto para AG^M , FW compete diretamente com os modelos DFS e FSDS.

Na Figura 20a, FW compete diretamente com FSDS para $Q_t^{SA} \leq 300$. Para esse intervalo, DFS alcança os melhores resultados, obtendo $AM^M = 0.819$ em $Q_t^{SA} = 100$. No entanto, para $Q_t^{SA} = \{400, 500\}$, FW supera os demais métodos. Em $Q_t^{SA} = 500$, obtém $AM^M = 0.785$. Para a mesma quantidade de atributos, OFS tem o pior resultado do experimento, obtendo $AM^M = 0.685$.

Na Figura 20b, os diferentes modelos são comparados a partir da acurácia geral. Essa métrica denota com maior clareza como o desempenho de FW cresce à medida que Q_t^{SA} aumenta. De $AG^M = 0.708$ em $Q_t^{SA} = 100$, FW obtém $AG^M = 0.813$ em $Q_t^{SA} = 500$. Esse crescimento de quase 15% na acurácia geral mostra como o *framework* proposto deve lidar bem com imagens de alta dimensão.

Os demais modelos apresentaram queda constante no desempenho após $Q_t^{SA} > 100$. Em $Q_t^{SA} = 100$, DFS obtém $AG^M = 0.833$, OFS obtém $AG^M = 0.752$ e FSDS obtém $AG^M = 0.784$. Porém, em $Q_t^{SA} = 500$, os valores caem até 13%. Para esse intervalo, DFS obtém $AG^M = 0.723$ (-13,12%), OFS obtém $AG^M = 0.687$ (-8,70%) e FSDS obtém $AG^M = 0.732$ (-6,65%).

Por fim, os resultados mostram que o *framework* proposto nesta tese alcança até 79,8% de acurácia geral e até 81,3% de acurácia média para a classificação de imagens em problemas multiclases. Tanto para CIFAR-10 quanto para MNIST, os melhores resultados aconteceram quando $Q_t^{SA} = \{400, 500\}$ para AM^M e AG^M . Alguns dos modelos comparados superaram esses resultados quando $Q_t^{SA} = 100$.

Discussão: o mecanismo de seleção de atributos do *framework* proposto apresenta bons resultados preditivos para a classificação multiclases. Os melhores resultados são obtidos para grandes quantidades de atributos selecionados pela regressão Elastic Net. Neste bloco de experimentos, a aprendizagem *online* mostrou ser capaz de destacar os coeficientes mais relevantes da regressão para cada lote de imagens selecionado, permitindo acurácias

elevadas. Embora a redução de atributos tenha sido eficiente, o principal desafio do *framework* foi a distinção das imagens em diferentes classes.

7.3 Aprendizado para Classificação Binária de Imagens

O segundo grupo de experimentos avaliou o desempenho preditivo dos modelos neurais construídos pelo *framework*. O foco desse grupo de experimentos é mensurar a capacidade preditiva da RNA em classificar corretamente as imagens do fluxo de dados. Os experimentos do grupo foram avaliados para as tarefas de classificação binária e classificação multiclases para as bases de dados de imagens analisadas.

Esta seção descreve os resultados dos experimentos para a classificação binária a partir das métricas AM^B e AG^B . Todos os experimentos relatados foram avaliados observando a variação no tamanho do lote das imagens selecionadas pelo fluxo de dados para a RNA. Os tamanhos dos lotes são $Q_t^{AP} = \{100, 200, 300, 400, 500\}$.

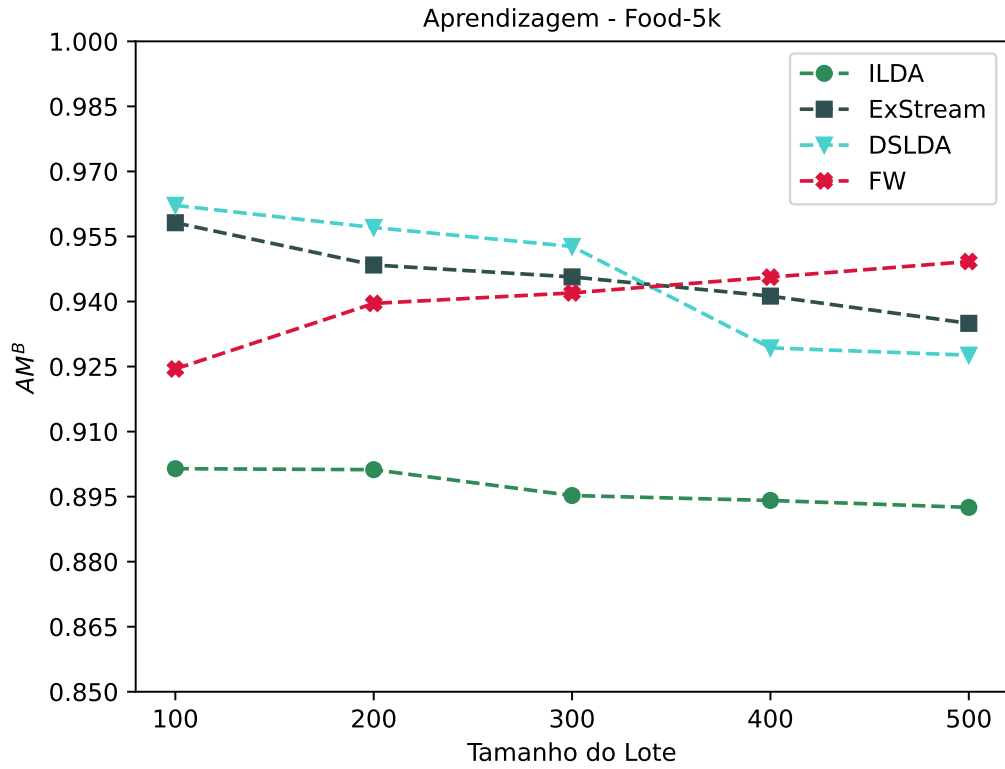
A Figura 21 ilustra o desempenho preditivo dos métodos comparados para a base de imagens Food-5k. Na Figura 21a, a métrica AM^B mostra a taxa de acerto ponderada em cada uma das classes $\{\text{alimento}, \text{não alimento}\}$ por lote para cada método avaliado. Para $Q_t^{AP} = \{100, 200, 300\}$, o método DSLDA apresenta os melhores resultados. Quando $Q_t^{AP} = 100$, DSLDA obtém $AM^B = 0.962$, sendo a melhor acurácia para o intervalo e também para o experimento.

No entanto, para $Q_t^{AP} = \{400, 500\}$, FW obtém os melhores resultados. Quando $Q_t^{AP} = 400$, FW obtém $AM^B = 0.946$. Quando $Q_t^{AP} = 500$, FW obtém $AM^B = 0.949$, um resultado ainda melhor. O método é o que mais cresce em desempenho à medida que o tamanho do lote aumenta. Os demais métodos mostram-se menos estáveis em tamanhos de lote maiores, especialmente DSLDA e ExStream.

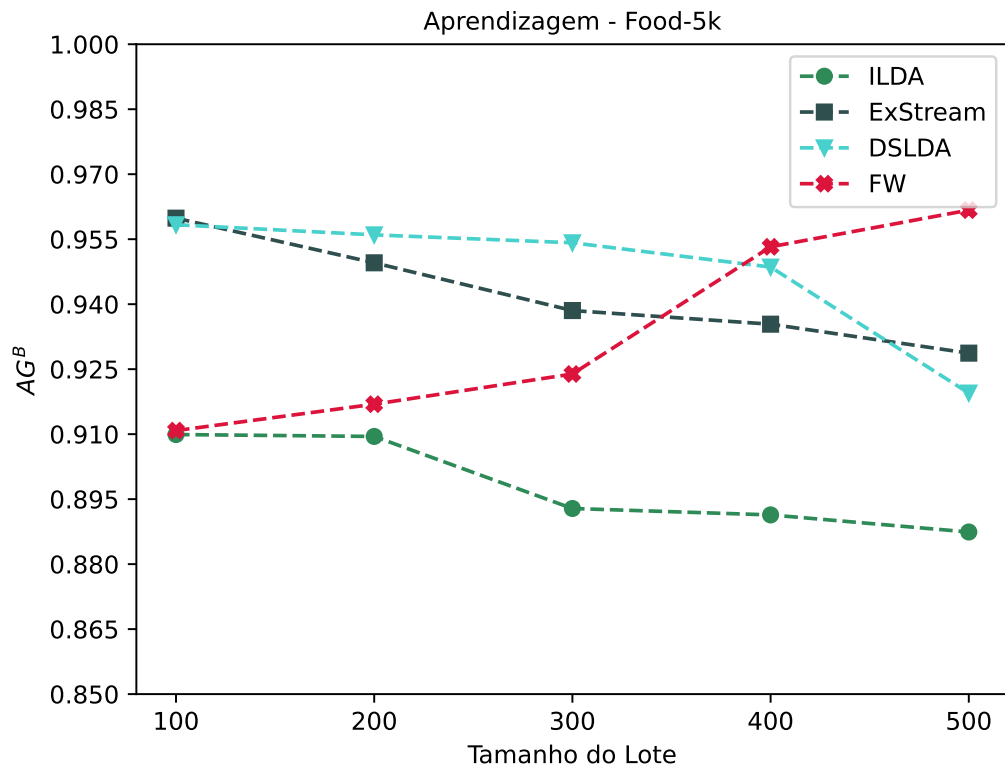
Na Figura 21b, o desempenho dos métodos comparados é ilustrado a partir do cálculo da métrica AG^B , que corresponde à taxa de acerto geral de cada uma das classes $\{\text{alimento}, \text{não alimento}\}$ por lote. Também nessa figura, DSLDA mostra instabilidade quando $Q_t^{AP} = 500$. Em contraste, FW segue crescendo para $Q_t^{AP} = \{300, 400, 500\}$. Quando $Q_t^{AP} = 500$, FW obtém $AG^B = 0.962$.

Na base de imagens Food-5k, a principal competição com o *framework* proposto FW é com o método ExStream. Ambos apresentam bom desempenho para tamanhos de lotes maiores. No entanto, FW apresenta melhor estabilidade à medida que o tamanho Q^{AP} do lote cresce. Como ExStream usa uma técnica de agrupamento para reunir estatísticas

Figura 21: Desempenho preditivo dos métodos avaliados sobre a base de dados Food-5k: (a) AM^B e (b) AG^B , conforme o tamanho do lote.



(a)



(b)

sobre as imagens, FW trabalha diretamente com a distribuição das imagens sobre as classes.

A Figura 22 ilustra o desempenho preditivo dos métodos comparados para a base de imagens CIFAR-10. Na Figura 22a, a métrica AM^B mostra a taxa de acerto ponderada em cada uma das classes $\{cachorro, gato\}$ por lote para cada método avaliado. Para $Q_t^{AP} = \{100, 200, 300\}$, o método DSLDA também apresenta os melhores resultados. Quando $Q_t^{AP} = 100$, DSLDA obtém $AM^B = 0.969$, sendo a melhor acurácia para o intervalo e também para o experimento.

