UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Faculdade de Engenharia
Câmpus de São João da Boa Vista

MAILSON GONÇALVES DA SILVA

Classificador Open-World para o Reconhecimento de Ameaças em Imagens Aéreas de Culturas Agrícolas

São João da Boa Vista - SP
2023

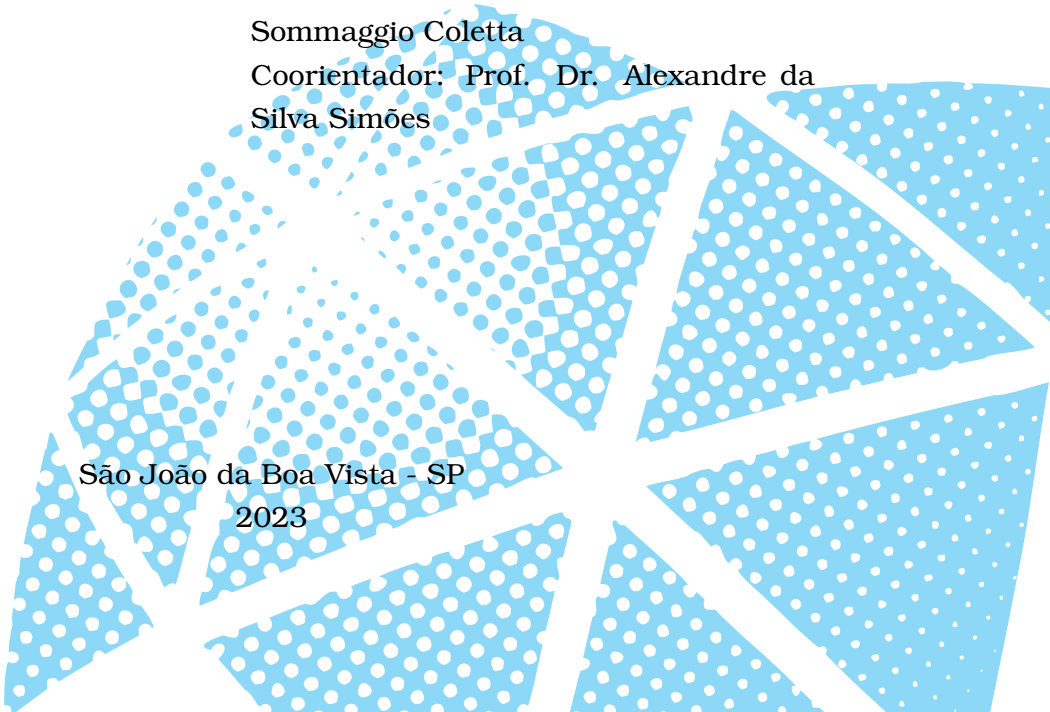
MAILSON GONÇALVES DA SILVA

Open-World Classifier for Threat Recognition in Aerial Images of Agriculture Crops

Classificador Open-World para o Reconhecimento de Ameaças em Imagens Aéreas de Culturas Agrícolas

Texto apresentado ao Programa de Pós-Graduação em Engenharia Elétrica (PGEE) da Universidade Estadual Paulista "Júlio de Mesquita Filho" - Faculdade de Engenharia - Campus de São João da Boa Vista como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Luiz Fernando Sommaggio Coletta
Coorientador: Prof. Dr. Alexandre da Silva Simões



São João da Boa Vista - SP
2023

S586c

Silva, Mailson Gonçalves da

Classificador Open-World para o reconhecimento de ameaças em imagens aéreas de culturas agrícolas / Mailson Gonçalves da Silva. -- São João da Boa Vista, 2023

105 p. : il., tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Faculdade de Engenharia, São João da Boa Vista

Orientador: Luiz Fernando Sommaggio Coletta

Coorientador: Alexandre da Silva Simões

1. Agricultura de precisão. 2. Inteligência artificial. 3. Processamento de imagens. 4. Pragas agrícolas. 5. Reconhecimento de padrões. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Engenharia, São João da Boa Vista. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Impacto esperado desta pesquisa

Esta pesquisa buscou apresentar uma metodologia para o desenvolvimento de classificadores capazes de aprender novas classes de forma incremental (na literatura chamado de modelos Open-World), com o mínimo de intervenção humana, sendo avaliada em um cenário agrícola. Espera-se que a abordagem proposta contribua na obtenção de modelos mais flexíveis e escaláveis, proporcionando uma melhor adaptação ao dinamismo do conjunto de dados e ampliando, também, a capacidade operacional de modelos de aprendizado de máquina.

Considerando o contexto agrícola, o uso de modelos Open-World pode auxiliar no combate de doenças e pragas ao permitir uma estratégia direcionada aos focos de tais ameaças no início de sua ocorrência, evitando o uso demasiado de agrotóxicos por exemplo. Por meio disso, é possível mitigar custos de produção, impactos ambientais e, também, melhorar a qualidade dos produtos uma vez que estarão em menos contato com defensivos agrícolas.

Potencial impact of this research

This research sought to present a methodology for developing classifiers capable of learning new classes incrementally (named in the literature as Open-World models), with minimal human intervention, being evaluated in an agricultural scenario. It is expected that the proposed approach will contribute to obtaining more flexible and scalable models, providing better adaptation to the dynamism of the data set and also improving the operational capacity of machine learning models.

Considering the agricultural context, the use of Open-World models can help in the fight against diseases and pests by allowing a strategy targeted at the sources of such threats at the beginning of their occurrence, avoiding the excessive use of pesticides, for example. Through this, it is possible to mitigate production costs, environmental impacts and also improve the quality of products since they will be in less contact with agricultural pesticides.

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Sorocaba

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Classificador Open-World para o Reconhecimento de Ameaças em Imagens Aéreas de Culturas Agrícolas

AUTOR: MAILSON GONÇALVES DA SILVA
ORIENTADOR: LUIZ FERNANDO SOMMAGGIO COLETTA
COORIENTADOR: ALEXANDRE DA SILVA SIMÕES

Aprovado como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica, área: Sistemas Eletrônicos pela Comissão Examinadora:

Dr. LUIZ FERNANDO SOMMAGGIO COLETTA (Participaçao Virtual)
Dell Technologies

Prof. Dr. LEOPOLDO ANDRÉ DUTRA LUSQUINO FILHO (Participaçao Virtual)
Departamento de Engenharia de Controle e Automação / Câmpus de Sorocaba

Prof. Dr. PABLO ANDRETTA JASKOWIAK (Participaçao Virtual)
Departamento de Engenharias da Mobilidade / Universidade Federal de Santa Catarina

Sorocaba, 25 de agosto de 2023

Agradecimentos

Agradeço a toda minha família, professores e servidores da Universidade Estadual Paulista Paulista que contribuíram diretamente ou indiretamente com a realização deste trabalho. Em especial agradeço:

- ao Prof. Dr. Luiz Fernando Sommaggio Coletta pela orientação, ensinamentos e todo suporte fornecido ao longo do desenvolvimento desta pesquisa, tornando possível a sua realização dentro do prazo estipulado;
- ao ex-docente do programa o Prof. Dr. Gilliard Nardel Malheiros Silveira[1] , pelo acompanhamento inicial, pelos trabalhos desenvolvidos em conjunto e por todos os ensinamentos passados;
- ao Prof. Dr. André Luis Debiaso Rossi pelas suas contribuições na confecção do artigo de revista obtido a partir dessa pesquisa;
- por fim, ao coordenador o Prof. Dr. Marcelo Luís Francisco Abbade e o conselho do programa, por todo auxilio fornecido diante dos imprevistos ocorridos.

[1] O Prof. Dr. Gilliard foi meu orientador no início do curso de Mestrado em 2021, mas por conta do encerramento de seu vínculo empregatício com a UNESP foi necessário iniciar um novo projeto de pesquisa em 2022. Desta vez, sob orientação do Prof. Dr. Luiz Fernando Sommaggio Coletta.

"O verdadeiro perigo não é que computadores começarão a pensar como homens,
mas que homens começarão a pensar como computadores."
Sydney J. Harris

Resumo

A Agricultura de Precisão aliada ao Sensoriamento Remoto tem provocado avanços sem precedentes nas mais diversas atividades do campo. Com este progresso, o uso de Veículos Aéreos Não Tripulados (VANTs) tem permitido a obtenção de imagens de alta qualidade a um custo cada vez mais baixo. Isto tem naturalmente incentivado uma explosão de aplicações a partir do uso desses equipamentos, como, por exemplo, as que envolvem o combate às ameaças em culturas (doenças, pragas e ervas daninhas), que afetam a produtividade e implicam em gastos, muitas vezes descontrolados. Este trabalho leva em conta o Aprendizado de Máquina (AM) como um componente essencial para o sucesso e automação do monitoramento de ameaças no campo. Mais especificamente, considera-se neste trabalho modelos de AM para o reconhecimento de padrões em imagens coletados por VANTs, de maneira a detectar automaticamente pragas/doenças em uma cultura agrícola, minimizando, assim, o esforço humano nesta tarefa. Em especial, busca-se um classificador que seja escalável, no sentido de ser apto a assimilar incrementalmente novos padrões de ameaças, a fim de manter o bom desempenho do modelo mesmo diante do comportamento dinâmico dos dados quando em produção. Assim, esta pesquisa propõe um *framework* que viabiliza a obtenção de classificadores flexíveis o suficiente para incorporar novas classes, antes inexistentes no treinamento do modelo, à medida que aparecem em novos dados a serem classificados. Inspirado em *Active Learning*, a alternativa proposta gera classificadores de Cenário Aberto (também conhecido como *Open-World*), tornando possível a expansão da sua base de conhecimento com o mínimo de intervenção humana. A estrutura é experimentada sobre um conjunto de dados de uma cultura de eucaliptos com focos da doença *Ceratocystis wilt*, a qual deve ser incrementalmente aprendida pelo modelo. Como estratégias para detecção de nova classe, foram estudados critérios baseados em entropia e silhueta, com a proposição de uma versão modificada da silhueta atuando em conjunto com o classificador incremental iCaRL para compor, assim, o modelo *Open-World* testado. Os resultados alcançados superaram os atingidos com o modelo NNO da literatura, com destaque para o uso de menor quantidade de dados rotulados no processo. Entre os critérios testados, a silhueta proposta se mostrou mais robusta ao desbalanceamento de classes e, por fim, os experimentos evidenciaram que o *framework* proposto é uma alternativa válida para a obtenção de modelos *Open-World*.

Palavras-chave: *Open-World Recognition*. Detecção de Novidades. Visão Computacional. Agricultura de Precisão. Sensoriamento Remoto.

Abstract

Precision Agriculture allied to Remote Sensing has caused advances without precedents in the most diverse field activities. With this progress, the use of Unmanned Aerial Vehicles (UAVs) has allowed obtaining high quality images at an increasingly low cost. This has naturally encouraged an explosion of applications from the use of this equipment, such as, for example, those involving the fight against threats to cultures (diseases, pests and weeds), that affect productivity and imply expenses, which are often uncontrolled. This work considers Machine Learning (ML) as an essential component for the success and automation of threat monitoring in the field. More specifically, it is considered in this work ML models for the recognition of patterns in images collected by UAVs, in order to automatically detect pests/diseases in an agricultural crop, minimizing human effort in this task. In particular, a scalable classifier is sought, in the sense of being able to incrementally assimilate new threat patterns, in order to maintain the good performance of the model even before the dynamic behavior of data when in production. Thus, this research proposes a framework that makes it possible to obtain classifiers that are flexible enough to incorporate new classes, which did not exist before in model training, as they appear in new data to be classified. Inspired by Active Learning, the proposed alternative generates Open World classifiers, making it possible to expand its knowledge base with minimal human intervention. The structure is tried on a set of data from a eucalyptus crop with Ceratocystis wilt disease foci, which must be incrementally learned by the model. As strategies for new class detection, criteria based on entropy and silhouette were studied, with the proposition of a modified version of the silhouette working together with the iCaRL incremental classifier to compose, thus, the tested Open-World model. The results achieved surpassed those achieved with the NNO model in the literature, mainly the use of lesser amount of labeled data in the process. Among the tested criteria, the proposed silhouette proved to be more robust to class imbalance and, finally, the experiments showed that the proposed framework is a valid alternative for obtaining Open-World models.

Keywords: Open-World Recognition. Novelty Detection. Computer vision. Precision agriculture. Remote sensing.

Lista de Figuras

1 Ciclo de execução de um algoritmo de AL. 21
2 Demonstração do desempenho do uso de AL em um conjunto de dados sintético. (a) Conjunto de dados de 400 amostras geradas por duas Gaussianas. (b) Fronteira de decisão obtida ao amostrar 30 objetos aleatoriamente para treinamento (acurácia de 0,7). (c) Fronteira de decisão obtida ao amostrar 30 objetos para treinamento através de AL (acurácia de 0,9). 22
3 Diferença entre processamento em lote e processamento em fluxo de dados. 25
4 Em *Task-IL* a etapa de predição conta com um identificador de *task* (*task*-ID) enquanto que *Class-IL* não faz uso desse artifício. 28
5 Categorias de métodos para detecção de novidades. 31
6 Amostras de imagens que compõe as bases de dados MNIST (à esquerda) e LFW (à direita). 32
7 Problemas de visão computacional em ordem crescente do valor de *openness*. Para alguns casos não é possível ter conhecimento de todas as classes durante o treino e deve-se então esperar que classes desconhecidas possam surgir ao longo da etapa de teste. 34
8 Comparação entre o *Cenário Fechado* e o *Cenário Aberto*. Enquanto o primeiro assume que novas classes não irão aparecer subsequente ao treinamento do modelo, o outro aprende algumas classes na etapa de treino, mas é consciente de que classes nunca vistas possam surgir sendo robusto a esta situação. 35
9 Na esquerda um sistema de visão computacional aplicado em carros autônomos da Tesla e na direta um modelo de detecção aplicado no contexto agrícola. Os dois casos precisam lidar com um ambiente que pode apresentar padrões não previstos para continuar cumprindo sua finalidade. 37
10 Esquema ilustrativo de um sistema *Open World* 38
11 Estrutura de um Autoencoder clássico. 46
12 Estrutura de um Autoencoder Variacional. 48
13 Fluxo de processamento no VAE. 52
14 Esquema ilustrativo do funcionamento do *framework* para o reconhecimento de novas classes. 55
15 Abordagem da Silhueta Baseada em Centroide para a detecção de novidades. 59

16 Abordagem da Silhueta Baseada em Instância para a detecção de novidades. 59
17 Representação dos critérios de a) Silhueta Baseada em Centroide e b) Silhueta Baseada em Instância, adaptadas para o modelo iCaRL. 60
18 Imagem capturada por drone em que os círculos vermelhos destacam algumas árvores de eucalipto doentes. 63
19 Classes que constituem o *dataset* em avaliação. 63
20 Estrutura do Autoencoder Variacional implementado, onde μ e σ^2 representam, respectivamente, a média e variância das variáveis latentes. 65
21 Gráfico destacando a variação dos critérios de detecção de novas classes quando um novo padrão surge nos dados. 69
22 Comportamento dos critérios de detecção de novas classes no conjunto de dados MNIST, onde o dígito 1 é a novidade a ser aprendida. O *baseline* é apresentado em a), o critério de entropia em b), a Silhueta Baseada em Instância em c) e de d) a f) são apresentados o critério de Silhueta Baseada em Centroide nas iterações 1,2 e 4, respectivamente. 72
23 Perfil dos critérios de detecção de novos padrões sobre os dados de teste: a) na iteração 1 e b) na iteração 3 do conjunto dp_cerotocystis1; c) na iteração 1 e d) na iteração 3 do conjunto dp_cerotocystis20. 74
24 Perfil dos critérios de detecção de novos padrões sobre os dados de teste: a) na iteração 1 e b) na iteração 3 do conjunto hc_cerotocystis1; c) na iteração 1 e d) na iteração 3 do conjunto hc_cerotocystis20. 75
25 Perfil dos critérios de detecção de novos padrões sobre os dados de teste: a) na iteração 1 e b) na iteração 3 do conjunto vae_cerotocystis1; c) na iteração 1 e d) na iteração 3 do conjunto vae_cerotocystis20. . . . 76
26 Acurácia sobre todos os dados não-rotulados para a) dp_ceratocystis1, b) dp_ceratocystis20, c) hc_ceratocystis1, d) hc_ceratocystis20, e) vae_ceratocystis1 e f) vae_ceratocystis20. 78
27 Mapas de calor da quantidade de amostras selecionadas da nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes. 79
28 Mapas de calor da acurácia de classificação (%) apenas sobre a nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes. 80
29 Gráficos da acurácia sobre todas as classes à cada iteração, para a representação em *deep features* considerando os modelos: a) NNO, b) iCaRL com seleção aleatória, c) iCaRL com entropia, d) iCaRL com Silhueta Baseada em Instância e e) iCaRL com Silhueta Baseada em Centroide. 85

30 Gráficos da acurácia sobre todas as classes à cada iteração, para a representação em *hand-crafted features* considerando os modelos: a) NNO, b) iCaRL com seleção aleatória, c) iCaRL com entropia, d) iCaRL com Silhueta Baseada em Instância e e) iCaRL com Silhueta Baseada em Centroide. 86
31 Gráficos da acurácia sobre todas as classes à cada iteração, para a representação em variáveis latentes com VAE considerando os modelos: a) NNO, b) iCaRL com seleção aleatória, c) iCaRL com entropia, d) iCaRL com Silhueta Baseada em Instância e e) iCaRL com Silhueta Baseada em Centroide. 87
32 Mapas de calor da quantidade de amostras selecionadas da nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes. 88
33 Mapas de calor da acurácia de classificação (%) apenas sobre a nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes. 90

Lista de Tabelas

1 Descrição de cada *dataset-base* obtido pelo processo de amostragem dos dados originais. 64

2 Valores de *F1-score* alcançado sobre todas as classes juntas em todos os experimentos. 81

3 Valores de *F1-score* alcançado sobre todas as classes juntas em todos os experimentos. 83

Sumário

1 INTRODUÇÃO 14
1.1 Objetivo 18
1.2 Organização do Texto 18

2 REVISÃO DE LITERATURA 20
2.1 Aprendizado Ativo 20
2.2 Aprendizado Incremental 24
2.2.1 Class-Incremental Learning 27
2.3 Detecção de novidades 29
2.3.1 Open-Set 32
2.3.2 Open World Recognition 36
2.3.3 Alternativas de algoritmos Open-World 40
2.3.3.1 Algoritmo IC-EDS 40
2.3.3.2 Nearest Non-Outlier (NNO) 42
2.3.3.3 Incremental Classifier and Representation Learning (iCaRL) . 44
2.4 Representação de dados: Autoencoder Variacional 45
2.5 Trabalhos Relacionados no Contexto Agrícola 52

3 ABORDAGEM PROPOSTA 54
3.1 Procedimento de aprendizado incremental de classes 54
3.2 Critérios para detecção de novas classes 57

4 METODOLOGIA DA PESQUISA 62
4.1 Materiais 62
4.2 Dataset 62
4.3 Espaço de Características 64
4.4 Configurações do Modelo 66
4.4.1 Procedimento experimental para o conjunto MNIST 67
4.4.2 Procedimento experimental para os *datasets* gerados com imagens de VANT 67
4.5 Métricas de avaliação dos resultados 68

5 RESULTADOS E DISCUSSÃO 71
5.1 Experimentos Ilustrativos Sobre o Conjunto MNIST 71
5.2 Avaliação dos Critérios de Detecção de Novidades Sobre Uma Tarefa do Mundo Real 74

5.3 Avaliação do *framework* em cenário *Open World* 82

6 CONCLUSÕES E TRABALHOS FUTUROS 91

BIBLIOGRAFIA 93

1 INTRODUÇÃO

O avanço da pesquisa e das aplicações de Aprendizado de Máquina (AM) (KUBAT, 2015; BERNARD, 2021; MITCHELL et al., 1997), com destaque ao *Deep Learning* (DL) (GOODFELLOW; BENGIO; COURVILLE, 2016), têm permitido com sucesso abordar muitos problemas complexos, os quais muitas vezes envolvem dados de alta dimensionalidade e/ou não-estruturados, como os provenientes de imagens, áudios e textos. Técnicas baseadas em *Autoencoders* (HINTON; SALAKHUTDINOV, 2006), Redes Neurais Convolucionais (CHAUHAN; GHANSHALA; JOSHI, 2018) e Redes Neurais Recorrentes (HU et al., 2020) tem beneficiado principalmente a área de visão computacional, a qual hoje em dia tem alcançado resultados surpreendentes em tarefas de classificação (HE, 2020; YU et al., 2022), detecção de objetos (RAKHSITH et al., 2021; WANG; YEH; LIAO, 2021; TERVEN; CORDOVA-ESPARZA, 2023) e segmentação de imagens (GEORGE; SCHROFF; ADAM, 2018; MO et al., 2022; KIRILLOV et al., 2023).

Com o tempo, naturalmente, os aperfeiçoamentos das implementações de AM levaram seu ferramental para além do âmbito computacional, de maneira a oferecer soluções assertivas para questões que impactam a sociedade em diferentes ramos, como o da engenharia, biologia, medicina, direito, agronomia, etc. (SHARP; AK; HEDBERG, 2018; MIRAFTABZADEH et al., 2019; BENOS et al., 2021; GARG; MAGO, 2021). Em especial, o grande volume de estudos na área tem permitido compreender, cada vez mais, as nuances envolvendo o funcionamento de diferentes modelos de aprendizado, de maneira que vários de seus pressupostos e limitações, outrora considerados razoáveis, estão hoje mais suscetíveis ao debate pela comunidade científica para, em algum momento, serem superados. Com essa ideia, o presente trabalho se preocupa com limitações de modelos supervisionados para a classificação de dados, os quais tem sido utilizados praticamente da mesma forma há mais de 50 anos (MINSKY, 1961; HO; AGRAWALA, 1968).

Fundamentalmente, os chamados classificadores de dados têm como objetivo aprender padrões/conceitos para, subsequentemente, predizê-los em instâncias do problema nunca antes apresentadas ao modelo (MITCHELL et al., 1997). Esta operação de aprendizado se dá em duas etapas (HAN; KAMBER, 2006): **(i)** inicialmente, um modelo/classificador, $\hat{f}(\mathbf{x}) = y$, é induzido a partir de um conjunto de dados de treino de N amostras, $\mathscr{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, no qual cada objeto $\mathbf{x}_i$ está rotulado de acordo com a classe a qual pertence podendo, portanto, ser representado como sendo uma tupla $<\mathbf{x}_i, y_\ell^i>$ em que $y_\ell \in C = \{c_1, \ldots, c_k\}$ trata-se de um conjunto finito de k classes (ex., *Planta*, *Daninha*, *Solo*, etc.); **(ii)** a seguir, o classificador $\hat{f}(\mathbf{x})$ obtido pode ser usado para inferir a classe, y^j, de novos objetos $\mathbf{x}_{j>N}$ não presentes na etapa de treinamento, cujo rótulo de classe se desconhece. Na prática, um sistema deste tipo pode ser construído a partir de um conjunto

significativamente grande de dados rotulados, o qual pode ser dividido em subconjuntos de treino, validação e/ou teste (KOHAVI, 1995), cada qual com amostras (balanceadas) de k classes conhecidas do problema que se tem em mãos. Enquanto o subconjunto de treino é usado para induzir o modelo propriamente dito, o restante pode avaliá-lo, ajudando a simular sua etapa preditiva e também realizar o *fine-tunning* dos hiperparâmetros (GARETH et al., 2013). Normalmente, este procedimento, em *batch* (ou também conhecido como *Offline Learning* (GÉRON; SAFARI, 2019)), permite colocar o classificador em produção, momento em que já treinado, validado/testado, permanecerá fixo (inalterável) predizendo a classe de objetos do mundo real nunca antes vistos pelo modelo. Destaca-se, como será visto a seguir, que esta abordagem é conivente com algumas premissas que dificultam o uso dos classificadores em muitos problemas práticos.

As dificuldades que classificadores de dados perpassam, especificamente, provém da necessidade de significativa quantidade de dados rotulados e que, ao mesmo tempo, possuam boa qualidade de representação das classes do problema (SETTLES, 2009). Como exemplo clássico, apesar do paradigma de DL produzir com frequência algoritmos no estado-da-arte, faz-se necessário, neste contexto, uma massiva quantidade de dados pré-rotulados para atingir tal nível de desempenho (ROH; HEO; WHANG, 2021). Para isso, normalmente é necessário, então, que um *especialista* rotule (manualmente) uma grande quantidade de dados coletados e isso torna o processo muito custoso e sujeito à erros, além de denotar uma intensa dependência humana para se ter bons classificadores. Por muitos anos, formas de amenizar esse gargalo têm sido exploradas através do Aprendizado Semi-Supervisionado (ZHU; GOLDBERG, 2009; CHAPELLE; SCHöLKOPF; ZIEN, 2006), o qual busca tirar proveito de dados não rotulados para aprimorar o desempenho dos modelos quando estes dispõem de apenas poucos dados rotulados.

Modelos de classificação também encaram o dilema de *bias*-variância, que explica em termos teóricos o motivo de um algoritmo de classificação apresentar boa *performance* em certos contextos e em outros não (OZA; TUMER, 2008; GEMAN; BIENENSTOCK; DOURSAT, 1992; MERENTITIS; DEBES; HEREMANS, 2014). O *bias* pode surgir de diferentes fontes, entre elas a própria formulação matemática do modelo, de tal forma que, a solução encontrada sobre diferentes conjuntos de dados tende a seguir sempre o viés da forma de construção do algoritmo e, portanto, soluções ruins podem ser propostas em alguns casos. Um remédio típico para amenizar esta questão é o uso de *ensembles* (ACHARYA et al., 2014; COLETTA et al., 2015), o qual envolve diversos modelos treinados sobre subconjuntos de dados distintos e que ao serem combinados (através de alguma estratégia de consenso) providenciam fronteiras de decisão mais flexíveis do que aquelas obtidas por um único modelo de classificação atuando isoladamente.