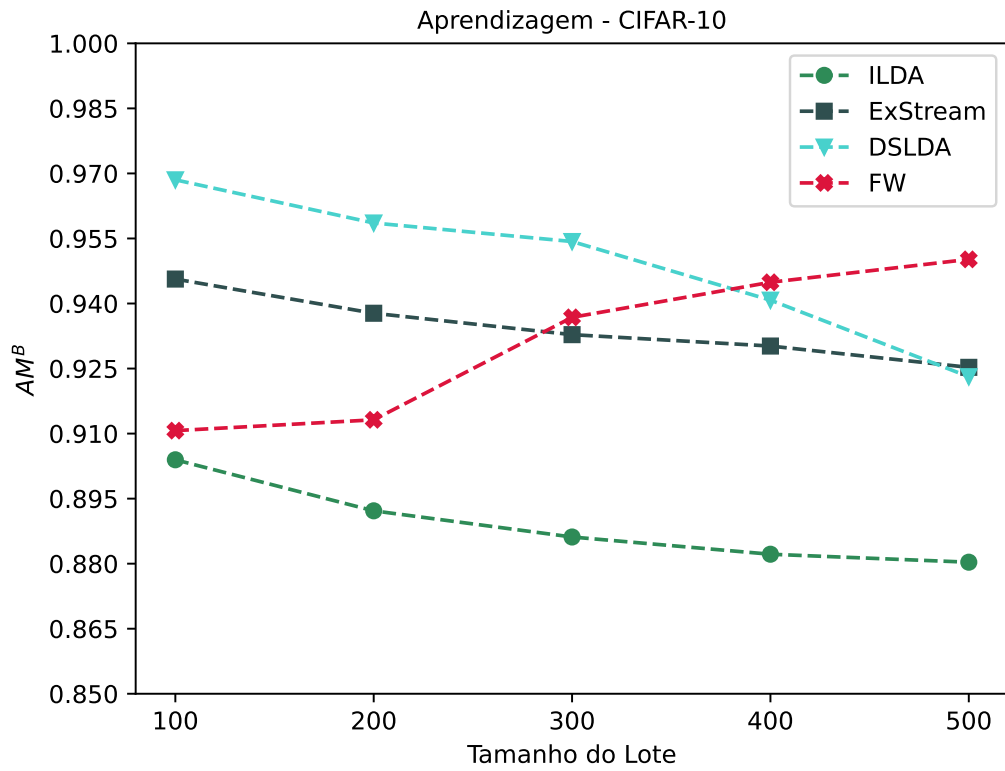
No entanto, em $Q_t^{AP} = 500$, DSLDA fica abaixo de ExStream e também FW, que apresenta $AM^B = 0.950$. Também na base de imagens CIFAR-10, o método FW cresce à medida que o tamanho do lote aumenta. Os demais métodos mostram maior instabilidade, especialmente ExStream e ILDA.

A métrica AG^B na Figura 22b também mostra o crescimento da acurácia para FW conforme o aumento na quantidade de imagens selecionadas para o lote. Quando $Q_t^{AP} = 100$, FW obtém $AG^B = 0.911$. Em $Q_t^{AP} = 500$, FW alcança $AG^B = 0.955$, um aumento de 4,83%, enquanto DSLDA decresce em quase 4,33%.

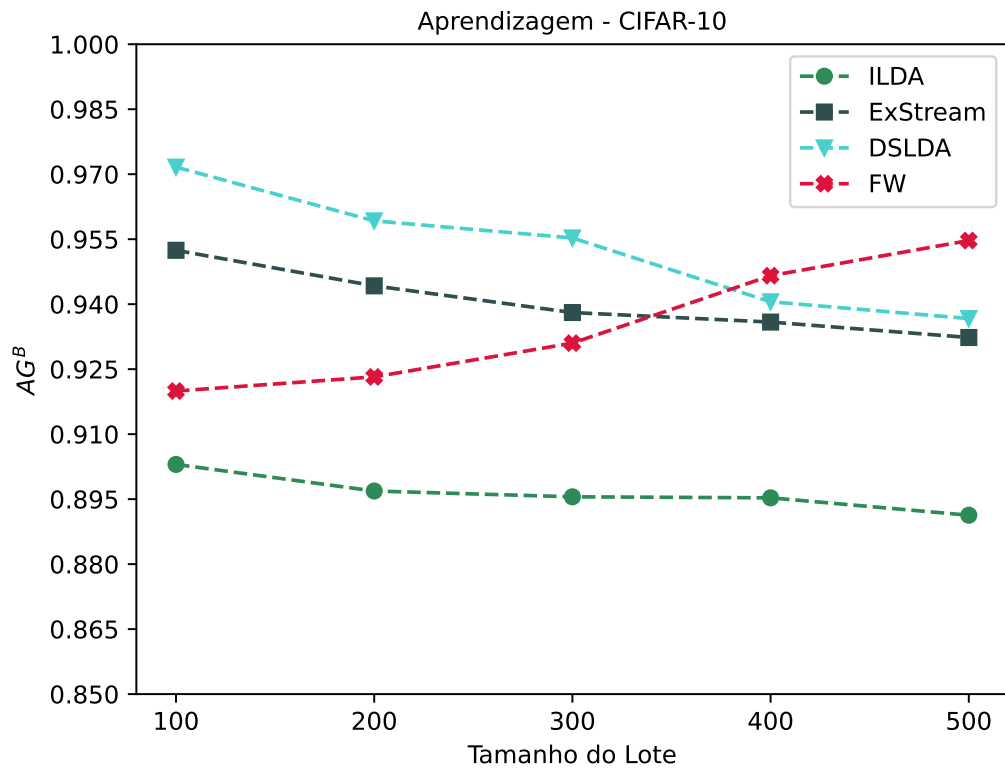
Na base de imagens CIFAR-10, o principal competidor com FW é o método DSDLA. Embora DSDLA dispute o segundo lugar em $Q_t^{AP} \leq 300$ mais com ExStream do que com FW, os resultados mostram que o *framework* proposto desempenha melhor a aprendizagem da rede neural em conjuntos de dados com volumes maiores de imagens. O método DSDLA não esquece completamente o que observou no fluxo de dados, impedindo que novos conceitos sejam atualizados em lotes maiores. No *framework* proposto, o mecanismo *online* atua para que a janela seja constantemente atualizada, permitindo que a classificação obtenha melhores respostas em janelas maiores.

A Figura 23 ilustra o desempenho preditivo dos métodos comparados para a base de imagens MNIST. Na Figura 23a, a métrica AM^B mostra a taxa de acerto ponderada em cada uma das classes $\{0, 1\}$ por lote para cada método avaliado. Para $Q_t^{AP} = \{100, 200, 300\}$, o método DSLDA também apresenta os melhores resultados. Quando $Q_t^{AP} = 100$, DSLDA obtém $AM^B = 0.969$, sendo a melhor acurácia para o tamanho do lote e também para o experimento. No entanto, quando $Q_t^{AP} = 500$, o *framework* proposto alcança melhor desempenho, obtendo $AM^B = 0.953$. Para o mesmo tamanho de lote, DSLDA obtém $AM^B = 0.923$, perdendo para o método ExStream, que alcança $AM^B = 0.925$.

Figura 22: Desempenho preditivo do *framework* sobre a base de dados CIFAR-10: (a) AM^B e (b) AG^B , conforme o tamanho do lote.

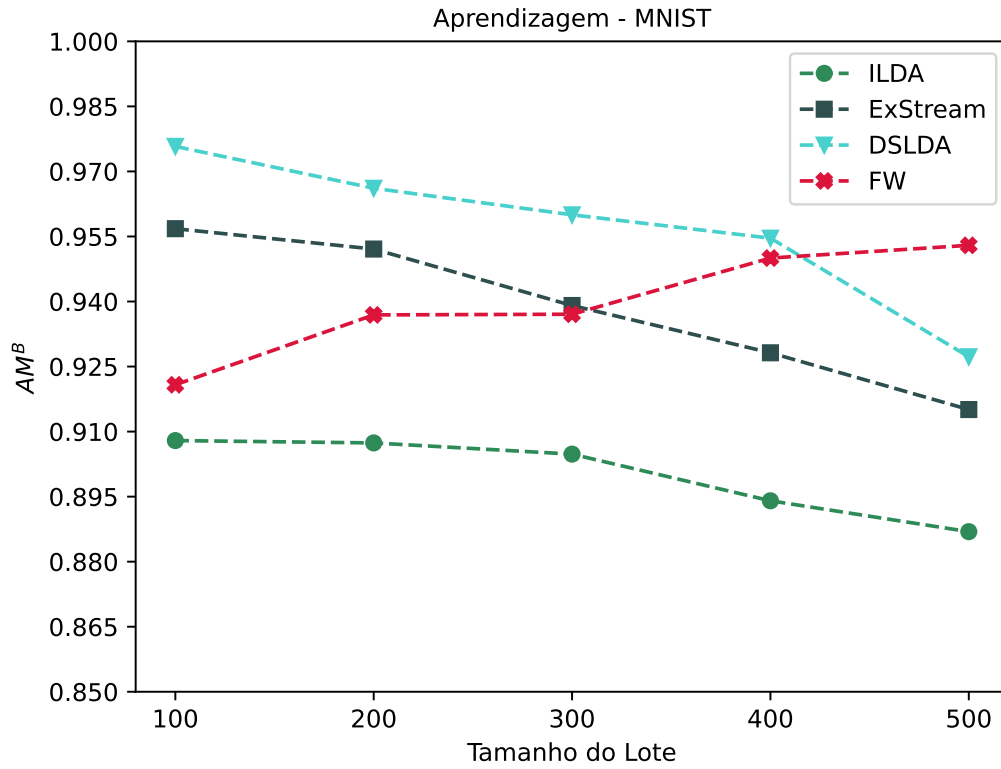


(a)

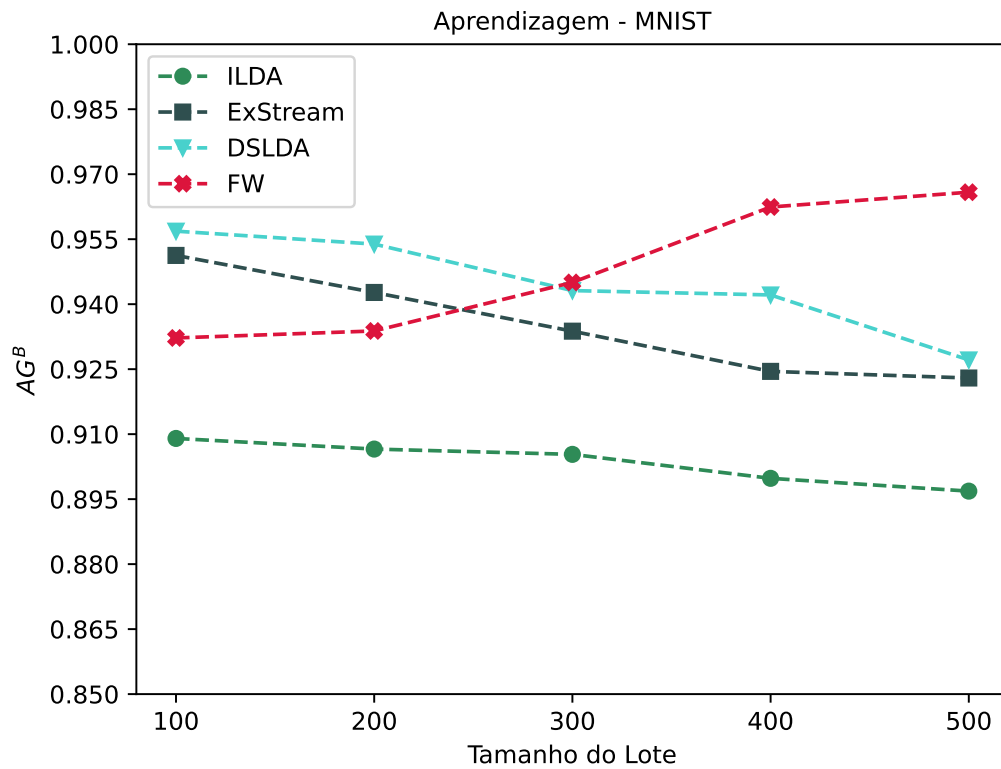


(b)

Figura 23: Desempenho preditivo do *framework* sobre a base de dados MNIST: (a) AM^B e (b) AG^B , conforme o tamanho do lote.