Outro entrave para os classificadores, mais próximo da motivação deste trabalho, reside na dificuldade dos modelos em perceber mudanças nos dados que se está classificando. Em específico, classificadores convencionais são, geralmente, obtidos a partir de

dados provenientes de ambientes controlados (BENDALE; BOULT, 2015), fato que colabora com a premissa de que as amostras de dados oferecidas ao modelo estejam sempre distribuídas de forma *idêntica* e *independente* ao longo do tempo (também chamado de *i.i.d. – independent and identically distributed*) (CASTEELS; HELLINCKX, 2020). Assim, por exemplo, considera-se que os conjuntos de treino e validação/teste são sempre oriundos de uma mesma distribuição de dados, não havendo mudanças. Em contrapartida o mundo real é dinâmico, de tal forma que o padrão de distribuição de instâncias de classes conhecidas pode sofrer *drift* (distorção) com o tempo (MASUD et al., 2011), deixando de ser *i.i.d.* Em outras palavras, mudanças podem ocorrer nos dados, principalmente ao se considerar o cenário em que novos dados possam surgir com o tempo para serem classificados. Assim, contextos como de *data streams* em que técnicas baseadas em *Online Learning* (SHALEV-SHWARTZ, 2012) são aplicadas, devem ser tratados com modelos mais robustos e conscientes de que novos dados possam estar fora da distribuição (na literatura também chamados de *o.o.d.*, *out of distribution*).

Com base no exposto, é natural considerar que, em cenários mais realístas, as mudanças na distribuição dos dados poderiam evoluir de tal maneira que signifiquem o surgimento de novos padrões/conceitos (MASUD et al., 2011). Tal situação desafia modelos de classificação convencionais, já que não são capazes de lidar com a presença de novas classes por assumirem sempre um único e finito conjunto de classes para o problema que se está tratando (SCHEIRER et al., 2013). Algumas abordagens podem estar na direção de resolver este problema ao se preocuparem em identificar conceitos *desconhecidos* nos dados, como os métodos de detecção de novidades e anomalias (VIGNOTTO; ENGELKE, 2018). Contudo, algoritmos desse tipo normalmente não consideram atualizar a sua base de conhecimento com os novos conceitos.

Scheirer et al. (2013), em especial, propuseram o contexto de *Open-Set*, onde assume-se que a fase de teste pode conter instâncias de classes não apresentadas na fase de treinamento e, para lidar com isso, o algoritmo busca aprender os limites de decisão que separam as regiões onde há classes conhecidas daquelas que possuem classes desconhecidas. Em termos práticos, o modelo deve ser capaz de identificar os dados de entrada como sendo de uma das classes conhecidas ou então atribui-se o rótulo "*desconhecido*" para a nova instância. Posteriormente, Bendale e Boult (2014) apresentaram uma nova categoria de problemas chamada de *Open World Recognition*, que lida com conjuntos de dados dinâmicos, onde novas classes podem surgir durante a operação do modelo de classificação e este deve, então, ser capaz de detectar e aprender esse novo conceito de forma incremental. Assim, um classificador, após a fase de treino, estaria apto a discriminar um número fixo de K classes conhecidas, $C^k = \{c_1, c_2, \ldots, c_K\}$, mas poderia também expandir seu reconhecimento sobre um valor previamente desconhecido de L novas classes, $C^u = \{c_{K+1}, \ldots, c_{K+L}\}$, de maneira a permitir, dinamicamente, o aprendizado contínuo de classes existentes para um determinado problema, $C^{all} = C^k \cup C^u$. Diante disso, o ob-

jetivo principal do presente trabalho foi propor e experimentar uma nova metodologia para obtenção de classificadores para o contexto *Open World*, onde novas classes podem surgir devendo ser assimiladas pelo modelo para que este consiga manter um bom desempenho e consistência com a realidade.

Como a exploração e avaliação dos mecanismos estudados e propostos nesta pesquisa dependem de um ambiente mais dinâmico e realista, pelo próprio contexto *Open World* a que se refere, optou-se por realizar as experimentações necessárias em um cenário agrícola. Naturalmente, a agricultura é um exemplo prático onde situações inesperadas podem ocorrer, pois devido a alterações climáticas ou então induzidas pelo homem, mudanças drásticas podem acontecer pondo em risco a produção (CEPEA, 2019). Entre as ameaças frequentes estão as pragas e plantas invasoras. As pragas ocorrem quando uma superpopulação de insetos, fungos, vírus, ou outro ser vivo se alastram pela plantação reduzindo a qualidade dos produtos e até mesmo levando a planta à morte (EMBRAPA, 2020). O mesmo pode ocorrer quando há presença de plantas invasoras, também chamadas de ervas daninhas. Plantas que não fazem parte da cultura podem crescer e se espalhar pela lavoura, gerando competição pelos nutrientes do solo e inclusive atraindo novas pragas que não são naturais daquele cultivo. Segundo a Embrapa (EMBRAPA, 2022), as plantas daninhas podem causar até 90% das perdas em cultivos se não forem controladas, representando um dos fatores que mais afeta o rendimento agrícola (CARVALHO; GUEDES; SALAME, 2020). Para enfrentar esses problemas, muitos produtores fazem uso de defensivos agrícolas cujas aplicações, muitas vezes, são feitas inadequadamente e/ou de forma abusiva (JÚNIOR et al., 2020). Com isso, incorrem elevadas despesas onerando também os custos de produção. Analisando também pelo viés socioambiental, defensivos em excesso podem provocar danos ao meio ambiente e aos seres humanos (SERRA et al., 2016).

Na literatura, muitos trabalhos têm lançado mão de técnicas computacionais sofisticadas, como as de Inteligência Artificial (IA) (RUSSELL; NORVIG, 2002), para apresentar soluções que aumentem a produtividade ao lidar com as perdas e ameaças no campo (MESHRAM et al., 2021). Ainda assim, são escassas as referências na literatura que usam abordagens como de *Open-Set* ou *Open World* para lidar com problemas em plantações. Assim, o intuito deste trabalho é também contribuir para o avanço da Agricultura. Especificamente, auxiliando na identificação de novas ameaças em lavouras a partir de imagens obtidas por um VANT (Veículo Aéreo Não Tripulado) e um classificador *Open World*, capaz de detectar e aprender novas classes.

O *framework* proposto para esta finalidade foi pensado de maneira a permitir o uso de diferentes classificadores, espaços de *features* e técnicas de detecção de novas classes, sendo configurável de acordo com a aplicação. Neste trabalho, tal proposta foi experimentada sobre imagens aéreas de uma plantação de eucalipto infectada com uma doença chamada *Ceratocystis wilt* (murcha de *Ceratocystis*), cujas classes (*Solo, Planta Saudável*

e Doença) são desbalanceadas. O objetivo da modelagem foi o de aprender a detectar a doença que, a priori, era desconhecida pelo sistema. Para identificar os objetos candidatos a uma nova classe, foram utilizados os critérios de silhueta e entropia (Seção 3.2). E como será explicado em mais detalhes adiante (Seção 3.1), o procedimento de aprendizado das novas classes ocorre de forma iterativa, semelhante a *Active Learning*, onde a cada iteração do processo um classificador é retreinado utilizando amostras candidatas de uma nova classe, detectadas pelos critérios de seleção. Os resultados obtidos nos experimentos foram comparados com os modelos IC-EDS (Seção 5.2) e NNO (Seção 5.3) da literatura para análise de desempenho.

1.1 Objetivo

O objetivo geral desta pesquisa foi o de investigar classificadores e métodos de detecção de novidades nos dados que possam estar em consonância com o contexto de *Open World*, para isso, um *framework* foi proposto. Assim, os objetivos específicos deste trabalho foram:

1. Propor uma estrutura algorítmica para tornar classificadores mais aptos a atuarem em cenários mais realistas, cujo dinamismo leva a mudanças nos dados e ao surgimento de novas classes;

2. Analisar o comportamento da estrutura proposta em **1** sobre imagens aéreas provindas de um VANT atuando em um contexto agrícola, o qual envolve a necessidade de detecção e aprendizagem (rápida) de padrões de ameaças em culturas;

3. Avaliar qualitativamente e quantitativamente os resultados obtidos em **2** com vistas a destacar o desempenho de cada componente do *framework* observando-se também suas interrelações.

1.2 Organização do Texto

Este trabalho está organizado conforme o disposto a seguir:

- Capítulo 1 apresenta a introdução e os objetivos do trabalho;

- Capítulo 2 expõe a base teórica sobre Autoencoder Variacional, *Active Learning*, *Incremental Learning*, *Open World Recognition*, entre outros tópicos importantes para o desenvolvimento da pesquisa, além de elencar os trabalhos relacionados ao tema da pesquisa;

- Capítulo 3 apresenta a abordagem proposta para o aprendizado incremental e contínuo de novas classes;

- Capítulo 4 discorre sobre os materiais, métodos e procedimentos experimentais aplicados para implementação e avaliação da abordagem proposta no capítulo 3;
- Capítulo 5 exibe e discute os resultados obtidos sobre os experimentos conduzidos;
- Capítulo 6 faz a conclusão do trabalho sintetizando as principais observações realizadas sobre os experimentos e, por fim, apresenta oportunidades de pesquisas futuras sobre o tema.

2 REVISÃO DE LITERATURA

Neste capítulo são apresentadas as bases teóricas que fundamentaram o desenvolvimento do *framework* proposto. A identificação e rotulação de novidades como sendo amostras de uma nova classe desconhecida pelo modelo nesta pesquisa, se inspira no princípio de Aprendizado Ativo, ou *Active Learning* (AL), cuja descrição é provida na Seção 2.1. Com base em AL, novos conceitos/padrões descobertos e passíveis de enriquecer um determinado modelo de aprendizado, podem ser aplicados em sua própria atualização tendo como fundamento o Aprendizado Incremental, o qual é revisado na Seção 2.2. Estes assuntos, amplamente difundidos na literatura especializada, podem, então, ajudar a tornar classificadores mais aptos para atuarem em cenários desafiadores, como os de *Open-Set* e *Open World*, os quais são descritos na Seção 2.3.

A Seção 2.4 discorre sobre a definição matemática e teórica de um *Autoencoder* Variacional, alternativa que foi utilizada neste trabalho com o intuito de fornecer um espaço de características mais interessante para a avaliação da metodologia concebida. Por fim, a Seção 2.5 discorre sobre os trabalhos que estão mais próximos ao tema desta pesquisa.

2.1 Aprendizado Ativo

No processo de indução de um modelo supervisionado necessita-se de várias amostras rotuladas das classes de interesse para se abstrair as fronteiras de decisão que melhor discriminam os dados (SETTLES, 2009). Atualmente, com o advento de DL ainda mais dados rotulados são necessários para treinar as redes neurais formadas por várias camadas (KIM; KO; SEO, 2020). Disso surge uma dificuldade: obter tais dados rotulados. Para as companhias que fornecem serviços *online* como sites de busca, *e-commerce* e *streamming*, dados rotulados são obtidos praticamente sem custo. Exemplo disto ocorre quando o usuário faz a classificação de um filme no serviço de *streamming* e então essa informação é aproveitada pelos sistemas de recomendação para aprimorar seu conhecimento acerca dos gostos do usuário.

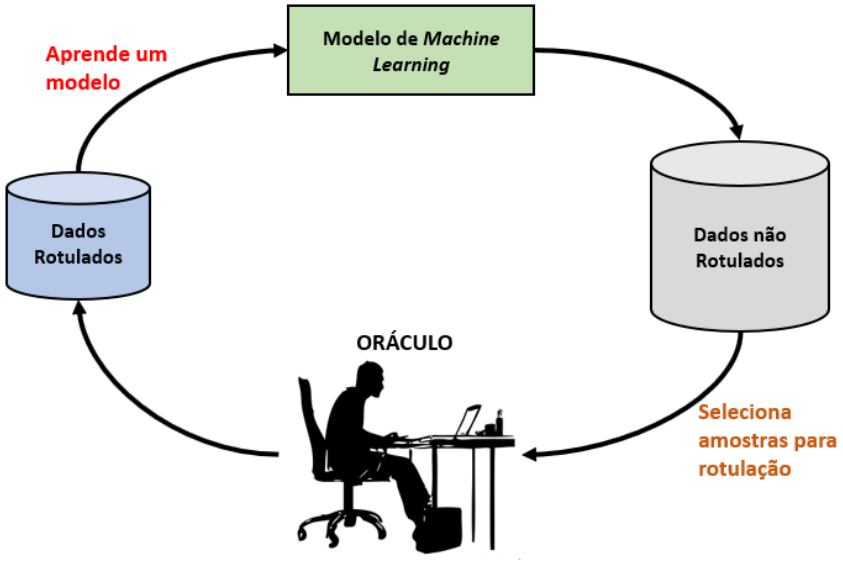
Contudo, em outros contextos a situação é diferente e obter dados rotulados se torna algo caro e demorado (MONTELEONI, 2007). Por exemplo, em aplicações médicas, os dados rotulados são produzidos por especialistas, que possuem profundo conhecimento sobre um tema em que se deseja aplicar um sistema de AM, e o trabalho de rotulação é realizado de forma manual. Assim cria-se um gargalo de tempo e mão-de-obra para o levantamento de amostras rotuladas.