(a)



(b)

Embora a diferença de AG^B dos métodos DSLDA e ExStream para o método proposto FW seja relativamente pequena, a análise de todo o gráfico mostra como os métodos comparados são instáveis para lotes de imagens maiores. Não apenas porque obtêm resultados inferiores que FW para $Q_t^{AP} = 500$, mas também porque seguem em queda à medida que o tamanho do lote aumenta.

A acurácia geral observada na Figura 23b mostra que essa instabilidade continua existindo, mesmo para uma métrica que controle melhor o desbalanceamento dos dados observados. Em $Q_t^{AP} = 500$, FW alcança o melhor desempenho do experimento, obtendo $AG^B = 0.966$. Por outro lado, o método ILDA obtém o pior desempenho na base de imagens MNIST, alcançando seu melhor resultado em $Q_t^{AP} = 100$, com $AG^B = 0.909$.

Os métodos comparativos ILDA, ExStream e DSLDA apresentam abordagens distintas para a tarefa de classificação binária de imagens. No entanto, todos os métodos apresentam o mesmo comportamento à medida que a quantidade de imagens aumenta no lote, que é a queda nas acurácias AM^B e AG^B . Apesar da semelhança entre as métricas, elas estão preparadas para captar as diferenças entre a classificação nas classes binárias do problema.

No caso do método ExStream, esse comportamento acontece por causa da formação de grupos sem supervisão que antecede a aprendizagem. O mecanismo de agrupamento utilizado pelo método é utilizado para permitir a aprendizagem de novos conceitos. Porém, para as bases de dados analisadas, ExStream não atualiza tão rapidamente novos grupos em contextos sem mudança de conceitos.

No caso do método DSLDA, o fenômeno acontece por razões semelhantes. Como o método é projetado para lidar com mudança de conceitos em fluxos de dados, a seleção de atributos não reconhece rapidamente a distribuição de classes em fluxos de dados pouco ou nada evolutivos. O fenômeno do esquecimento catastrófico é mitigado, mas a aprendizagem passa a ignorar imagens já observadas.

No caso do método ILDA, o mecanismo de seleção de atributos é bastante semelhante à implementação no *framework* proposto. Embora o método seja estável na classificação, todos os resultados foram aquém dos demais métodos. Uma vez que os atributos são extraídos das imagens a partir da constante fatoração da matriz de dados, o algoritmo passa a ignorar novas entradas, enfatizando aquelas já existentes. Dessa forma, lotes maiores de imagens podem fazer com que características antigas sejam descartadas.

Por fim, o método FW não apresentou os melhores desempenhos para lotes pequenos. Uma vez que a janela de imagens mantém uma pequena porção de imagens em memória,

a aprendizagem não conseguiu lidar com a regressão Elastic Net, que seleciona e reduz os dados de imagem. No entanto, o método se mostrou eficiente para o maior número de imagens no lote ($Q_t^{AP} = 500$). FW supera os demais métodos, mesmo em conjuntos de dados menores.

Discussão: o *framework* proposto apresenta um método eficiente para a classificação binária de imagens em fluxos de dados. A comparação com outros métodos presentes no estado-da-arte mostra que a proposta é mais estável na extração de atributos relevantes como entrada para uma Rede Neural Convolutacional. No *framework*, a regressão Elastic Net inicialmente definida seleciona rapidamente e de maneira *online* os atributos das imagens presentes no lote, capturando a janela mais atualizada. Isso permite que as camadas convolucionais e densas da RNA classifiquem corretamente a imagem antes que um novo conceito apareça.

7.4 Aprendizado para Classificação Multiclasses de Imagens

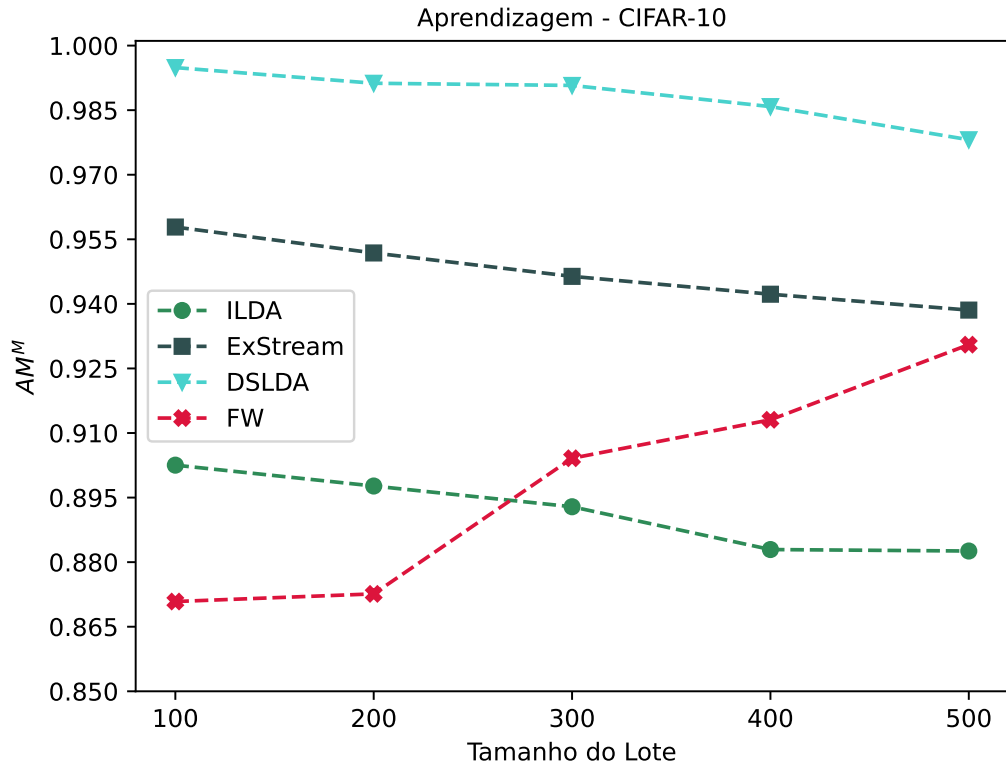
O segundo grupo de experimentos também avaliou o desempenho preditivo dos modelos neurais construídos pelo *framework* em tarefas de classificação multiclasses. Esta seção descreve os resultados dos experimentos a partir das métricas AM^M e AG^M . Todos os experimentos observaram a variação no tamanho do lote das imagens selecionadas pelo fluxo de dados para a RNA. Os tamanhos dos lotes são $Q_t^{AP} = \{100, 200, 300, 400, 500\}$.

A Figura 24 ilustra o desempenho preditivo dos métodos comparados para a base de imagens CIFAR-10. Ao contrário dos resultados na classificação binária, FW não alcança o melhor desempenho em $Q_t^{AP} = 500$. Embora o método continue sendo aquele com maior crescimento, FW permanece atrás de DSLDA e ExStream em todo o período de avaliação.

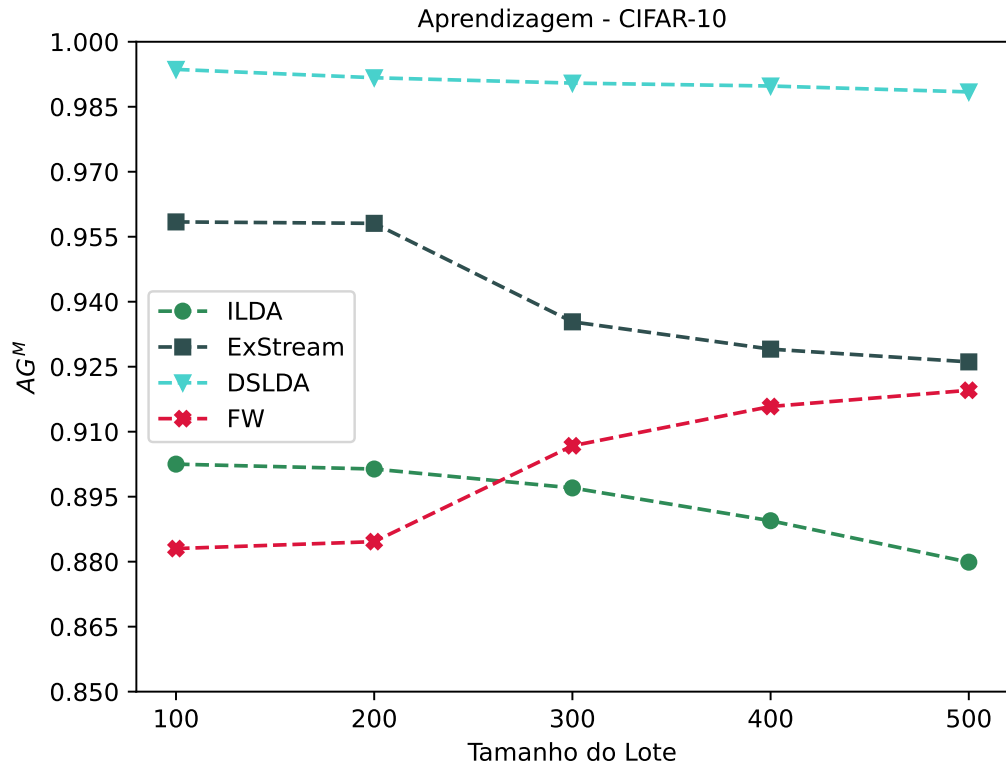
Essa análise inicial mostra a necessidade de ajustes no *framework* para permitir a classificação de imagens em multiclasses. Enquanto DSLDA e ExStream aproveitam melhor os dados passados para não prejudicar o desempenho, FW enfatiza a configuração incremental da aprendizagem para mitigar o esquecimento catastrófico. Dessa forma, ajustes no parâmetro α da regressão Elastic Net poderão melhorar o desempenho do *framework*.

A acurácia média observada na Figura 24a mostra que o método DSLDA obtém $AM^M = 0.995$ em $Q_t^{AP} = 100$, sendo o melhor resultado experimental deste trabalho. Além disso, o desempenho se mantém constante e estável durante todo o experimento. A proposta FW obtém inicialmente o pior desempenho, com $AM^M = 0.871$ em $Q_t^{AP} = 100$.

Figura 24: Desempenho preditivo do *framework* usando modelos de seleção de atributos sobre a base de dados CIFAR-10: (a) AM^M e (b) AG^M , conforme o número de atributos selecionados.



(a)



(b)

Embora o resultado alcance quase 90% de acertos, a comparação com os demais métodos mostra como FW fornece melhores resultados para problemas de classificação binária.

Em $Q_t^{AP} = 500$, FW melhora quase 7% a partir do lote inicial e obtém $AM^M = 0.931$. FW continua sendo o único método que melhora o desempenho à medida que o tamanho do lote aumenta. ExStream cai de $AM^M = 0.958$ em $Q_t^{AP} = 100$ para $AM^M = 0.939$ em $Q_t^{AP} = 500$. ILDA também cai de $AM^M = 0.903$ em $Q_t^{AP} = 100$ para $AM^M = 0.883$ em $Q_t^{AP} = 500$, obtendo desempenhos inferiores à proposta $Q_t^{AP} \geq 300$.

Na Figura 24b, a acurácia geral na base de imagens CIFAR-10 mostra que o *framework* proposto aprende melhor em contextos com menos classes. Embora nessa avaliação FW também seja o único método a aumentar o desempenho à medida que o tamanho do lote aumenta, ExStream e DSLDA superam os resultados para a classificação multiclases.

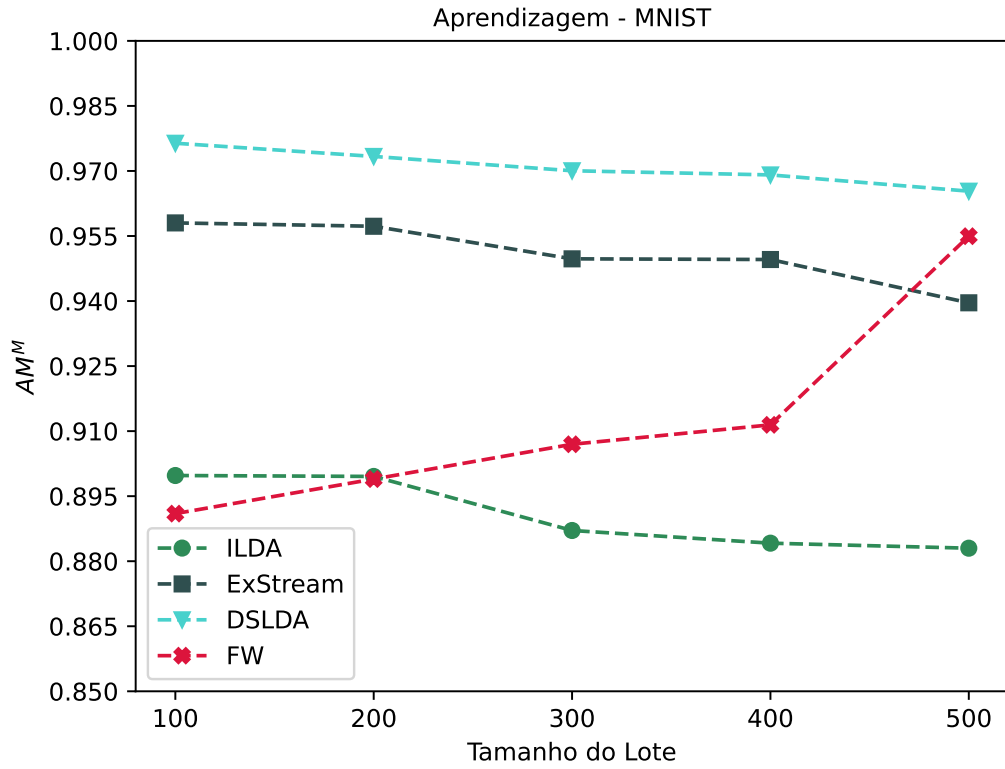
Em $Q_t^{AP} = 100$, FW obtém $AG^M = 0.883$. Em $Q_t^{AP} = 500$, FW melhora e atinge $AG^M = 0.920$. Mesmo assim, DSLDA e ExStream conseguiram resultados melhores para a classificação multiclases. Em $Q_t^{AP} = 100$, DSLDA obtém $AG^M = 0.994$, sendo o melhor resultado para a avaliação da métrica AG^M . Em $Q_t^{AP} = 500$, os resultados de FW equivalem aos de ExStream, que obtém $AG^M = 0.926$. Por fim, ILDA apresenta queda constante nos resultados quando $Q_t^{AP} \geq 300$. Em $Q_t^{AP} = 500$, ILDA obtém o pior desempenho, com $AG^M = 0.880$.

A Figura 25 ilustra o desempenho preditivo dos métodos comparados na base de imagens MNIST. Os resultados são semelhantes aos encontrados na base CIFAR-10. O método DSLDA apresenta os melhores resultados para todos os tamanhos de lote de imagens. O *framework* proposto ocupa o segundo lugar, ficando à frente do método ExStream. E novamente o método ILDA apresenta as piores quedas de desempenho.

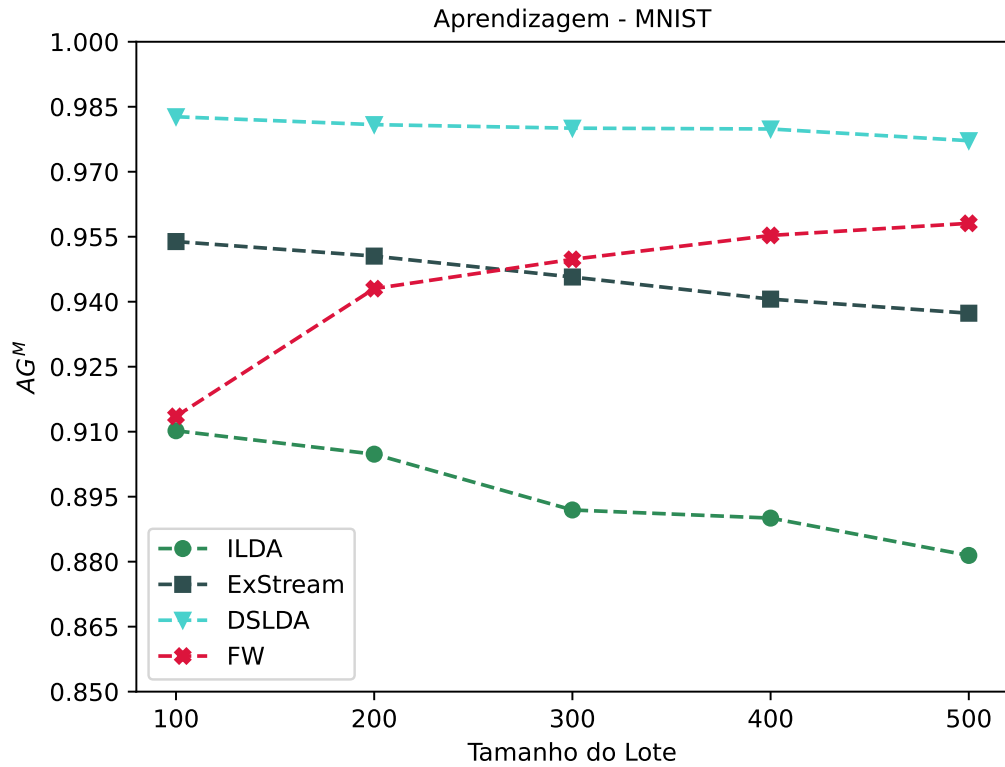
A acurácia média observada na Figura 25a mostra que o método DSLDA obtém $AM^M = 0.976$ em $Q_t^{AP} = 100$, sendo o melhor resultado deste trabalho para base completa de MNIST. Tanto DSLDA quanto ExStream permanecem estáveis à medida que o tamanho do lote aumenta, enquanto FW melhora os resultados. Em $Q_t^{AP} = 100$, FW alcança o pior resultado, obtendo $AM^M = 0.891$. Porém, em $Q_t^{AP} = 500$, alcança o segundo melhor resultado, obtendo $AM^M = 0.955$. ILDA pior à medida que Q_t^{AP} aumenta, e obtém $AM^M = 0.883$ em $Q_t^{AP} = 500$.

Na Figura 25b, a acurácia geral dos métodos é comparada para a base completa de MNIST. Os resultados são bastante semelhantes aos anteriores. Porém, o desempenho de FW cresce mais rapidamente quando as classes são comparadas isoladamente. Em $Q_t^{AP} = 100$, obtém $AG^M = 0.914$. Em $Q_t^{AP} = 500$, cresce até 5% e alcança $AG^M = 0.958$.

Figura 25: Desempenho preditivo do *framework* usando modelos de seleção de atributos sobre a base de dados MNIST: (a) AM^M e (b) AG^M , conforme o número de atributos selecionados.



(a)



(b)

ExStream compete com FW em $Q_t^{AP} = \{200, 300\}$, obtendo resultados melhores que a proposta para valores menores de Q_t^{AP} .

Os resultados analisados para as bases completas de CIFAR-10 e MNIST mostraram que o *framework* proposto é eficiente na predição para a tarefa de classificação multiclases. A proposta não possui um mecanismo de adaptação de conceitos, como os demais métodos. Mesmo assim, ela alcançou resultados equivalentes e até superiores em cenários com múltiplas classes. Esses resultados mostram também que o mecanismo de aprendizagem incremental e *online* adotada é eficaz na redução do esquecimento catastrófico da rede convolucional.

Discussão: o *framework* proposto apresenta bons resultados para a classificação multiclases, mas tem vantagens sobre os demais métodos na classificação binária. A proposta apresenta instabilidade na variação do tamanho dos lotes para uma análise geral dos resultados, mas é o método que melhor se adapta a grandes quantidades de imagens selecionadas por lote. Em cenários não simulados, a proposta seria mais facilmente adaptável a ambientes de alta velocidade, embora ainda não lide com mudança de conceitos.

7.5 Considerações Finais

Este capítulo apresentou os resultados experimentais que validam o *framework* proposto nesta tese. Dois grupos de experimentos foram construídos para avaliar o poder preditivo da proposta em classificar novas imagens em fluxos de dados. Os experimentos foram avaliados a partir de duas métricas de acurácia adaptadas para a proposta em três base de dados de imagens volumosas distintas. Duas tarefas de classificação avaliam os modelos: classificação binária e classificação multiclases.

O primeiro grupo de experimentos (*SA*) avaliou a capacidade do mecanismo de seleção de atributos em extrair as características mais relevantes das imagens. A capacidade preditiva do *framework* é analisada a partir do volume de características selecionadas (Q^{AP}). Os resultados são comparados com três modelos de seleção de atributos: **DFS** (LI; CHEN; WASSERMAN, 2016), **OFS** (WANG et al., 2014) e **FSDS** (HUANG; YOO; KASIVISWANATHAN, 2015).

Para realizar comparações com equidade, o mecanismo de seleção de atributos foi substituído pelos modelos. Dessa forma, os resultados são comparados entre o mecanismo original do *framework* (**FW**) e os modelos que utilizam o *framework* (**FW+**). A qualidade dos atributos é avaliada por meio do poder preditivo do *framework* em realizar predições

nas bases de imagens. As métricas da Acurácia Média (AM) e da Acurácia Geral (AG) são computadas e reportadas como taxas de acerto.

Para a tarefa de classificação binária, FW supera os demais métodos para a maior parte das comparações. Em lotes de imagens com maior quantidade de atributos selecionados ($Q^{SA} \geq 300$), FW alcança até 86,3% de acertos na acurácia média (AM^B), e 85,9% de acertos na acurácia geral (AG^B).

O mecanismo proposto compete diretamente com o modelo DFS, que também apresenta bons resultados. Embora DFS utilize Aprendizado Profundo para a seleção dos atributos, a estratégia do método é bastante semelhante a um método comum de Regressão Linear. O *framework* proposto consegue ser mais eficiente que DFS, pois filtra antecipadamente as características usando uma regressão Elastic Net e transmite como entrada para uma Rede Convolutacional efetuar a aprendizagem incrementalmente.