A abordagem de AL possui como premissa possibilitar que o modelo “escolha” os dados que serão utilizados para a fase de treino, com o intuito de obter a melhor acurácia

na predição usando a menor quantidade de dados rotulados possível. Para isso o algoritmo faz uso de estratégias para identificar as instâncias que são mais informativas (SETTLES, 2009).

A Figura 1 ilustra o funcionamento de um modelo desse tipo, chamado na literatura de *pool-based Active Learning*.

Figura 1 – Ciclo de execução de um algoritmo de AL.



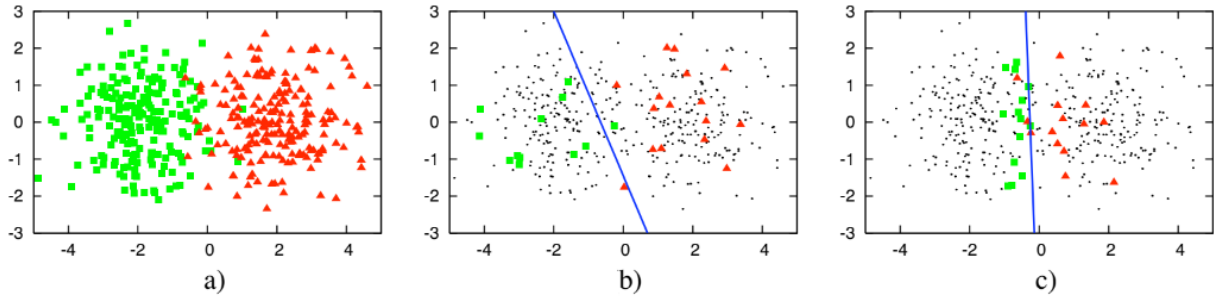
Fonte: Adaptado de Settles (2009).

Nessa implementação o algoritmo é treinado inicialmente com uma quantidade pequena de amostras rotuladas, em seguida o modelo analisa os dados não rotulados de teste e seleciona aqueles que apresentam maior ganho de informação, ao considerar por exemplo as amostras que geram maior incerteza de classificação. Então estes dados são enviados a um agente externo chamado de "oráculo" que rotula as amostras. Por fim os novos dados rotulados são acrescentados no conjunto de treino para atualização do modelo e então o processo recomeça.

Em Settles (2009) é apresentado um exemplo simples que ilustra o poder de AL. Um conjunto de dados sintético é produzido por duas gaussianas centradas em (-2,0) e (2,0), com $\sigma = 1$ onde cada uma representa uma distribuição de classe diferente. O *dataset* é composto por 400 amostras (200 de cada classe) em um espaço bidimensional como mostra a Figura 2 a). Ao se treinar um modelo supervisionado de regressão logística, utilizando 30 amostras aleatórias rotuladas deste conjunto e mantendo o restante sem rótulo para predição, se tem o resultado da Figura 2 b). Nesse caso a fronteira de decisão se encontra distante de zero no eixo horizontal que seria a posição ideal, alcançando

uma acurácia de 0,7 para o restante dos dados.

Figura 2 – Demonstração do desempenho do uso de AL em um conjunto de dados sintético. (a) Conjunto de dados de 400 amostras geradas por duas Gaussianas. (b) Fronteira de decisão obtida ao amostrar 30 objetos aleatoriamente para treinamento (acurácia de 0,7). (c) Fronteira de decisão obtida ao amostrar 30 objetos para treinamento através de AL (acurácia de 0,9).



Fonte: Settles (2009).

Por outro lado, na Figura 2 c), ao aplicar a medida de incerteza na seleção de objetos para treinamento, os 30 dados mais próximos da fronteira de decisão são escolhidos e então o novo modelo treinado com esses objetos avalia o restante do *dataset*, alcançando acurácia de 0,9. Ao evitar requisitar rótulos para instâncias redundantes ou que contribui pouco para reduzir a incerteza de classificação, o uso de AL reduziu em 67% o erro de predição, em comparação com uma amostragem aleatória, ao rotular menos de 10% dos dados.

Na literatura, a implementação de *Active Learning* em modelos de AM normalmente ocorre de acordo com os seguintes cenários:

- *Membership query synthesis:* apresentado por Angluin (1988), esse método se baseia na geração de instâncias artificiais com base na distribuição dos dados originais. De forma geral, solicita-se a rotulação das novas amostras e esses rótulos são aplicados então sobre os dados originais de onde a amostra foi gerada. Uma crítica a essa abordagem é que os dados gerados para rotulação podem não ser compreensíveis e no caso do anotador ser um humano, isso dificulta ou até impossibilita a rotulação. Entretanto, se o rótulo provém de algum processo ou agente autônomo este método pode ser interessante para construir algoritmos de descoberta automática.

- *Stream-based selective sampling:* esta forma de AL proposto por Cohn, Ladner e Waibel (1994), assume que obter dados não rotulados tem custo baixo ou até mesmo

é grátis, então é feita uma amostragem sobre os dados de forma que apenas uma instância por vez é apresentada ao modelo que deve em seguida decidir solicitar a consulta (*query*) do seu rótulo ou então descartar o dado. Essa decisão é tomada segundo uma estratégia baseada no nível de incerteza do modelo durante a classificação da instância. Esta forma de implementação é interessante em casos onde se tem recursos de memória e processamento limitados, como dispositivos móveis ou embarcados.

- *Pool-based Active Learning:* neste cenário considera-se que há uma quantidade pequena de dados rotulados L e que vários dados não rotulados podem ser coletados de uma vez, formando um conjunto grande de dados não rotulados U. De forma semelhante ao *Stream-based selective sampling*, o modelo aplica estratégias de *query* para selecionar os dados em U a serem rotulados, porém isso é feito para várias amostras de uma vez (LEWIS; GALE, 1994). Este método é o mais comum de ser utilizado.

Existem várias heurísticas que podem ser utilizadas com a finalidade de encontrar as instâncias que são mais informativas para serem posteriormente rotuladas por um "oráculo". As principais são:

- *Uncertainty Sampling* (Amostragem por incerteza): é uma das métricas mais simples e utilizadas para selecionar os objetos (LEWIS; GALE, 1994), sendo geralmente aplicada em modelos probabilísticos mas pode ser utilizada também por modelos não-probabilísticos. De forma simples essa estratégia seleciona as amostras que estão mais próximas da fronteira de decisão, que são também as instâncias que provocam maior incerteza na hora do modelo atribuir um rótulo. O cálculo da incerteza pode ser feito usando a medida de entropia (SHANNON, 1948) da Equação 1 ou então a métrica que averigua qual dado cujo rótulo predito pelo modelo é menos confiável, dado pela Equação 2 onde y^* é o rótulo mais provável de x.

$$x_{ENT} = argmax_x - \sum_i P(y_i|x) \log P(y_i|x), \tag{1}$$

$$x_{LC} = argmin_x P(y^*|x) \tag{2}$$

- *Query-By-Committee (QBC):* neste caso um *Ensemble* $C = (\theta^1, \theta^2, ..., \theta^C)$ é treinado com os dados rotulados, onde cada classificador θ^i membro do *Ensemble*, implementa uma hipótese diferente sobre os dados. Então os membros desse comitê C votam em qual instância deve ser selecionada para ser rotulada e aquela que apresentar maior discordância entre os membros é escolhida (SEUNG; OPPER; SOMPOLINSKY, 1992). É uma extensão do método de amostragem por incerteza, contudo

considerando um conjunto de modelos no processo e não apenas um. Em QBC duas métricas são muito usadas, a primeira é a entropia de votos (DAGAN; ENGELSON, 1995) dada pela Equação 3 e a segunda a divergência *Kullback-Leibler* (KL) média (MCCALLUM; NIGAM, 1998) da Equação 4. Nas equações a seguir $V(y_i)$ é o número de votos recebidos pelo rótulo y_i e $P(y_i|x;C)$ é o consenso do *Ensemble* C de que y_i é o rótulo correto.

$$x_{VE} = argmax_x - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}, \tag{3}$$

$$\begin{aligned} x_{KL} &= argmax_x - \frac{1}{C}\sum_i^C D(P_{\theta(c)}||P_C), \\ D(P_{\theta(c)}||P_C) &= \sum_i P(y_i|x;\theta^{(c)}) \log \frac{P(y_i|x;\theta^{(c)})}{P(y_i|x;C)} \end{aligned} \tag{4}$$

- *Expected Model Change* (Mudança esperada no modelo): foi proposta por Settles, Craven e Ray (2007) e seleciona a amostra que cause maior mudança na definição do modelo, se seu rótulo fosse conhecido.

- *Density Weighted Methods* (Métodos ponderados por densidade): medida apresentada em Settles e Craven (2008) que procura encontrar amostras de dados não rotulados que possuem alta incerteza e também que estejam localizadas em regiões densas do espaço de busca, visando assim evitar a seleção de *outliers* pois, apesar de apresentarem altos níveis de incerteza possuem pouca informação a respeito da distribuição dos dados.

O uso de AL tem sido tema central de muitos trabalhos nos últimos anos. Em Schumann e Rehbein (2019) é proposto uma metodologia para uso de *Membership query synthesis* em problemas de Processamento de Linguagem Natural, ao utilizar *Autoencoder* Variacional para geração das *queries*. Essa hipótese foi testada na tarefa de classificação de texto e demostrou eficiência próxima ao de *Pool-based Active Learning* ao mesmo tempo que reduziu consideravelmente o tempo para anotação dos rótulos.

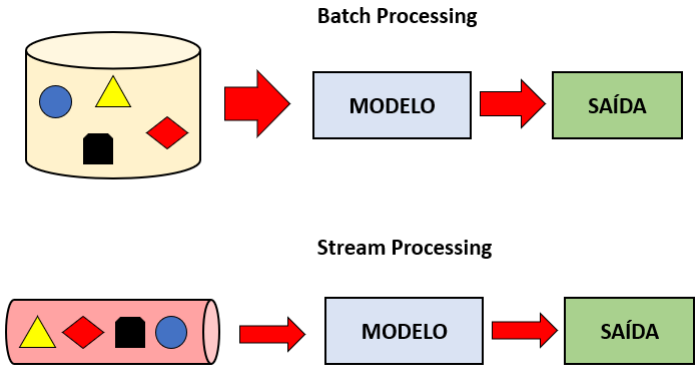
Sheikh et al. (2020) utilizam AL para reduzir o custo de obtenção de dados rotulados que são necessários para treinamento de um modelo de segmentação semântica concebido para a aplicação na agricultura, com o objetivo de discriminar ervas daninhas da lavoura a partir de imagens obtidas por plataforma robótica.

2.2 Aprendizado Incremental

Uma abordagem muito recorrente na obtenção de modelos de AM consiste em assumir que todos os dados estão disponíveis para serem acessados ao mesmo tempo na fase

de treinamento, método este chamado de processamento em lote (ou em inglês *batch processing*). Contudo tal ideia nem sempre corresponde ao que é encontrado em problemas reais. Em vários cenários mais realistas, os dados se apresentam gradualmente ao longo do tempo na forma de um fluxo de dados (*data streams*) sequenciais (HE et al., 2011). Tal situação demanda um modelo que possa formar sua base de conhecimento acompanhando o ritmo em que as novas amostras de treino se tornam disponíveis. A Figura 3 mostra a diferença entre *stream processing* e *batch processing*.

Figura 3 – Diferença entre processamento em lote e processamento em fluxo de dados.



Fonte: Elaboração própria.

Algoritmos com capacidade de lidar com *data streams* fazem parte do paradigma de *Incremental Learning* (Aprendizado Incremental). Na literatura não há um consenso sobre a definição deste conceito, alguns autores usam os termos *Online Learning* ou então *Continual Learning* como sinônimos de *Incremental Learning* (IL) (VEN; TOLIAS, 2019; LANGE et al., 2019), enquanto outros tratam *Online Learning* e IL como conceitos distintos (GEPPERTH; HAMMER, 2016; SHAHEEN et al., 2021).