Na tarefa de classificação multiclases, os resultados para FW crescem/ mais rapidamente em cenários com maior quantidade de atributos ($Q^{SA} = \{400, 500\}$). Os demais métodos apresentam desempenho pior e em constante queda. Para o bloco de experimentos dessa tarefa, o mecanismo de aprendizagem *online* se destaca pela eficiência na seleção das características das imagens presentes em lotes com diferentes classes.

O segundo grupo de experimentos (AP) avaliou a capacidade preditiva do *framework*, que é analisada a partir da aptidão dos mecanismos de aprendizagem e classificação em, respectivamente, aprender e prever lotes de imagens. Os resultados obtidos pelo *framework* proposto (**FW**) são comparados com três métodos de Aprendizado em Fluxos de Dados: **ILDA** (PANG; OZAWA; KASABOV, 2005; CHU et al., 2015), **ExStream** (HAYES; CAHILL; KANAN, 2019) e **DSLDA** (HAYES; KANAN, 2020). Adicionalmente aos resultados preditivos, os mecanismos também são discutidos a partir da capacidade em mitigar o fenômeno do esquecimento catastrófico.

Na classificação binária, FW compete principalmente com DSLDA e ExStream, mostrando-se mais estável para diferentes tamanhos de lotes de imagens. Em lotes maiores ($Q^{AP} = \{400, 500\}$), FW supera os demais métodos. FW alcança até 95,3% de acertos na acurácia média (AM^M), e 96,6% de acertos na acurácia geral (AG^M). Todos os resultados obtidos na classificação binária superam 90% de acertos, sendo os melhores resultados para lotes maiores de imagens.

Os experimentos desse bloco mostram que o uso da técnica de Janelamento Amor-tecido (YUN et al., 2018) foi eficiente para a seleção das imagens mais recentes para a formação do lote, permitindo maiores acertos pela RNA. Essa escolha também permitiu

que o *framework* como um todo suavizasse os efeitos do esquecimento catastrófico da RNA no cenário *online*.

Para a tarefa de classificação multiclases, FW apresenta instabilidade para o desempenho na variação do tamanho dos lotes de imagens. Essa instabilidade é causada pelo esquecimento catastrófico da RNA no mecanismo de aprendizagem. A proposta compete principalmente com ExStream, mas não com DSLDA, que apresenta melhores resultados na maior parte dos cenários. FW alcança até 95,5% de acertos na acurácia média (AM^M), e 95,8% de acertos na acurácia geral (AG^M).

Considerando a complexidade da tarefa de classificação multiclases, FW apresenta uma proposta simples para filtrar os atributos mais relevantes que serão utilizados pela RNA. Os resultados são próximos aos obtidos pelos demais métodos. Embora a regressão Elastic Net tenha se mostrado bastante eficiente na classificação binária, a RNA do método DSLDA conseguiu capturar melhor a complexidade das classes observadas.

Os resultados apresentados e discutidos neste capítulo cumprem com os objetivos da Seção 1.4 para este tese, incluindo: simulação de um fluxos de imagens para seleção de atributos; proposição de um *framework* para a construção e validação de modelos de redes neurais profundas aplicados à classificação de imagens em fluxos de dados; e comparação e avaliação de métodos de classificação de imagens em fluxos de dados.

8 CONCLUSÃO

Este capítulo conclui os trabalhos realizados nesta tese, que implementa algoritmos para a classificação de imagens utilizando Aprendizado Profundo. O trabalho concentra-se na investigação de modelos e algoritmos para o processamento e a aprendizagem de imagens em fluxos de dados.

Inicialmente, o trabalho investiga a literatura recente de Aprendizado Profundo em fluxos de dados. Essa investigação é conduzida a partir das questões de pesquisa definidas na Seção 1.5, que tem o propósito de identificar técnicas de processamento de imagens, modelos de seleção de atributos e algoritmos para a aprendizagem e classificação de imagens.

Em seguida, o trabalho propõe um *framework*, que consiste em um método para a construção e validação de modelos de redes neurais profundas. O método é baseado no processo de classificação de imagens. A ideia básica é aprender em fluxos contínuos de dados, observando as características intrínsecas dos fluxos e seus requisitos de processamento computacional.

A formulação do *framework* envolve primeiramente a construção de um simulador de fluxos de dados para processar as imagens. Essa é a primeira etapa do *framework*, que recebe as imagens como entrada no fluxo contínuo de dados. As imagens são tratadas como itens de dados no fluxo e processadas para serem utilizadas no processo de aprendizagem. Esse processo acontece de maneira *online*.

As etapas subsequentes extraem as características das imagens e as utilizam nos processos de aprendizagem e classificação. No processo de extração, os atributos mais relevantes das imagens são selecionados por um método de Regressão Elastic Net, que mantém o tratamento *online* dos dados em processamento.

Na etapa de aprendizagem, os dados dos lotes de imagens são escolhidos para a construção de um modelo de decisão incremental e constantemente atualizado. Esse modelo é formado por uma Rede Neural Convolutiva capaz de aprender as características visuais

das imagens. O conjunto de mecanismos do *framework* permite que a janela de imagens se mantenha atualizada sem o descarte de conceitos antigos já processados. Dessa forma, o *framework* evita o esquecimento catastrófico.

Por fim, um conjunto de experimentos são planejados para avaliar o desempenho dos modelos construídos pelo *framework*. As etapas da abordagem experimental adotada são detalhadas na Seção 6.1, que descreve o processo de avaliação do método em dez etapas sequenciais. O *framework* é comparado com outros modelos de seleção de atributos e métodos de aprendizagem que utilizam Aprendizado Profundo.

Os experimentos são conduzidos sobre três diferentes bases de dados de imagens volumosas em duas tarefas de classificação de imagens. Cada tarefa ainda é avaliada a partir do mecanismo de seleção de atributos ou aprendizagem e classificação do *framework*. Duas métricas de acurácia são adaptadas e computadas para cada grupo de experimentos em cada tarefa.

Portanto, os experimentos conduzidos mostram como o *framework* proposto confirma a hipótese desta tese. A combinação de redes neurais profundas e algoritmos para a aprendizagem em fluxos de dados permite a classificação de imagens em fluxos de dados.

As seções seguintes detalham os principais resultados obtidos com o *framework* proposto, as contribuições da tese, as limitações do trabalho realizado e os trabalhos futuros.

8.1 Principais Resultados

O principal resultado desta tese é a implementação de algoritmos para a classificação de imagens em fluxos de dados utilizando Aprendizado Profundo. O conjunto das implementações resulta na proposição de um *framework* completo e versátil para a construção e validação de modelos de redes neurais profundas. O *framework* também oferece um conjunto abrangente de ferramentas para a otimização desses modelos. O uso e as aplicações derivadas do *framework* são direcionadas à tarefa de classificação de imagens em ambientes e contextos baseados em fluxos de dados.

Os experimentos conduzidos com o *framework* proposto oferecem respostas sólidas e conclusivas às questões centrais elaboradas para esta pesquisa, consolidando, assim, a validade da hipótese subjacente a esta tese. Os resultados obtidos afirmam que o *framework* não apenas atende às expectativas aos objetivos definidos, mas também supera as metas estabelecidas, destacando seu potencial como uma ferramenta fundamental para

avançar no entendimento e aplicação de Aprendizado Profundo em ambientes de FD dinâmicos. Em suma, as questões são respondidas desta forma:

- **Q1: Como extrair atributos relevantes de imagens em fluxos de dados em tempo de processamento constante?**

A revisão de literatura e o desenvolvimento do mecanismo de seleção de atributos do *framework* proposto identificam métodos lineares como o conjunto de técnicas responsáveis pela extração das características mais relevantes de uma imagem que trafega em um fluxo contínuo de dados.

Nesta tese, o método linear adotado foi a técnica de regressão linear Elastic Net, uma estratégia de regularização comumente adotada para seleção de variáveis em modelos estatísticos. O mecanismo de seleção de atributos implementa essa estratégia. A abordagem é combinada com um modelo de RNA para habilitar a Aprendizagem Incremental. O *framework* utiliza lotes de imagens para facilitar a entrada dos dados de imagem na RNA.

- **Q2: Como construir modelos de classificação de imagens em fluxos de dados?**

A construção de modelos de classificação é uma tarefa complexa, especialmente em ambientes dinâmicos e de alta velocidade. O desenvolvimento de modelos de aprendizagem requer a implementação de mecanismos que habilitem a aprendizagem e a predição ao longo do tempo. Dessa forma, a construção do modelo de classificação se torna uma tarefa mais eficaz e versátil, permitindo aprimorar a capacidade de generalização e a retenção de conhecimento relevante.

Nesta tese, a construção de modelos de classificação implementa mecanismos *online* e *offline* para a aprendizagem e predição de imagens. No *framework*, esses mecanismos são mesclados na proposta com outras abordagens para não agravar o esquecimento catastrófico na RNA. Dessa forma, todo o processo trabalha de maneira *online*, mas apenas no mecanismo de aprendizagem a configuração incremental é habilitada.

- **Q3: Como avaliar modelos de classificação de imagens em fluxos de dados?**

A revisão de literatura identificou que as estratégias de avaliação enfatizam a tarefa e o domínio da aplicação. Embora o erro frequencial e outras métricas sejam tradicionalmente adotadas para o Aprendizado em Fluxos de Dados, a tarefa da classificação de imagens é tradicionalmente avaliada por métricas da Visão Computacional.

Assim, para avaliar o *framework* proposto e permitir a comparação com outros métodos, esta tese adota a métrica da acurácia. Uma vez que as bases de imagens

diversas selecionadas para os experimentos possuem propósitos diferentes, variações da acurácia tornam-se mais úteis para a validação de modelos preditivos de propósito geral.

Nesta tese, a Acurácia Média e a Acurácia Geral são adaptadas para as tarefas de classificação binária e multiclases. Essas versões são ajustadas para trabalhar mais facilmente com a configuração incremental do mecanismo de aprendizagem e permitir a seleção *online* do lote de imagens para a predição.

- **Q4: É possível utilizar modelos de Aprendizado Profundo para classificar imagens em fluxos de dados?**

Sim, é possível utilizar modelos de Aprendizado Profundo na tarefa de classificação de imagens que trafegam em fluxos de dados. A revisão de literatura mais recente apresenta diversas aplicações que enfatizam o uso de modelos neurais, sobretudo de Aprendizado Profundo. No entanto, essa adoção ainda enfrenta desafios.

Nesta tese, um *framework* é proposto para amenizar esses desafios. Parte da extração de atributos comumente executada por uma CNN é delegada para um mecanismo específico de seleção de dados. O processamento em tempo constante das imagens é resolvido dessa forma.

Adicionalmente, o processo do *framework* ignora as imagens excedentes antes que elas sejam apresentadas à RNA, amenizando os efeitos do esquecimento catastrófico. O mecanismo de aprendizagem retém apenas as imagens que formulam novos conceitos ou modificam conceitos existentes. Assim, quando o modelo é atualizado, ele não esquece o que foi aprendido.

Portanto, a hipótese desta tese é aceita a partir dos experimentos que validam o *framework* proposto. A comparação dos resultados preditivos com outros métodos mostram a eficácia da proposta em combinar redes neurais profundas e algoritmos de aprendizagem em fluxos de dados para classificar imagens. O método concilia as abordagens incremental e *online* para habilitar a aprendizagem de imagens em fluxos de dados.