Apesar das diferentes interpretações, IL é comumente entendido como um modelo dinâmico que aprende por meio de um constante *data stream* e portanto sua base de conhecimento é atualizada continuamente conforme a chegada das novas amostras (GENG; SMITH-MILES, 2009). Diferente do que ocorre com *batch learning*, o algoritmo desconhece *a priori* a quantidade total de dados de treino já que os objetos são apresentados em sequência ao longo do tempo. O intuito é induzir um modelo M_t depois de cada período de tempo, a partir das amostras atuais e do modelo anterior M_{t-1} (GEPPERTH; HAMMER, 2016).

O treinamento de um algoritmo incremental ocorre por meio de uma função de custo que é otimizada a cada amostra de treino ou então usando um pequeno conjunto de amostras por etapa (*mini-batches*). Além de ser aplicado sobre *data streams*, IL pode ser usado também quando a base de dados é muito grande (*Big Data*) e não há recurso de

memória suficiente para armazenar toda essa informação (GEPPERTH; HAMMER, 2016), sendo necessário fazer o aprendizado em *mini-batches*. Esta situação também ocorre quando a aplicação dispõe de pouco hardware de memória, como sistemas embarcados.

Os modelos de IL lidam com vários desafios (ADE; DESHMUKH, 2013), dentre eles:

- Adaptação gradual do modelo ao longo do tempo de acordo com os novos dados apenas, sem retreinar o modelo do princípio;
- Aprender com as novas amostras sem perder o conhecimento assimilado anteriormente pelo modelo, conhecido como problema de *catastrophic forgetting*;
- Dilema da plasticidade x estabilidade;

O problema de *catastrophic forgetting* foi apresentado pela primeira vez por McCloskey e Cohen (1989) ao lidar com redes neurais perceptron de múltiplas camadas e basicamente consiste no esquecimento das informações aprendidas anteriormente quando o modelo é treinado com novos dados. No caso das redes neurais, isso ocorre pois os pesos dos neurônios são alterados ao se atualizar o modelo, de tal forma que a nova informação se sobrepõe à antiga e com a mudança dos pesos dificilmente o conhecimento *a priori* é preservado (MCRAE, 1993). Os algoritmos incrementais, mesmo aqueles que não utilizam redes neurais, passam pelo mesmo problema.

Na literatura várias soluções foram propostas para superar este problema como ortogonalidade (LEWANDOWSKY; LI, 1995), regra da novidade (KORTGE, 1990), redes pré-treinadas (MCRAE, 1993), mecanismo de *rehearsal* (ROBINS, 1995), aprendizado latente (GUTSTEIN; STUMP, 2015), regularização (KIRKPATRICK et al., 2016) e mudança dinâmica da arquitetura do modelo (RUSU et al., 2016). Eliminar ou amenizar o efeito de *catastrophic forgetting* possibilita atualizar o modelo apenas com os novos dados, sem retreinar todo o modelo.

Ao lidar com problemas realistas, a distribuição dos dados é dinâmica e pode sofrer mudanças ao longo do tempo. Um exemplo de contexto onde isso ocorre são as séries temporais, como os casos de previsão do tempo e de bolsa de valores. Mudanças dessa natureza são chamadas na literatura de *concept drift* (KULKARNI; ADE, 2014). Modelos de IL podem lidar com esse dinamismo nos dados uma vez que se adaptam de acordo com as amostras recebidas. Contudo, disso surge a questão de quando e como fazer essa adaptação. Essa dúvida é conhecida como dilema da plasticidade x estabilidade.

A plasticidade vem da capacidade do modelo de se adaptar a um novo conhecimento, enquanto a estabilidade é a capacidade de reter conhecimentos passados (PARISI; LOMONACO, 2020). Um algoritmo incremental deve encontrar um equilíbrio entre estes dois requisitos. Pois uma adaptação rápida aos dados provoca também rápido esquecimento das informações já adquiridas e uma adaptação lenta preserva conhecimentos

anteriores por mais tempo, porém em contrapartida o modelo perde capacidade de reagir a mudanças no sistema (GEPPERTH; HAMMER, 2016).

Uma maneira de enfrentar este dilema é aprimorar as regras de aprendizagem, determinando como e quando aprender. Outra abordagem que é utilizada são os mecanismos de aprendizagem bio-inspirados, por exemplo, regularização sináptica, plasticidade estrutural e repetição de memória (SHAHEEN et al., 2021).

Vários algoritmos foram propostos ao longo do tempo visando proporcionar IL e superar os desafios mencionados acima. Alguns exemplos são: *Support Vector Machine* (SVM) incremental (DIEHL; CAUWENBERGHS, 2003), redes RBF (BRUZZONE; FERNÁNDEZ-PRIETO, 1999), *Ensembles* de *Random Forest* incrementais (MA; BEN-ARIE, 2014), *Incremental Classifier and Representation Learning* (iCaRL) (REBUFFI; KOLESNIKOV; LAMPERT, 2016), *Generative Adversarial Network* (GAN) (SHIN et al., 2017), *Elastic Weight Consolidation* (EWC) (KIRKPATRICK et al., 2016), *Dynamically Expandable Networks* (DEN) (LEE et al., 2017), entre outros.

Modelos de IL encontram vastas aplicações em problemas do mundo real onde se tem a necessidade de um sistema que possa aprender continuamente sem gerar alta demanda por recursos de memória e processamento. Em Shaheen et al. (2021) é explorado a aplicação em sistemas de carros autônomos, veículos aéreos-não tripulados e robôs urbanos.

2.2.1 Class-Incremental Learning

Em IL, o fluxo de dados pode ser são dividido em *tasks*, onde cada *task* pode representar diferentes conjuntos de dados segundo um determinado aspecto. Shaheen et al. (2021) categoriza três aspectos/cenários diferentes onde IL é aplicado tipicamente, os quais são descritos abaixo:

- ***Domain-Incremental Learning (Domain-IL):*** neste caso a *task* é sempre a mesma, porém, a distribuição dos dados de entrada muda ao longo do tempo (*drift*). No contexto de visão computacional, essa alteração na distribuição pode surgir por conta de ruídos ou mudanças nas condições de iluminação por exemplo. Aqui um único modelo é utilizado visando reaproveitar o conhecimento anterior na avaliação dos novos dados, mantendo o desempenho sobre *tasks* antigas enquanto apresenta resultados satisfatórios em novas *tasks*.

- ***Task-Incremental Learning (Task-IL):*** neste cenário cada *task* diz respeito a um *dataset* diferente contendo um conjunto de dimensões também diferentes, definindo cada qual um novo problema. Por exemplo, uma *task* antiga pode se tratar de um problema de classificação com 5 classes enquanto a nova *task* se refere a um problema de regressão. Em outro caso pode-se ter um sistema de detecção de veículos que precisa ser adaptado para detectar também pedestres, prédios, árvores

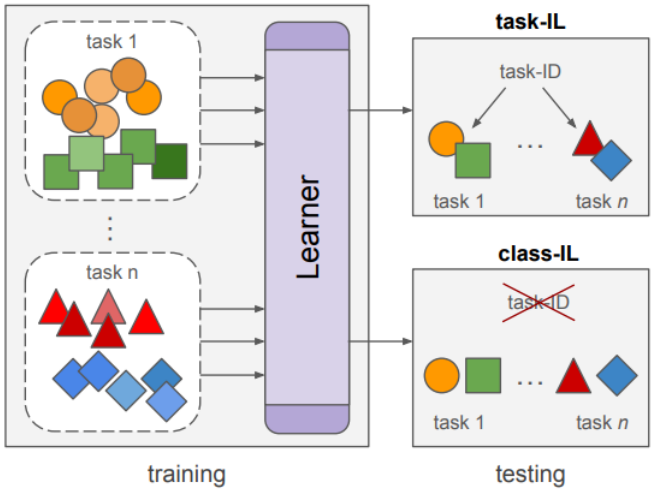
e cada uma dessas novas funcionalidades são definidas como *tasks* que o modelo deve aprender (HOSSAIN et al., 2022). Diferente de *Class-IL*, aqui na etapa de teste é fornecido um identificador para cada *task*, orientando o modelo sobre qual tarefa deve ser executada.

- ***Class-Incremental Learning (Class-IL):*** este tipo de problema lida com classificação multi-classe e tem como objetivo aprender incrementalmente novas classes. Aqui cada *task* inclui um conjunto diferente de novas classes ao problema enquanto os dados de categorias já conhecidas se tornam indisponíveis ao modelo. O algoritmo treinado deve ser capaz de inferir sobre as classes atuais e antigas por conta própria.

Neste trabalho, busca-se desenvolver um modelo com objetivo semelhante ao *Class-IL*. Os estudos na área de IL envolvendo DL teve foco inicial em problemas de *Task-IL* (MASANA et al., 2020), contudo essa tendência vem mudando e vários algoritmos vem sendo propostos para *Class-IL*, que é um cenário mais complexo e desafiador que *Task-IL*.

Em *Task-IL* o acesso a um identificador de *task* na etapa de inferência traz a vantagem de que o modelo não precisa aprender a discriminar entre as classes vindas de diferentes *tasks*. O acesso a tal identificador não existe em *Class-IL* e, portanto, o algoritmo precisa aprender a distinguir entre as classes vindas de todas as *tasks* (MASANA et al., 2020). A Figura 4 ilustra a diferença entre *Task-IL* e *Class-IL*.

Figura 4 – Em *Task-IL* a etapa de predição conta com um identificador de *task* (*task*-ID) enquanto que *Class-IL* não faz uso desse artifício.



Fonte: Masana et al. (2020).

Algoritmos de *Class-IL* processam dados vindos de distribuições não-estacionárias, que é o caso típico das bases de dados de problemas do mundo real, onde as amostras

não são *i.i.d.* Modelos deste tipo devem ser capazes de escalonar para um grande número de *tasks* sem aumentar excessivamente o uso de memória e custo de processamento. O intuito é tirar proveito de informações das classes já conhecidas para melhorar a aprendizagem das novas classes e também aprender com os novos dados para melhorar o desempenho sobre *tasks* antigas (LOPEZ-PAZ; RANZATO, 2017).

Class-IL enfrenta os mesmos problemas de outros algoritmos incrementais (dilema plasticidade x estabilidade e *catastrophic forgetting*) e ao tentar solucionar o problema de *catastrophic forgetting* outro surge: intransigência (resistência ao aprendizado de uma nova classe) (CHAUDHRY et al., 2018). Para lidar com este cenário, muitos dos modelos de *Class-IL* utilizam DL como extrator de características (*features*) para alimentar classificadores de construção mais simples, como é o caso de iCaRL que aplica ResNet para obter *features* e um modelo baseado em *Nearest Class Mean* (NCM) como classificador (REBUFFI; KOLESNIKOV; LAMPERT, 2016).

Existe para linguagem Python um *toolbox open source* que fornece vários algoritmos de *Class-IL* chamado PyCIL[1], que facilita o desenvolvimento de aplicações para o mundo real onde é necessário aprender continuamente novas classes.

2.3 Detecção de novidades

Desenvolver algoritmos capazes de identificar padrões desconhecidos nos dados não é uma preocupação recente na área de AM, visto que é possível encontrar pesquisas de 20 anos atrás que já abordavam o assunto (DIEHL; HAMPSHIRE, 2002). Contudo, ainda hoje a pesquisa na área segue ativa, sendo tema central de vários estudos (TACK et al., 2020; YANG et al., 2021).

A forma comum de gerar um modelo de AM segue a premissa de que se conhece *a priori* todos os possíveis padrões existentes nos dados de um determinado problema, de maneira que este conhecimento é usado, então, como referência ao inferir/generalizar sobre novos dados. Em ambientes controlados, como um laboratório, tal abordagem funciona muito bem, porém, este modelo ao ser inserido no mundo real apresenta dificuldades em lidar com a imprevisibilidade que o cenário traz consigo (SUN et al., 2020; MILLER et al., 2020).

Assim, para muitas situações, é importante que o modelo seja capaz de detectar mudanças nos dados. Em soluções *data-driven*, nas quais decisões estratégicas são tomadas com base na análise de dados, é fundamental, por exemplo, que novidades nos dados sejam detectáveis ao longo do tempo, uma vez que tais informações servem de subsídio para a melhoria e adaptação do modelo.

Apesar do grande volume de estudos envolvendo detecção de novidade, não existe uma definição universalmente aceita para o que é uma novidade. Muitos autores (CRUPI;

[1] <https://github.com/G-U-N/PyCIL>

GUGLIELMINO; MILAZZO, 2004; MILJKOVIć, 2010; PARK; HUANG; DING, 2010) fornecem uma definição genérica para o tema considerando uma novidade como aqueles dados ou sinais que não seguem o comportamento definido como normal sendo, portanto, algo desconhecido, ou então, são objetos que não podem ser explicados usando o conhecimento anterior do modelo obtido durante a fase de treinamento.