8.2 Contribuições

A principal contribuição desta tese é a integração entre os diferentes mecanismos do *framework* proposto, que trabalham em modo *online* e estão habilitados para a configuração incremental. Dessa forma, o mecanismo de aprendizagem é capaz de incorporar novos

conceitos ou classes de objetos sem a necessidade de treinar novamente por completo a RNP, economizando tempo e recursos computacionais.

O *framework* consiste em um método computacional baseado no processo de classificação de imagens que observa e restringe-se às características intrínsecas dos FDs e seus requisitos de processamento computacional. A capacidade do *framework* em processar imagens em tempo real garante que a análise seja realizada sem atrasos significativos. Isso torna a proposta adequada a aplicações que exigem resposta imediata.

Ainda, o *framework* é aplicável às tarefas de classificação binária e classificação multiclases de imagens, com ambas podendo ser executadas em FDs. Por ser formado por mecanismos que se conectam, o *framework* pode ainda ser adaptado a diferentes aplicações reais e simuladores de fluxos de dados. Neste trabalho, as aplicações são direcionadas à classificação de imagens digitais.

Este trabalho ainda revisa os procedimentos para a extração e seleção de atributos de imagens em fluxos de dados. A revisão desses procedimentos é relatada com foco na aprendizagem. Assim, ela não apenas descreve os métodos tradicionais, mas também discute sobre novas técnicas e estratégias que têm o potencial de impulsionar significativamente o desempenho de sistemas de processamento de imagens em fluxos de dados evolutivos.

Duas revisões com foco na tarefa de classificação também são conduzidas e contribuem para o entendimento de novos métodos em fluxos de dados. A primeira delas compreende o uso de imagens em classificadores baseados em fluxos de dados e as metodologias comumente adotadas para informações visuais. A segunda revisão relata o uso de algoritmos de Aprendizado Profundo em fluxos de dados. Ambas as revisões destacam a importância da adaptabilidade dos procedimentos estudados em ambientes dinâmicos.

Por fim, no âmbito experimental, este trabalho discute o uso de modelos de redes neurais profundas em fluxos de dados. Diferentes métodos de classificação de imagens baseados em redes neurais profundas são analisados e discutidos em termos de capacidade preditiva e eficiência computacional. Experimentos abrangentes utilizando base de imagens representativas avaliam o desempenho em um *framework* completo e adaptável de ponta a ponta.

8.3 Limitações

Esta tese trabalhou com as características intrínsecas dos fluxos de dados e seus requisitos computacionais. O *framework* desenvolvido observa apenas a sequencialidade das

imagens que chegam ao fluxo em tempo real, sendo também capaz de processá-las em tempo constante e de forma limitada. O modelo de aprendizado pode ser fornecido a qualquer momento e de maneira atualizada.

Conforme os experimentos conduzidos, o *framework* está habilitado para classificar imagens em fluxos de dados. As tarefas de classificação binária e multiclases foram experimentadas e validadas por meio de diferentes experimentos em bases de imagens distintas e volumosas. Diferentes segmentos de mercado da TD podem ser beneficiados com a proposta.

8.4 Trabalhos Futuros

O primeiro trabalho futuro que deve ser realizado a partir desta tese é a melhoria nos mecanismos centrais do *framework*. A comparação de resultados com outros métodos mostra que ajustes podem ser realizados para aumentar a acurácia preditiva da proposta, especialmente em cenários multiclases.

Para isso, duas adaptações podem ser realizadas. A primeira delas é no mecanismo de seleção de atributos. A simplicidade do método de regressão deixa de capturar características mais complexas das imagens, embora em tempo linear. Máquinas de Vetores de Suporte (SVM) são a sugestão principal para essa adaptação, substituindo a regressão Elastic Net (ZHOU et al., 2015).

SVMs são mais complexas e devem capturar melhor as características das imagens. No entanto, para se adequar ao *framework*, o algoritmo deve ser implementado na configuração incremental (XIAO; WANG; ZHANG, 2000). Essa implementação abriria caminhos também para novos experimentos em Unidades Gráficas de Processamento (GPU).

A segunda adaptação é no mecanismo de aprendizagem. Apesar da proposta implementar uma CNN com poucas camadas, a principal sugestão é no uso de arquiteturas consolidadas, como ResNet (HE et al., 2016). O uso de blocos residuais no *framework* deve permitir que os lotes de imagem sejam treinados com mais velocidade, permitindo o aumento da acurácia em cenários multiclases.

Entre as novidades, o *framework* pode aderir a outros desafios do AFD, como a classificação multirrótulos, a detecção de mudança de conceitos e o desbalanceamento de classes. Esses desafios também são tradicionalmente estudados em AM. Em CV, experimentos com imagens em altas dimensões mostrariam a capacidade preditiva da proposta em ambientes ainda mais complexos do que aqueles que foram abordados nesta tese.

REFERÊNCIAS

- AGGARWAL, C. C.; HAN, J.; WANG, J.; YU, P. S. On demand classification of data streams. In: **Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2004. p. 503–508.
- AGGARWAL, C. C.; PHILIP, S. Y.; HAN, J.; WANG, J. A framework for clustering evolving data streams. In: ELSEVIER. **Proceedings 2003 VLDB conference**. [S.l.], 2003. p. 81–92.
- AI, Y.; HU, W.; LI, Y.; WOODRUFF, D. P. New characterizations in turnstile streams with applications. In: SCHLOSS DAGSTUHL-LEIBNIZ-ZENTRUM FUER INFORMATIK. **31st Conference on Computational Complexity (CCC 2016)**. [S.l.], 2016.
- ASHFAHANI, A. Autonomous deep learning: incremental learning of deep neural networks for evolving data streams. In: IEEE. **2019 International Conference on Data Mining Workshops (ICDMW)**. [S.l.], 2019. p. 83–90.
- BAHRI, M.; BIFET, A.; GAMA, J.; GOMES, H. M.; MANIU, S. Data stream analysis: Foundations, major tasks and tools. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 11, n. 3, p. e1405, 2021.
- BESEDIN, A.; BLANCHART, P.; CRUCIANU, M.; FERECATU, M. Evolutive deep models for online learning on data streams with no storage. In: **ECML/PKDD 2017 Workshop on Large-scale Learning from Data Streams in Evolving Environments**. [S.l.: s.n.], 2017.
- BIFET, A.; GAVALDA, R. Learning from time-changing data with adaptive windowing. In: SIAM. **Proceedings of the 2007 SIAM international conference on data mining**. [S.l.], 2007. p. 443–448.
- BIFET, A.; HOLMES, G.; PFAHRINGER, B.; GAVALDA, R. Improving adaptive bagging methods for evolving data streams. In: SPRINGER. **Advances in Machine Learning: First Asian Conference on Machine Learning, ACML 2009, Nanjing, China, November 2-4, 2009. Proceedings 1**. [S.l.], 2009. p. 23–37.
- BIFET, A.; HOLMES, G.; PFAHRINGER, B.; FRANK, E. Fast perceptron decision tree learning from evolving data streams. In: SPRINGER. **Advances in Knowledge Discovery and Data Mining: 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part II 14**. [S.l.], 2010. p. 299–310.
- BRZEZINSKI, D.; STEFANOWSKI, J. Prequential auc for classifier evaluation and drift detection in evolving data streams. In: SPRINGER. **New Frontiers in Mining Complex Patterns: Third International Workshop, NFMCP 2014, Held in Conjunction with ECML-PKDD 2014, Nancy, France, September 19, 2014, Revised Selected Papers 3**. [S.l.], 2015. p. 87–101.

- CALANDRA, R.; RAIKO, T.; DEISENROTH, M. P.; POUZOLS, F. M. Learning deep belief networks from non-stationary streams. In: SPRINGER. **Artificial Neural Networks and Machine Learning–ICANN 2012: 22nd International Conference on Artificial Neural Networks, Lausanne, Switzerland, September 11-14, 2012, Proceedings, Part II 22**. [S.l.], 2012. p. 379–386.
- CASTRO, F. M.; MARÍN-JIMÉNEZ, M. J.; GUIL, N.; SCHMID, C.; ALAHARI, K. End-to-end incremental learning. In: **Proceedings of the European conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 233–248.
- CHU, D.; LIAO, L.-Z.; NG, M. K.-P.; WANG, X. Incremental linear discriminant analysis: A fast algorithm and comparisons. **IEEE transactions on neural networks and learning systems**, IEEE, v. 26, n. 11, p. 2716–2735, 2015.
- DATAR, M.; GIONIS, A.; INDYK, P.; MOTWANI, R. Maintaining stream statistics over sliding windows. **SIAM journal on computing**, SIAM, v. 31, n. 6, p. 1794–1813, 2002.
- DAVIES, E. R. **Computer vision: principles, algorithms, applications, learning**. [S.l.]: Academic Press, 2017.
- DOMINGOS, P.; HULTEN, G. Mining high-speed data streams. In: **Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2000. p. 71–80.
- DOMINGOS, P. M.; HULTEN, G. Catching up with the data: Research issues in mining data streams. **DMKD**, 2001.
- DUDA, P.; JAWORSKI, M.; CADER, A.; WANG, L. On training deep neural networks using a streaming approach. **Journal of Artificial Intelligence and Soft Computing Research**, v. 10, 2020.
- DUDA, P.; WANG, L. On a streaming approach for training denoising auto-encoders. In: SPRINGER. **International Conference on Artificial Intelligence and Soft Computing**. [S.l.], 2020. p. 315–324.
- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997.
- GAMA, J.; SEBASTIAO, R.; RODRIGUES, P. P. On evaluating stream learning algorithms. **Machine learning**, Springer, v. 90, p. 317–346, 2013.
- GEBRESLASSIE, A. K.; BENOIS-PINEAU, J.; ZEMMARI, A. Streaming learning with move-to-data approach for image classification. In: **Proceedings of the 19th International Conference on Content-based Multimedia Indexing**. [S.l.: s.n.], 2022. p. 167–173.
- GILBERT, A. C.; KOTIDIS, Y.; MUTHUKRISHNAN, S.; STRAUSS, M. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In: **Vldb**. [S.l.: s.n.], 2001. v. 1, p. 79–88.

- GOMES, H. M.; READ, J.; BIFET, A.; BARDDAL, J. P.; GAMA, J. Machine learning for streaming data: state of the art, challenges, and opportunities. **ACM SIGKDD Explorations Newsletter**, ACM New York, NY, USA, v. 21, n. 2, p. 6–22, 2019.
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. [S.l.]: Pearson, 2018.
- GRZENDA, M.; GOMES, H. M.; BIFET, A. Delayed labelling evaluation for data streams. **Data Mining and Knowledge Discovery**, Springer, v. 34, n. 5, p. 1237–1266, 2020.
- HAYES, T. L.; CAHILL, N. D.; KANAN, C. Memory efficient experience replay for streaming learning. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 9769–9776.
- HAYES, T. L.; KAFLE, K.; SHRESTHA, R.; ACHARYA, M.; KANAN, C. Remind your neural network to prevent catastrophic forgetting. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2020. p. 466–483.
- HAYES, T. L.; KANAN, C. Lifelong machine learning with deep streaming linear discriminant analysis. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2020. p. 220–221.
- HE, J.; MAO, R.; SHAO, Z.; ZHU, F. Incremental learning in online scenario. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 13926–13935.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.
- HIDALGO, J. I. G.; MACIEL, B. I.; BARROS, R. S. Experimenting with prequential variations for data stream learning evaluation. **Computational Intelligence**, Wiley Online Library, v. 35, n. 4, p. 670–692, 2019.
- HU, W.; ZHANG, B. Study of sampling techniques and algorithms in data stream environments. In: IEEE. **2012 9th International Conference on Fuzzy Systems and Knowledge Discovery**. [S.l.], 2012. p. 1028–1034.
- HUANG, H.; YOO, S.; KASIVISWANATHAN, S. P. Unsupervised feature selection on data streams. In: **Proceedings of the 24th ACM International on Conference on Information and Knowledge Management**. [S.l.: s.n.], 2015. p. 1031–1040.
- JAMEEL, S. M.; HASHMANI, M. A.; REHMAN, M.; BUDIMAN, A. An adaptive deep learning framework for dynamic image classification in the internet of things environment. **Sensors**, MDPI, v. 20, n. 20, p. 5811, 2020.
- JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R.; TAYLOR, J. **An Introduction to Statistical Learning with Applications in Python**. [S.l.]: Springer, 2023.
- JAWORSKI, M.; RUTKOWSKI, L.; ANGELOV, P. Concept drift detection using autoencoders in data streams processing. In: SPRINGER. **International Conference on Artificial Intelligence and Soft Computing**. [S.l.], 2020. p. 124–133.