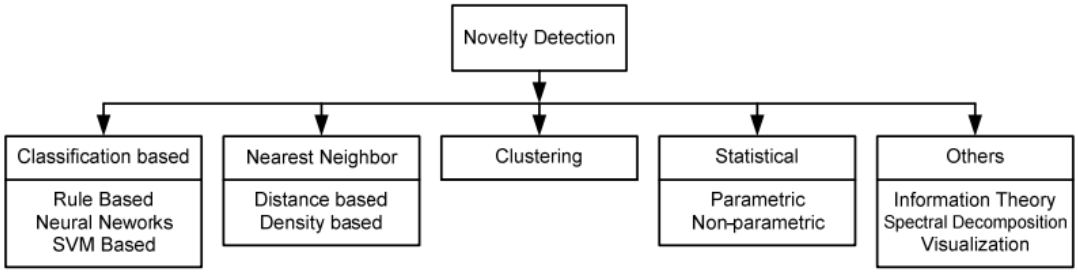
Este conceito de novidade é muito comum na literatura e com frequência gera confusão com outras terminologias como detecção de *outliers* ou anomalias. Alguns trabalhos usam esses termos como sinônimos ou então tratam a detecção de novidades como uma forma de detecção de *outliers* (MILJKOVIć, 2010; YANG et al., 2022; ASAVALERTPALAKORN; SINGHATANADGID; ARDSOMANG, 2022). Neste caso, as novidades ou dados anômalos ocorrem raramente nos dados e geralmente usa-se uma abordagem *one-class*, onde todos os dados de treino representam uma única classe normal, e o intuito é identificar os objetos no conjunto de teste que não seguem o padrão observado durante o treinamento.

Outros autores preferem separar o conceito de detecção de anomalias e novidades, inclusive destacando a diferença com outras linhas de pesquisa semelhantes como *Open Set Recognition* e *Out-of-Distribution* (OOD) (SALEHI et al., 2021). Neste caso, em específico, é tido que a detecção de anomalias utiliza dados obtidos sem qualquer uso de filtros, i.e, o conjunto de treino pode conter ruídos e dados anormais, enquanto que no caso da novidade todas as amostras de treino descrevem o comportamento normal da aplicação.

Em certos problemas, a novidade é tida como um objeto originado por uma mudança no mecanismo gerador dos dados, por exemplo, mudanças no ambiente, como condições de iluminação, desaparecimento de classes conhecidas e surgimento de novas, de tal maneira que o conhecimento atual da distribuição dos dados deve ser atualizado (KUNCHEVA, 2008). Disso surgem os conceitos de *concept-drift* e *concept-evolution* (MASUD et al., 2010) em que o primeiro se refere a uma mudança na distribuição dos objetos mas sem alteração das classes conhecidas, enquanto que o segundo trata de uma distorção mais forte na distribuição devido ao surgimento de novas classes. No artigo de Miljković (2010) é feita uma revisão sobre o tema, apresentando uma breve explicação dos métodos que podem ser utilizados para realizar a detecção de novidades. O autor organiza estes métodos como é mostrado na Figura 5.

Para Salehi et al. (2021), as novidades nos dados são encontradas aplicando técnicas discriminativas ou generativas. No primeiro caso, algoritmos de classificação ou *Self-supervised learning* podem ser usados para discriminar padrões anormais daqueles que são conhecidos. Para a abordagem generativa emprega-se *Autoencoder* (AE) ou *Generative Adversarial Network* (GAN) por meio de redes neurais. Ao aplicar AE é verificado o erro de reconstrução sobre as novas amostras. Se o novo objeto remete ao comportamento normal dos dados, então o erro de reconstrução é baixo, caso a nova amostra represente um padrão diferente do conhecido, espera-se que o erro de reconstrução seja

Figura 5 – Categorias de métodos para detecção de novidades.



Fonte: Miljković (2010).

alto. Entretanto estudos recentes (SALEHI et al., 2020; ZAHEER et al., 2020) mostram que esta abordagem pode gerar alto erro de reconstrução em amostras conhecidas se estas apresentarem ruído ou algum tipo de distorção. Quando se utiliza GAN, assume-se que para objetos conhecidos o modelo é capaz de encontrar um vetor latente que gera uma saída com baixa discrepância com a entrada e o oposto ocorre caso uma novidade apareça. Apesar de mostrar bons resultados em alguns problemas, o uso de GANs apresenta algumas dificuldades como processo de treinamento instável e resultados irreprodutíveis (ARJOVSKY; BOTTOU, 2017).

Mesmo com a definição dessas categorias de métodos para detecção de novidades, a forma usual de desenvolver aplicações deste tipo continua sendo projetar o modelo de acordo com o conjunto de dados e as especificações do problema. Como exemplo, Buono et al. (2022) desenvolve um algoritmo para prever falhas em máquinas no contexto industrial como parte da estratégia de manutenção preditiva, buscando minimizar paradas inesperadas na produção. Para isso, aplica AE para encontrar as anomalias nos dados com base no erro de reconstrução das amostras, incluindo uma camada para fazer a discriminação entre as condições de operação encontradas e um localizador que encontra os componentes de sinais mais influentes na aparição da novidade, permitindo explicar sua ocorrência. O AE proposto é testado usando quatro modelos diferentes de DL.

O estudo recente de Boult et al. (2021) procurou formalizar o entendimento de novidade, diante de diferentes contextos, oferecendo um *framework* para elaboração de modelos que lidam com esta questão. Para isso os autores elencam os elementos básicos que envolvem qualquer aplicação na área e, então, determinam os tipos de novidade que são possíveis de acordo com a maneira como esses elementos interagem entre si em um dado problema. Segundo os autores, a estrutura proposta pode ser aplicada desde IA simbólica e aprendizado por reforço, até contextos mais complexos como *Open-Set* e *Open World*, que se relacionam com o tema deste trabalho e serão abordados nas seções seguintes.

2.3.1 Open-Set

Considere os *datasets* MNIST[2] (LECUN; CORTES; BURGES, 2010) e *Labelled Faces in the Wild* (LFW)[3] já bem conhecidos na literatura. O primeiro diz respeito a um conjunto de imagens de dígitos manuscritos de 0 a 9, enquanto o segundo é composto por várias imagens de rostos para desenvolvimento de modelos de reconhecimento facial. O conteúdo dessas bases de dados é mostrado na Figura 6.

Figura 6 – Amostras de imagens que compõe as bases de dados MNIST (à esquerda) e LFW (à direita).

Fonte: LeCun, Cortes e Burges (2010) e Huang et al. (2007).

Esses *datasets* representam dois problemas frequentes em visão computacional, a classificação e o reconhecimento. Na classificação busca-se obter um modelo capaz de discriminar os objetos entre uma coleção de classes bem definidas *a priori*, como por exemplo identificar o valor de um dígito em uma imagem (MNIST). Por outro lado, a tarefa de reconhecimento de objetos, em termos gerais, visa identificar rótulos específicos definidos para instâncias de uma mesma categoria, tratando estes rótulos como classes do problema. Uma aplicação típica dessa abordagem é em reconhecimento facial (LFW).

Na classificação a tarefa é definida sob um conjunto finito de classes, onde considera-se que todas as categorias presentes no conjunto de teste são conhecidas no conjunto de treino. Tal premissa é chamada de Cenário Fechado, do inglês *Closed-Set*, e gera bons resultados para ambientes controlados onde se tem garantia de que apenas classes conhecidas serão apresentadas ao modelo, porém é ineficaz em contextos realistas onde padrões não previstos podem emergir dos dados. Na tarefa de reconhecimento também é aplicado o princípio de *Closed-Set*, no entanto modelos desse tipo apresentam uma abertura ainda maior ao aparecimento de classes novas (SCHEIRER et al., 2013).

[2] <http://yann.lecun.com/exdb/mnist/>
[3] <https://www.kaggle.com/datasets/jessicali9530/lfw-dataset>

A principal crítica a abordagem de Cenário Fechado é que o classificador geralmente aplica fronteiras de decisão que divide o espaço de busca em um número fixo de regiões, relativo a cada classe conhecida. Dessa forma uma categoria desconhecida *a priori* é obrigatoriamente alocada em um destes rótulos, o que gera inconsistências nos resultados de predição, diminuindo a confiabilidade do modelo. Um algoritmo que lide com contextos do mundo real precisa ter a habilidade de identificar os padrões que não seguem o perfil da base de conhecimento do modelo, mantendo estes objetos em uma região separada daquelas onde residem as classes previamente conhecidas.

Modelos com tal capacidade foram definidos pela primeira vez no artigo de Scheirer et al. (2013), onde aplicações que demandam esse tipo de implementação receberam a denominação de problemas *Open-Set* (Cenário Aberto). O artigo também fornece uma forma de calcular o nível de abertura de um problema ou *openness*, descrito pela Equação 5, que mensura o quão forte é a presença de classes desconhecidas na tarefa de AM com base no número de classes usadas no treinamento (número de classes de treino), no teste (número de classes de teste) e no número de classes alvo a serem identificadas (número de classes alvo).

$$openness = 1 - \sqrt{\frac{2 \times |\text{num. classes de treino}|}{|\text{num. classes de teste}| + |\text{num. de classes alvo}|}} \tag{5}$$

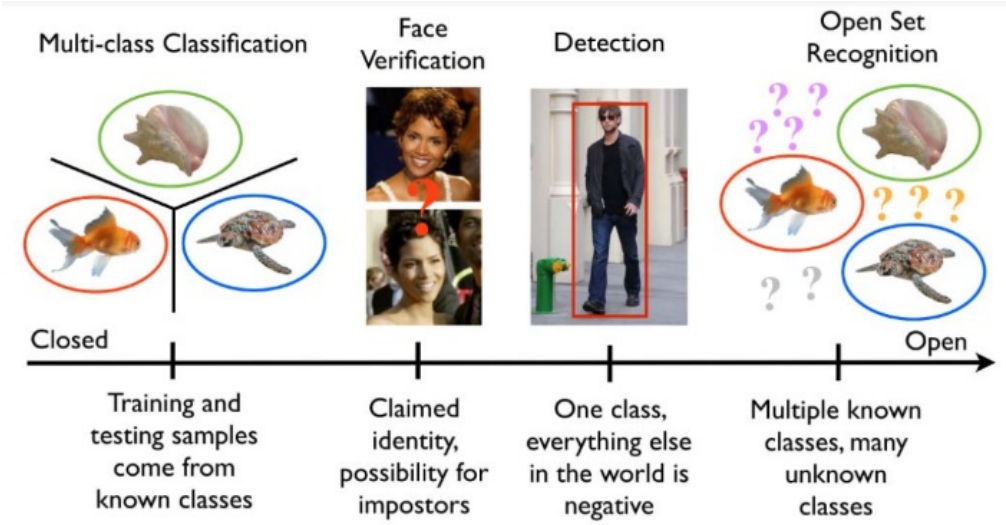
Assim, para uma quantidade fixa de classes de treinamento, quanto maior for a quantidade de classes alvo ou de teste, maior o valor de abertura resultante enquanto que um número maior de classes de treinamento, sobre os outros conjuntos, reduz o valor da medida. A medida de *openness* varia de 0% a 100%, onde o valor mínimo denota um problema *Closed-Set* e valores maiores tarefas mais abertas a novas classes. A Figura 7 exibe a abertura de alguns problemas de visão computacional.

No final dessa escala se tem os problemas onde uma quantidade considerável de categorias desconhecidas fazem parte do contexto da tarefa. Assim, o conceito de *Open-Set* designa um cenário mais realista de aprendizado, no qual o conhecimento incompleto do mundo está presente durante o treinamento do modelo e classes desconhecidas podem aparecer com o tempo para serem classificadas.

A Figura 8 ilustra este cenário contrastando com uma situação mais restritiva de *Closed-Set*, o qual é sensível a novas classes que possam emergir com o tempo.

Scheirer et al. (2013) definem formalmente o problema de Reconhecimento em Cenário Aberto ou *Open Set Recognition* assumindo amostras $\hat{V} = \{v_1, ..., v_m\}$ de um conjunto P como sendo os dados positivos de treino e as amostras $\hat{K} = \{k_1, ..., k_n\}$ de outras K classes os dados negativos. Seja U o universo de classes desconhecidas (negativas) que aparecem apenas na fase de teste e $T = \{t_1, ..., t_z\}, t_i \in P \cup K \cup U$ o conjunto de dados de teste com valor de *openness* maior que 0. Dado um conjunto de treino $\hat{V} \cup \hat{K}$, uma função de risco de espaço aberto R_O e uma função de risco empírico R_ε, a aplicação de *Open*

Figura 7 – Problemas de visão computacional em ordem crescente do valor de *openness*. Para alguns casos não é possível ter conhecimento de todas as classes durante o treino e deve-se então esperar que classes desconhecidas possam surgir ao longo da etapa de teste.



Fonte: Scheirer et al. (2013).