- KRIZHEVSKY, A.; HINTON, G. et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009.
- LECUN, Y.; CORTES, C.; BURGESS, C. et al. **MNIST handwritten digit database**. [S.l.]: Florham Park, NJ, USA, 2010.
- LI, X.; DING, M.; PIŽURICA, A. Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 58, n. 4, p. 2615–2629, 2019.
- LI, Y.; CHEN, C.-Y.; WASSERMAN, W. W. Deep feature selection: theory and application to identify enhancers and promoters. **Journal of Computational Biology**, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 23, n. 5, p. 322–336, 2016.
- LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid search, random search, genetic algorithm: a big comparison for nas. **arXiv preprint arXiv:1912.06059**, 2019.
- LIBERTY, E. Simple and deterministic matrix sketching. In: **Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.: s.n.], 2013. p. 581–588.
- MAI, Z.; LI, R.; JEONG, J.; QUISPE, D.; KIM, H.; SANNER, S. Online continual learning in image classification: An empirical survey. **Neurocomputing**, Elsevier, v. 469, p. 28–51, 2022.
- MASANA, M.; LIU, X.; TWARDOWSKI, B.; MENTA, M.; BAGDANOV, A. D.; WEIJER, J. V. D. Class-incremental learning: survey and performance evaluation on image classification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 45, n. 5, p. 5513–5533, 2022.
- MOHAMMADI, M.; AL-FUQAHA, A.; SOROUR, S.; GUIZANI, M. Deep learning for iot big data and streaming analytics: A survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 20, n. 4, p. 2923–2960, 2018.
- MUTHUKRISHNAN, S. et al. Data streams: Algorithms and applications. **Foundations and Trends® in Theoretical Computer Science**, Now Publishers, Inc., v. 1, n. 2, p. 117–236, 2005.
- NAJAFABADI, M. M.; VILLANUSTRE, F.; KHOSHGOFTAAR, T. M.; SELIYA, N.; WALD, R.; MUHAREMAGIC, E. Deep learning applications and challenges in big data analytics. **Journal of big data**, SpringerOpen, v. 2, n. 1, p. 1–21, 2015.
- NGUYEN, H.-L.; WOON, Y.-K.; NG, W.-K. A survey on data stream clustering and classification. **Knowledge and information systems**, Springer, v. 45, p. 535–569, 2015.
- NGUYEN, T. D.; SHIH, M.-H.; SRIVASTAVA, D.; TIRTHAPURA, S.; XU, B. Stratified random sampling from streaming and stored data. **Distributed and Parallel Databases**, Springer, v. 39, p. 665–710, 2021.
- OTTAVIANI, E.; FRANCESCANGELI, M.; GJECI, N.; FERNANDEZ, J. del R.; AGUZZI, J.; MARINI, S. Assessing the image concept drift at the obsea coastal underwater cabled observatory. **Frontiers in Marine Science**, Frontiers, v. 9, p. 840088, 2022.

- O'MAHONY, N.; CAMPBELL, S.; CARVALHO, A.; HARAPANAHALLI, S.; HERNANDEZ, G. V.; KRPALKOVA, L.; RIORDAN, D.; WALSH, J. Deep learning vs. traditional computer vision. In: SPRINGER. **Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1**. [S.l.], 2020. p. 128–144.
- PANG, S.; OZAWA, S.; KASABOV, N. Incremental linear discriminant analysis for classification of data streams. **IEEE transactions on Systems, Man, and Cybernetics, part B (Cybernetics)**, IEEE, v. 35, n. 5, p. 905–914, 2005.
- POURSANIDIS, M.; BENOIS-PINEAU, J.; ZEMMARI, A.; MANSENCA, B.; RUGY, A. de. Move-to-data: A new continual learning approach with deep cnns, application for image-class recognition. **arXiv preprint arXiv:2006.07152**, 2020.
- 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**. [S.l.: s.n.], 2017. p. 2001–2010.
- ROADY, R.; HAYES, T. L.; VAIDYA, H.; KANAN, C. Stream-51: Streaming classification and novelty detection from videos. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2020. p. 228–229.
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M. et al. Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, p. 211–252, 2015.
- SAHOO, D.; PHAM, Q.; LU, J.; HOI, S. C. Online deep learning: learning deep neural networks on the fly. In: **Proceedings of the 27th International Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2018. p. 2660–2666.
- SILVA JUNIOR, O. d.; LORENA, A. C. **Introdução à Classificação de Fluxos de Dados**. Santo André, Brasil, 2014.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SINGLA, A.; YUAN, L.; EBRAHIMI, T. Food/non-food image classification and food categorization using pre-trained googlenet model. In: **Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management**. [S.l.: s.n.], 2016. p. 3–11.
- SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.
- TEKIN, C.; SCHAAR, M. van der. Active learning in context-driven stream mining with an application to image mining. **IEEE Transactions on Image Processing**, IEEE, v. 24, n. 11, p. 3666–3679, 2015.

VITTER, J. S. Random sampling with a reservoir. **ACM Transactions on Mathematical Software (TOMS)**, ACM New York, NY, USA, v. 11, n. 1, p. 37–57, 1985.

WANG, J.; ZHAO, P.; HOI, S. C.; JIN, R. Online feature selection and its applications. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 26, n. 3, p. 698–710, 2014.

WANG, Z.; OATES, T. et al. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In: **AAAI MENLO PARK, CA, USA. Workshops at the twenty-ninth AAAI conference on artificial intelligence**. [S.l.], 2015. v. 1.

WOLFE, C. R.; KYRILLIDIS, A. Cold start streaming learning for deep networks. **arXiv preprint arXiv:2211.04624**, 2022.

XIAO, R.; WANG, J.; ZHANG, F. An approach to incremental svm learning algorithm. In: **IEEE. Proceedings 12th IEEE Internationals Conference on Tools with Artificial Intelligence. ICTAI 2000**. [S.l.], 2000. p. 268–273.

YU, K.; WU, X.; DING, W.; PEI, J. Scalable and accurate online feature selection for big data. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, ACM New York, NY, USA, v. 11, n. 2, p. 1–39, 2016.

YUN, U.; KIM, D.; YOON, E.; FUJITA, H. Damped window based high average utility pattern mining over data streams. **Knowledge-Based Systems**, Elsevier, v. 144, p. 188–205, 2018.

ZHOU, P.; WANG, N.; ZHAO, S. Online group streaming feature selection considering feature interaction. **Knowledge-Based Systems**, Elsevier, v. 226, p. 107157, 2021.

ZHOU, Q.; CHEN, W.; SONG, S.; GARDNER, J.; WEINBERGER, K.; CHEN, Y. A reduction of the elastic net to support vector machines with an application to gpu computing. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2015. v. 29, n. 1.

APÊNDICE A – RESULTADOS DOS EXPERIMENTOS

Tabela 7: Desempenho preditivo dos métodos para a seleção de atributos na classificação binária na base de dados Food-5k.

Experimento	Método	Atributos	AM^B	AG^B
<i>SA</i> – 1	DFS+FW	100	0.859	0.880
<i>SA</i> – 2	OFS+FW	100	0.754	0.755
<i>SA</i> – 3	FSDS+FW	100	0.709	0.719
<i>SA</i> – 4	FW	100	0.791	0.816
<i>SA</i> – 21	DFS+FW	200	0.857	0.867
<i>SA</i> – 22	OFS+FW	200	0.749	0.747
<i>SA</i> – 23	FSDS+FW	200	0.694	0.719
<i>SA</i> – 24	FW	200	0.817	0.816
<i>SA</i> – 41	DFS+FW	300	0.810	0.800
<i>SA</i> – 42	OFS+FW	300	0.744	0.729
<i>SA</i> – 43	FSDS+FW	300	0.693	0.703
<i>SA</i> – 44	FW	300	0.834	0.828
<i>SA</i> – 61	DFS+FW	400	0.803	0.798
<i>SA</i> – 62	OFS+FW	400	0.731	0.725
<i>SA</i> – 63	FSDS+FW	400	0.685	0.684
<i>SA</i> – 64	FW	400	0.845	0.848
<i>SA</i> – 81	DFS+FW	500	0.799	0.785
<i>SA</i> – 82	OFS+FW	500	0.715	0.723
<i>SA</i> – 83	FSDS+FW	500	0.677	0.682
<i>SA</i> – 84	FW	500	0.845	0.859

Tabela 8: Desempenho preditivo dos métodos para a seleção de atributos na classificação binária na base de dados CIFAR-10.

Experimento	Método	Atributos	AM^B	AG^B
<i>SA</i> – 5	DFS+FW	100	0.849	0.848
<i>SA</i> – 6	OFS+FW	100	0.738	0.745
<i>SA</i> – 7	FSDS+FW	100	0.672	0.686
<i>SA</i> – 8	FW	100	0.825	0.856
<i>SA</i> – 25	DFS+FW	200	0.869	0.870
<i>SA</i> – 26	OFS+FW	200	0.759	0.767
<i>SA</i> – 27	FSDS+FW	200	0.695	0.707
<i>SA</i> – 28	FW	200	0.795	0.810
<i>SA</i> – 45	DFS+FW	300	0.856	0.869
<i>SA</i> – 46	OFS+FW	300	0.749	0.753
<i>SA</i> – 47	FSDS+FW	300	0.683	0.689
<i>SA</i> – 48	FW	300	0.798	0.811
<i>SA</i> – 65	DFS+FW	400	0.820	0.821
<i>SA</i> – 66	OFS+FW	400	0.726	0.725
<i>SA</i> – 67	FSDS+FW	400	0.677	0.689
<i>SA</i> – 68	FW	400	0.799	0.849
<i>SA</i> – 85	DFS+FW	500	0.815	0.797
<i>SA</i> – 86	OFS+FW	500	0.723	0.720
<i>SA</i> – 87	FSDS+FW	500	0.670	0.684
<i>SA</i> – 88	FW	500	0.859	0.856

Tabela 9: Desempenho preditivo dos métodos para a seleção de atributos na classificação binária na base de dados MNIST.

Experimento	Método	Atributos	AM^B	AG^B
<i>SA</i> – 9	DFS+FW	100	0.820	0.787
<i>SA</i> – 10	OFS+FW	100	0.757	0.764
<i>SA</i> – 11	FSDS+FW	100	0.705	0.709
<i>SA</i> – 12	FW	100	0.781	0.816
<i>SA</i> – 29	DFS+FW	200	0.865	0.856
<i>SA</i> – 30	OFS+FW	200	0.738	0.763
<i>SA</i> – 31	FSDS+FW	200	0.703	0.707
<i>SA</i> – 32	FW	200	0.794	0.819
<i>SA</i> – 49	DFS+FW	300	0.846	0.853
<i>SA</i> – 50	OFS+FW	300	0.713	0.757
<i>SA</i> – 51	FSDS+FW	300	0.690	0.689
<i>SA</i> – 52	FW	300	0.827	0.843
<i>SA</i> – 69	DFS+FW	400	0.842	0.836
<i>SA</i> – 70	OFS+FW	400	0.713	0.736
<i>SA</i> – 71	FSDS+FW	400	0.687	0.684
<i>SA</i> – 72	FW	400	0.857	0.848
<i>SA</i> – 89	DFS+FW	500	0.835	0.793
<i>SA</i> – 90	OFS+FW	500	0.713	0.724
<i>SA</i> – 91	FSDS+FW	500	0.675	0.681
<i>SA</i> – 92	FW	500	0.863	0.850

Tabela 10: Desempenho preditivo dos métodos para a seleção de atributos na classificação multiclassificada na base de dados CIFAR-10.

Experimento	Método	Atributos	AM^M	AG^M
<i>SA</i> – 13	DFS+FW	100	0.809	0.796
<i>SA</i> – 14	OFS+FW	100	0.733	0.751
<i>SA</i> – 15	FSDS+FW	100	0.772	0.775
<i>SA</i> – 16	FW	100	0.706	0.709
<i>SA</i> – 33	DFS+FW	200	0.759	0.794
<i>SA</i> – 34	OFS+FW	200	0.710	0.750
<i>SA</i> – 35	FSDS+FW	200	0.764	0.769
<i>SA</i> – 36	FW	200	0.709	0.724
<i>SA</i> – 53	DFS+FW	300	0.744	0.785
<i>SA</i> – 54	OFS+FW	300	0.701	0.739
<i>SA</i> – 55	FSDS+FW	300	0.762	0.719
<i>SA</i> – 56	FW	300	0.712	0.775
<i>SA</i> – 73	DFS+FW	400	0.727	0.723
<i>SA</i> – 74	OFS+FW	400	0.694	0.723
<i>SA</i> – 75	FSDS+FW	400	0.718	0.711
<i>SA</i> – 76	FW	400	0.751	0.793
<i>SA</i> – 93	DFS+FW	500	0.718	0.711
<i>SA</i> – 94	OFS+FW	500	0.694	0.694
<i>SA</i> – 95	FSDS+FW	500	0.702	0.711
<i>SA</i> – 96	FW	500	0.798	0.794

Tabela 11: Desempenho preditivo dos métodos para a seleção de atributos na classificação multiclassificada na base de dados MNIST.

Experimento	Método	Atributos	AM^M	AG^M
<i>SA</i> – 17	DFS+FW	100	0.819	0.833
<i>SA</i> – 18	OFS+FW	100	0.734	0.752
<i>SA</i> – 19	FSDS+FW	100	0.780	0.784
<i>SA</i> – 20	FW	100	0.768	0.708
<i>SA</i> – 37	DFS+FW	200	0.817	0.823
<i>SA</i> – 38	OFS+FW	200	0.700	0.740
<i>SA</i> – 39	FSDS+FW	200	0.775	0.776
<i>SA</i> – 40	FW	200	0.772	0.742
<i>SA</i> – 57	DFS+FW	300	0.804	0.805
<i>SA</i> – 58	OFS+FW	300	0.689	0.697
<i>SA</i> – 59	FSDS+FW	300	0.775	0.763
<i>SA</i> – 60	FW	300	0.775	0.751
<i>SA</i> – 77	DFS+FW	400	0.768	0.793
<i>SA</i> – 78	OFS+FW	400	0.689	0.689
<i>SA</i> – 79	FSDS+FW	400	0.770	0.737
<i>SA</i> – 80	FW	400	0.780	0.799
<i>SA</i> – 97	DFS+FW	500	0.738	0.723
<i>SA</i> – 98	OFS+FW	500	0.685	0.687
<i>SA</i> – 99	FSDS+FW	500	0.741	0.732
<i>SA</i> – 100	FW	500	0.785	0.813

Tabela 12: Desempenho preditivo dos métodos de aprendizagem na classificação binária na base de dados Food-5k.

Experimento	Método	Lote	AM^B	AG^B
$AP - 1$	ILDA	100	0.901	0.910
$AP - 2$	ExStream	100	0.958	0.960
$AP - 3$	DSLDA	100	0.962	0.958
$AP - 4$	FW	100	0.924	0.911
$AP - 21$	ILDA	200	0.901	0.909
$AP - 22$	ExStream	200	0.948	0.950
$AP - 23$	DSLDA	200	0.957	0.956
$AP - 24$	FW	200	0.940	0.917
$AP - 41$	ILDA	300	0.895	0.893
$AP - 42$	ExStream	300	0.946	0.939
$AP - 43$	DSLDA	300	0.953	0.954
$AP - 44$	FW	300	0.942	0.924
$AP - 61$	ILDA	400	0.894	0.891
$AP - 62$	ExStream	400	0.941	0.935
$AP - 63$	DSLDA	400	0.929	0.949
$AP - 64$	FW	400	0.946	0.953
$AP - 81$	ILDA	500	0.893	0.887
$AP - 82$	ExStream	500	0.935	0.929
$AP - 83$	DSLDA	500	0.928	0.919
$AP - 84$	FW	500	0.949	0.962

Tabela 13: Desempenho preditivo dos métodos de aprendizagem na classificação binária na base de dados CIFAR-10.

Experimento	Método	Lote	AM^B	AG^B
<i>AP</i> – 5	ILDA	100	0.904	0.903
<i>AP</i> – 6	ExStream	100	0.946	0.952
<i>AP</i> – 7	DSLDA	100	0.969	0.972
<i>AP</i> – 8	FW	100	0.911	0.920
<i>AP</i> – 25	ILDA	200	0.892	0.897
<i>AP</i> – 26	ExStream	200	0.938	0.944
<i>AP</i> – 27	DSLDA	200	0.959	0.959
<i>AP</i> – 28	FW	200	0.913	0.923
<i>AP</i> – 45	ILDA	300	0.886	0.896
<i>AP</i> – 46	ExStream	300	0.933	0.938
<i>AP</i> – 47	DSLDA	300	0.954	0.955
<i>AP</i> – 48	FW	300	0.937	0.931
<i>AP</i> – 65	ILDA	400	0.882	0.895
<i>AP</i> – 66	ExStream	400	0.930	0.936
<i>AP</i> – 67	DSLDA	400	0.941	0.941
<i>AP</i> – 68	FW	400	0.945	0.947
<i>AP</i> – 85	ILDA	500	0.880	0.891
<i>AP</i> – 86	ExStream	500	0.925	0.932
<i>AP</i> – 87	DSLDA	500	0.923	0.937
<i>AP</i> – 88	FW	500	0.950	0.955

Tabela 14: Desempenho preditivo dos métodos de aprendizagem na classificação binária na base de dados MNIST.

Experimento	Método	Lote	AM^B	AG^B
<i>AP</i> – 9	ILDA	100	0.908	0.909
<i>AP</i> – 10	ExStream	100	0.957	0.951
<i>AP</i> – 11	DSLDA	100	0.976	0.957
<i>AP</i> – 12	FW	100	0.921	0.932
<i>AP</i> – 29	ILDA	200	0.907	0.907
<i>AP</i> – 30	ExStream	200	0.952	0.943
<i>AP</i> – 31	DSLDA	200	0.966	0.954
<i>AP</i> – 32	FW	200	0.937	0.934
<i>AP</i> – 49	ILDA	300	0.905	0.905
<i>AP</i> – 50	ExStream	300	0.939	0.934
<i>AP</i> – 51	DSLDA	300	0.960	0.943
<i>AP</i> – 52	FW	300	0.937	0.945
<i>AP</i> – 69	ILDA	400	0.894	0.900
<i>AP</i> – 70	ExStream	400	0.928	0.924
<i>AP</i> – 71	DSLDA	400	0.955	0.942
<i>AP</i> – 72	FW	400	0.950	0.962
<i>AP</i> – 89	ILDA	500	0.887	0.897
<i>AP</i> – 90	ExStream	500	0.925	0.923
<i>AP</i> – 91	DSLDA	500	0.923	0.927
<i>AP</i> – 92	FW	500	0.953	0.966

Tabela 15: Desempenho preditivo dos métodos de aprendizagem na classificação multi-classes na base de dados CIFAR-10.

Experimento	Método	Lote	AM^M	AG^M
<i>AP</i> – 13	ILDA	100	0.903	0.903
<i>AP</i> – 14	ExStream	100	0.958	0.958
<i>AP</i> – 15	DSLDA	100	0.995	0.994
<i>AP</i> – 16	FW	100	0.871	0.883
<i>AP</i> – 33	ILDA	200	0.898	0.901
<i>AP</i> – 34	ExStream	200	0.952	0.958
<i>AP</i> – 35	DSLDA	200	0.991	0.992
<i>AP</i> – 36	FW	200	0.873	0.885
<i>AP</i> – 53	ILDA	300	0.893	0.897
<i>AP</i> – 54	ExStream	300	0.946	0.935
<i>AP</i> – 55	DSLDA	300	0.991	0.990
<i>AP</i> – 56	FW	300	0.904	0.907
<i>AP</i> – 73	ILDA	400	0.883	0.889
<i>AP</i> – 74	ExStream	400	0.942	0.929
<i>AP</i> – 75	DSLDA	400	0.986	0.990
<i>AP</i> – 76	FW	400	0.913	0.916
<i>AP</i> – 93	ILDA	500	0.883	0.880
<i>AP</i> – 94	ExStream	500	0.939	0.926
<i>AP</i> – 95	DSLDA	500	0.978	0.988
<i>AP</i> – 96	FW	500	0.931	0.920

Tabela 16: Desempenho preditivo dos métodos de aprendizagem na classificação multi-classes na base de dados MNIST.

Experimento	Método	Lote	AM^M	AG^M
<i>AP</i> – 17	ILDA	100	0.900	0.910
<i>AP</i> – 18	ExStream	100	0.958	0.954
<i>AP</i> – 19	DSLDA	100	0.976	0.983
<i>AP</i> – 20	FW	100	0.891	0.914
<i>AP</i> – 37	ILDA	200	0.900	0.905
<i>AP</i> – 38	ExStream	200	0.957	0.951
<i>AP</i> – 39	DSLDA	200	0.973	0.981
<i>AP</i> – 40	FW	200	0.899	0.943
<i>AP</i> – 57	ILDA	300	0.887	0.892
<i>AP</i> – 58	ExStream	300	0.950	0.946
<i>AP</i> – 59	DSLDA	300	0.970	0.980
<i>AP</i> – 60	FW	300	0.907	0.950
<i>AP</i> – 77	ILDA	400	0.884	0.890
<i>AP</i> – 78	ExStream	400	0.950	0.941
<i>AP</i> – 79	DSLDA	400	0.969	0.980
<i>AP</i> – 80	FW	400	0.911	0.955
<i>AP</i> – 97	ILDA	500	0.883	0.881
<i>AP</i> – 98	ExStream	500	0.940	0.937
<i>AP</i> – 99	DSLDA	500	0.965	0.977
<i>AP</i> – 100	FW	500	0.955	0.958