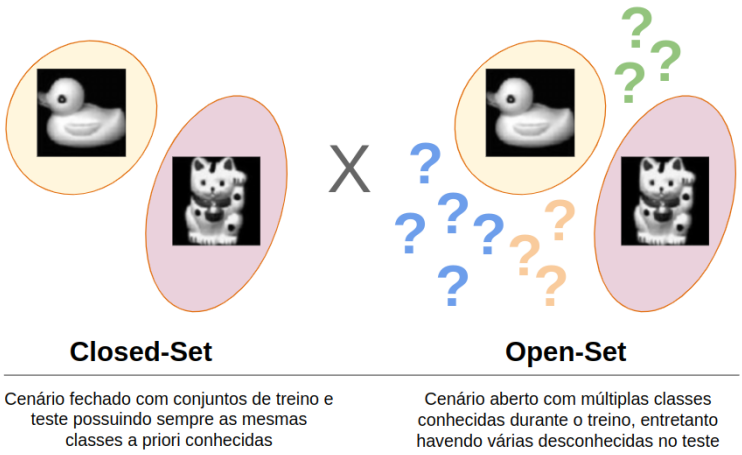
Set Recognition deve gerar um modelo f que minimize a função de custo da Equação 6 chamada de risco de cenário aberto (do inglês *Open Set Risk*), onde λ_r é uma constante de regularização.

$$\underset{f}{\operatorname{argmin}}\{R_O(f) + \lambda_r R_\varepsilon(f(\hat{V} \cup \hat{K}))\} \tag{6}$$

De forma simplificada, a indução do modelo em um contexto de *Open-Set* está atrelada a minimização de uma função de custo composta por um termo de erro convencional utilizado na etapa de treinamento, R_ε, e por um termo R_O que representa o risco de uma amostra residir em uma região do espaço de busca que é desconhecida pelo classificador. Segundo os autores, o termo R_O pode ser definido matematicamente de diferentes formas.

A avaliação de um algoritmo dessa natureza segue lógica semelhante ao que é feito em modelos *Closed-Set*. Na avaliação de modelos convencionais instâncias desconhecidas pelo modelo durante o treino são apresentadas na fase de teste, para verificar o desempenho em dados novos. Para o caso de *Open-Set* é necessário o uso de amostras de classes desconhecidas para avaliação do modelo, dessa forma algumas classes conhecidas podem ser excluídas da etapa de treino, sendo apresentadas apenas durante a fase de teste. Esse conjunto de teste deve ter *openness* > 0, idealmente assumindo diferentes valores de *openness*. Assim um algoritmo de *Open-Set* deve ser capaz de identificar um

Figura 8 – Comparação entre o *Cenário Fechado* e o *Cenário Aberto*. Enquanto o primeiro assume que novas classes não irão aparecer subsequente ao treinamento do modelo, o outro aprende algumas classes na etapa de treino, mas é consciente de que classes nunca vistas possam surgir sendo robusto a esta situação.



Fonte: Elaboração própria.

dado como pertencente a uma classe conhecida ou então atribuir o rótulo “desconhecido” para a amostra. Deve então ser capaz de rejeitar um objeto.

Vários problemas dinâmicos do mundo real, onde não se tem conhecimento de todo o conjunto de classes possíveis durante o treinamento, podem se beneficiar de modelos mais flexíveis como este. Tais modelos poderiam atuar com maior sucesso, por exemplo, em cenários como:

- De redes de câmeras de vigilância, onde atividades diferentes daquelas conhecidas pelo seu sistema de reconhecimento podem facilmente ocorrer;
- De sensoriamento remoto, no qual naturalmente podem surgir objetos em solo distintos daqueles conhecidos pela ferramenta de segmentação;
- De monitoramento de máquinas e equipamentos, em que um sistema de controle pode, com facilidade, se deparar com sinais anômalos de falhas não previstas pelo modelo aprendido;
- De uso de *wearables* e dispositivos móveis para analisar o comportamento, hábitos e ações de seus usuários onde facilmente podem ocorrer mudanças de rotina e situações não previstas que serão novidades para o modelo de reconhecimento treinado.

Scheirer et al. (2013) abordam cenários abertos usando a técnica *one-vs-set*, que é uma extensão do *one-class* SVM para problemas binários, no qual um plano adicional

é usado para otimizar o risco empírico e de espaço aberto. Ao empregar este plano, o classificador limita a região positiva evitando estender os limites de decisão para além das regiões negativas e desconhecidas. Em Scheirer, Jain e Boult (2014), os autores estenderam a ideia para classificadores não lineares em um ambiente multi-classe. Estes trabalhos, então, conseguem reconhecer não apenas objetos de classes conhecidas, mas também aqueles desconhecidos com base em limiares ou fronteiras de decisão otimizadas no próprio processo de aprendizado do modelo (JÚNIOR et al., 2021).

Em um trabalho mais abrangente, Bendale e Boult (2015) estendem a ideia de reconhecimento em cenário aberto tornando-a escalável com a categorização de objetos marcados como desconhecidos em um procedimento de aprendizado incremental, no qual o classificador não precisa ser retreinado a todo momento em que uma nova classe surge. Os mesmo autores se preocuparam em estudar no contexto das Redes Neurais Profundas meios de se evitar que o modelo cometa enganos classificando imagens que não fazem sentido como sendo pertencentes a uma classe do problema. Para isso, introduzem uma nova camada na rede, *OpenMax*, capaz de estimar a probabilidade de um novo objeto ser categorizado como desconhecido para não afetar o desempenho de classificação para as classes de importância para o problema que está sendo tratado (BENDALE; BOULT, 2016).

A próxima seção abordará a categoria de problemas apresentado por Bendale e Boult (2015) onde este trabalho se insere.

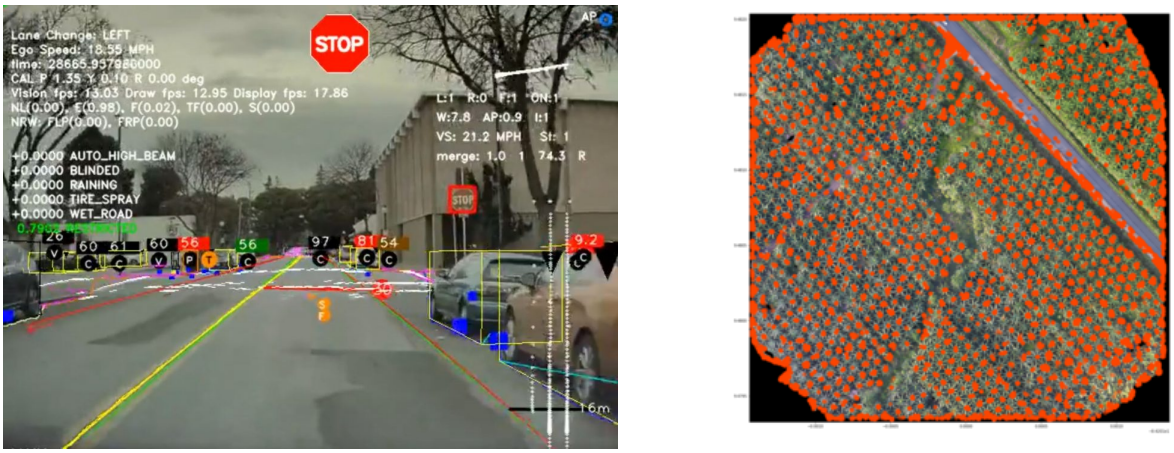
2.3.2 Open World Recognition

O ser humano tem talento natural para identificar objetos/fenômenos desconhecidos e depois aprender sua definição quando o conhecimento necessário para isso se torna disponível (JOSEPH et al., 2021). Algoritmos convencionais de AM não conseguem reproduzir essa habilidade. Dessa forma, como já foi dito na Seção 2.3.1, esses modelos tem boa atuação em ambientes controlados, como o de laboratório, porém várias dificuldades são encontradas ao implementar tais soluções no mundo real onde os dados se comportam frequentemente de forma inesperada.

Assim, em problemas realistas, a base de dados tem comportamento dinâmico e mudanças sempre ocorrerão na distribuição das amostras. Um algoritmo capaz de acompanhar esse fluxo se adapta melhor a tarefas do mundo real e atende as demandas operacionais de muitos problemas práticos que não são satisfeitas por modelos convencionais de AM. Algumas das áreas que poderiam tirar vantagem de um sistema desse tipo são:

- Robótica: atualmente robôs tem sido desenvolvidos para as mais variadas tarefas e em muitas delas o robô interage com ambientes diversos, como aqueles usados em missões de resgate ou então o caso dos carros autônomos. Em aplicações como

Figura 9 – Na esquerda um sistema de visão computacional aplicado em carros autônomos da Tesla e na direta um modelo de detecção aplicado no contexto agrícola. Os dois casos precisam lidar com um ambiente que pode apresentar padrões não previstos para continuar cumprindo sua finalidade.



a) b)

Fonte: Carscoops (2020) e Gidahatar (2020).

essas, não é possível definir precisamente o domínio do problema *a priori* e o sistema deve ser capaz de se adaptar em tempo real aos variados cenários para atingir a versatilidade esperada (Figura 9 a)).

- Indústria: no contexto industrial, um algoritmo que analisa os sinais de operação de uma máquina conseguiria identificar padrões que indicam falha na operação do sistema e ao se adaptar a esses eventos, o modelo poderia fornecer a descrição do provável defeito e sua causa, agilizando a manutenção para reduzir o tempo de parada da linha de produção.

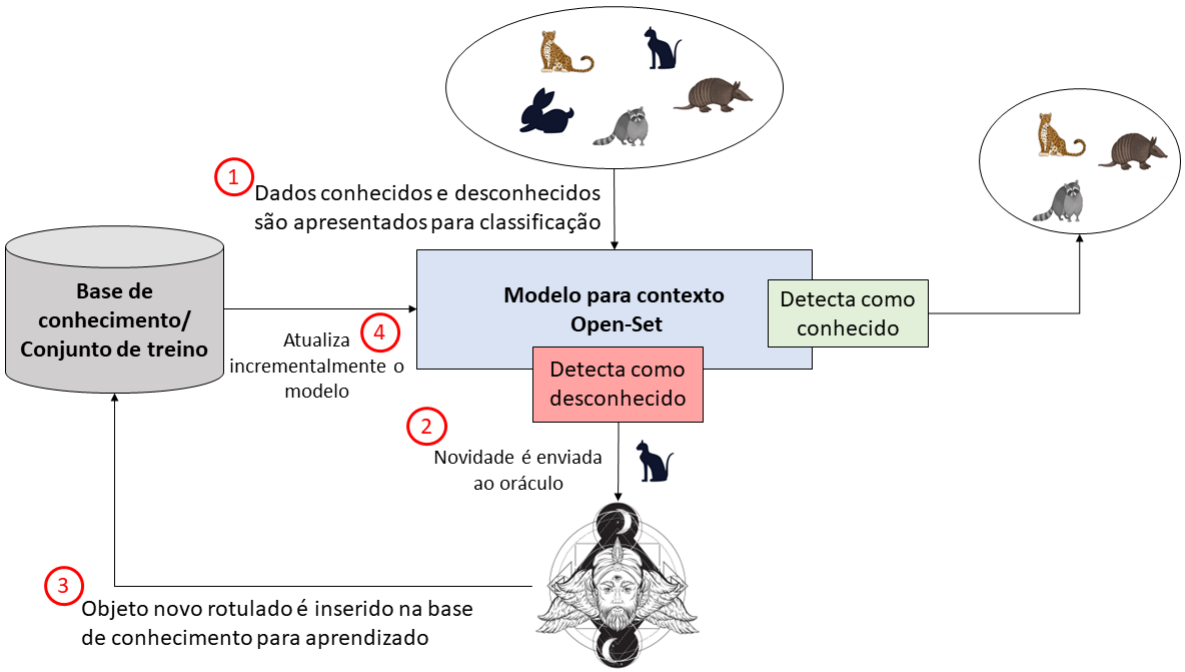
- Agricultura: no ambiente rural mudanças naturais ocorrem nas lavouras, devido a alteração da fase de crescimento da planta, mudança de estação ou até mesmo aparecimento de novas ameaças. Um sistema de monitoramento neste contexto deveria, idealmente, ser capaz de acompanhar essas alterações ao longo do tempo de vida do plantio a fim de manter seu desempenho (Figura 9 b)).

Na categoria de problemas *Open-Set*, o modelo é induzido considerando que há conhecimento incompleto do mundo, mas seu algoritmo é estático no sentido de que não aprimora conhecimento, mantendo sempre a mesma quantidade de classes conhecidas no conjunto de treino apesar da detecção de potenciais novos padrões (rotulados como "desconhedos" durante a predição). O conceito de *Open World Recognition* ou Reconhecimento em Cenário Aberto tem como objetivo superar essa limitação do *Open-Set* para

lidar com cenários mais realistas. Essa ideia foi definida pela primeira vez por Bendale e Boult (2015) e expande a abordagem de *Open-Set* com o intuito de permitir que o modelo seja escalável ao ser capaz de aprender a reconhecer novas classes durante a fase de teste.

Neste novo paradigma, inicialmente o algoritmo executa um processo de *Open-Set* onde o modelo é capaz de associar os objetos de entrada a uma classe conhecida ou então rotulá-lo como "desconhecido". Os dados classificados como "desconhecidos" são transferidos para um agente externo, como um ser humano, que define seu real rótulo. Posteriormente quando houver uma quantidade suficiente de novidades rotuladas, o algoritmo deve aprender as novas classes de forma incremental expandindo o rol de categorias conhecidas. Esse processo ocorre continuamente como um ciclo. Um sistema *Open World* é então não apenas robusto a classes desconhecidas, mas é também capaz de atualizar incrementalmente o modelo ao incorporar as novas classes. A Figura 10 ilustra o processo descrito anteriormente destacando seus quatro passos essenciais.

Figura 10 – Esquema ilustrativo de um sistema *Open World*



Fonte: Elaboração própria.

Em Bendale e Boult (2015) é destacado que um modelo *Open World* deve satisfazer dois requisitos. O primeiro é a capacidade de detectar continuamente amostras de novas classes para serem rotuladas externamente por um oráculo. O segundo requi-

sito diz respeito ao aprendizado das novas classes. Retreinar o modelo desde o início para incorporar as novas classes pode ter desempenho aceitável em conjuntos pequenos de dados/categorias, mas quando essa quantidade aumenta, essa estratégia apresenta custo proibitivo impedindo a escalabilidade do algoritmo. Portanto um sistema *Open World* precisa aprender as novas classes sem retreinar todo o modelo.

Desenvolver um modelo assim não é algo trivial, exigindo uma forte generalização (JOSEPH et al., 2021) além de impossibilitar, da mesma forma que o problema *Open-Set*, o uso da Lei de Probalibilidade Total do Teorema de Bayes para estimar a probabilidade de ocorrência das classes (SCHEIRER; JAIN; BOULT, 2014), pois se faz necessário o conhecimento prévio de todas as categorias que compõe o problema. Assim em *Open World Recognition* é necessário uma definição do valor de probabilidade de classe que decresce a medida que novos padrões são encontrados e novas classes são aprendidas (SCHEIRER et al., 2013).

O problema *Open World* é relativamente recente e possui poucos trabalhos na literatura abordando o tema. Em Bendale e Boult (2015) é apresentada a definição formal do problema propondo um algoritmo de reconhecimento para um modelo *Open World*, além de estabelecer um protocolo de avaliação para modelos deste tipo. O algoritmo desenvolvido no trabalho tem inspiração no classificador baseado em distância *Nearest Class Mean* (NCM) e se chama *Nearest Non-Outlier* (NNO) definido para balancear o Risco de Espaço Aberto (do inglês *Open Space Risk*), que se refere ao termo R_O da Equação 6, e a acurácia.

Joseph et al. (2021) aplica o princípio de *Open World* na tarefa de detecção de objetos, chamando essa nova abordagem de *Open World Object Detection*. No problema de detecção de objetos, o modelo é ensinado geralmente a considerar qualquer instância de classe desconhecida como sendo *background*, assim aprender novas classes nesse contexto é mais complexo do que em Bendale e Boult (2015), já que deve ocorrer uma diferenciação entre *background* e classes desconhecidas. O modelo *Open World Object Detector* (ORE) é proposto para esta tarefa com base em *contrastive clustering* e na formulação de Energia livre de Helmholtz da termodinâmica, como uma medida de identificação de objetos desconhecidos. Uma comparação com detectores incrementais de objetos do estado da arte mostrou que o algoritmo ORE é mais apropriado para *Open World Object Detection*.

Em Fontanel et al. (2021) modelos *Open World* propostos na literatura são testados sob um cenário onde a distribuição das classes conhecidas no treinamento sofre mudanças durante a etapa de teste (*domain-shift*). Nesse trabalho uma comparação é feita entre os algoritmos NNO (BENDALE; BOULT, 2015), *Deep Nearest Non-Outlier* (DeepNNO) (MANCINI et al., 2019) e *Boosting Deep OWR by Clustering* (B-DOC) (FONTANEL et al., 2020), os resultados dos testes mostraram que os modelos existentes de *Open World* não conseguem se adaptar a ocorrência de *domain-shift*, sofrendo gravemente do problema de esquecimento dos conceitos aprendidos. Assim, o problema ainda está longe de ser

resolvido e continua em aberto.

2.3.3 Alternativas de algoritmos Open-World

Nesta seção são expostos os algoritmos empregados na parte experimental desta pesquisa, especificamente para análise da metodologia proposta na Seção 3.1. Os algoritmos IC-EDS e NNO aplicados para testes são explicados nas Seções 2.3.3.1 e 2.3.3.2, enquanto que o modelo iCaRL empregado na avaliação do *framework* em contexto *Open World* é discutido na Seção 2.3.3.3.

2.3.3.1 Algoritmo IC-EDS

No artigo de Coletta et al. (2019) é introduzido um classificador semi-supervisionado para identificar e aprender novas classes. Este algoritmo utiliza um *ensemble* que combina informações de modelos supervisionados e não-supervisionados para executar a classificação sobre dados não rotulados de teste.

De maneira mais formal, seja $U = \{x_i\}_{i=1}^{n}$ o conjunto de n amostras não rotuladas a serem classificadas pelo *ensemble* que recebe como entradas um conjunto $\{\pi_i\}_{i=1}^{n}$ em que cada π_i representa um vetor de distribuição de probabilidades de classes de tamanho c igual ao número de classes do problema. Estes vetores são gerados por um ou mais classificadores supervisionados, como o SVM. Além disso, o *ensemble* recebe também uma matriz de similaridade $S = \{s_{ij}\}_{i,j=1}^{n}$ fornecida por algum método de agrupamento (ex: *K-means*), contendo a informação de similaridade entre cada par possível de objetos do conjunto U.

A premissa é que fazer uso de informações sobre a estrutura dos dados pode aprimorar o desempenho de classificação do modelo. Com isso, os vetores $\{\pi_i\}_{i=1}^{n}$ são atualizados com informações provenientes de S resultando no novo conjunto de vetores $\{y_i\}_{i=1}^{n}$ de distribuição de probabilidade de classe. Esse processo ocorre visando minimizar a função de custo J da Equação 7 e os vetores $\{y_i\}_{i=1}^{n}$ são refinados iterativamente de acordo com a Equação 8.

$$J = \frac{1}{2}\sum_{i\in U} ||y_i - \pi_i||^2 + \alpha\frac{1}{2}\sum_{(i,j)\in U} s_{ij}||y_i - y_j||^2 \tag{7}$$

$$y_i = \frac{\pi_i + 2\alpha\sum_{i\neq j} s_{ij}y_j}{1 + 2\alpha\sum_{i\neq j} s_{ij}} \tag{8}$$

O coeficiente α regula a importância da informação supervisionada e não-supervisionada ao longo das iterações. Segundo os autores, um bom valor para α se encontra no intervalo [0,1]. A partir disso a classificação de novas instâncias recebe restrições adicionais vindas do termo não-supervisionado, assumindo que amostras parecidas no conjunto U são mais prováveis de pertencerem a uma mesma classe.

Durante o processo descrito anteriormente, as informações de y_i e s_{ij} podem ser aproveitadas para identificar novos padrões nos dados. Em Coletta et al. (2022) isto é feito aplicando a medida de entropia e_i sobre y_i para encontrar amostras que causem maior incerteza de classificação e ao calcular a densidade d_i a partir de s_{ij} visando identificar o nível de representatividade das amostras em ser uma nova classe.

Em *Active Learning* a entropia é utilizada para detectar os objetos que estão mais próximos da fronteira de decisão para serem posteriormente rotulados. No artigo, a entropia foi empregada para determinar o nível de incerteza de um vetor de distribuição de probabilidades de classe, fornecidos pelo classificador. A entropia e_i de um objeto i dos dados é obtida pela Equação 9, onde y_{ij} é a probabilidade da amostra i de pertencer à classe j em que $j \in \{1,...,c\}$.

$$e_i = \frac{-\sum_{j=1}^{c} y_{ij} \log_2 y_{ij}}{\log_2 c} \tag{9}$$

A hipótese é que diante de dados fora dos padrões conhecidos, o valor de entropia é alto e, portanto, as amostras que demonstram maior incerteza de classificação apresentam traços diferentes dos conhecidos pelo modelo e são fortes candidatos a representarem uma nova classe.

Partindo da premissa de que um objeto é representativo de uma classe se sua vizinhança possuir alta densidade, a medida de densidade é calculada de acordo com a Equação 10, onde d_i é o valor da densidade do objeto i, V é o conjunto dos v-vizinhos mais próximos e s_{ij} representa um valor de distância ou similaridade entre a amostra i e sua vizinha j. Dessa forma, os objetos candidatos a nova classe que apresentam maiores valores para d_i são os mais representativos de um novo padrão.

$$d_i = \frac{1}{v} \sum_{j \in V} s_{ij} \tag{10}$$

Em Coletta et al. (2022), os objetos do conjunto U que indicam aparecimento de uma nova categoria, são aqueles que maximizam a medida $(e_i \times d_i)$. Procedimento este chamado de *Entropy and Density-based Selection* (EDS). Os objetos selecionados por EDS são então enviados para um agente externo rotular e posteriormente os dados são acrescentados ao conjunto de treino para atualizar o modelo. Ao fazer isso a matriz S deve ser reduzida eliminando as informações de similaridade envolvendo as amostras selecionadas pelo EDS.

Por fim, esse processo de seleção de dados via EDS e atualização do modelo pode ocorrer várias vezes, constituindo um classificador iterativo (do inglês *Iterative Classifier* (IC)) daí o nome IC-EDS (*Iterative Classifier - Entropy and Density-based Selection*). Este modelo pode detectar e aprender novas classes iterativamente explorando instâncias que habitam regiões de alta densidade e com rótulos altamente incertos.

Neste trabalho, a configuração do IC-EDS seguiu aquelas utilizadas em Coletta et al. (2022), no qual o algoritmo foi apresentado usando o SVM não-linear para gerar $\{\pi_i\}_{i=1}^{n}$ e o *K-means* para calcular $S = \{s_{ij}\}_{i,j=1}^{n}$ com critério de qualidade via silhueta.

2.3.3.2 Nearest Non-Outlier (NNO)

Segundo Bendale e Boult (2015), algoritmos do tipo NCM podem ser utilizados para aplicações *Open World*. Porém, algumas adaptações precisam ser feitas para que o modelo esteja alinhado com os requisitos exigidos para soluções dentro deste contexto.

O algoritmo NCM é um classificador baseado em distância (MENSINK et al., 2013; RISTIN et al., 2014), onde o processo de inferência se dá por meio do cálculo da distância entre o objeto não rotulado e valores de média de cada classe conhecida. Em seguida, a classe cuja média se mostrar mais próxima da amostra avaliada é então tida como sendo o rótulo do objeto. Em Bendale e Boult (2015), o modelo NCM é apresentado, formalmente, como se segue: considere um dado (como uma imagem) representado por um vetor de características de dimensão d, tal que, $x \in \mathbb{R}^d$. E seja $\mathscr{K}$ o conjunto de classes de objetos do problema, cada qual com um centroide μ_k onde $k \in \mathscr{K}$. A média μ_k relativa a cada classe é calculada pela Equação 11, onde I_k corresponde ao conjunto de amostras da classe k. Por fim, a classificação via NCM de uma amostra de teste representada por um vetor de características x, é determinada pelo centroide de classe mais próximo c^*, dado pela Equação 12, onde $\mathbf{d}$ é uma medida de distância, como a euclidiana. Outra forma muito usual de obter $\mathbf{d}$, é por meio da distância Mahalanobis de baixo posto (MENSINK et al., 2013), sendo este o tipo de distância adotado em Bendale e Boult (2015). Neste caso, este tipo de distância é otimizada sobre os dados de treino, buscando gerar uma representação linear dos dados para que então possam ser associados a uma média de classe (FUKUNAGA, 2013).

$$\mu_k = \frac{1}{|I_k|} \sum_{x_i \in I_k} x_i \tag{11}$$

$$c^* = \underset{k \in \mathscr{K}}{\operatorname{argmin}}\, \mathbf{d}(x, \mu_k) \tag{12}$$

Classificadores deste tipo fornecem uma alternativa simples para a construção de modelos escaláveis capazes de aprender incrementalmente novas classes, em que o conhecimento de novos padrões são incorporados ao modelo por meio da atualização do conjunto de médias de classe. Contudo, o modelo NCM assume que o problema é do tipo *Closed-set* e, portanto, não considera a existência de padrões desconhecidos nos dados.

Assim, o algoritmo *Nearest Non-Outlier* (NNO) proposto por Bendale e Boult (2015) visa extender, sem muitos custos, o modelo NCM para problemas *Open World* de maneira a definir fronteiras ao redor das classes e instâncias que residem longe destas fronteiras são atribuidas à uma classe desconhecida (ROSA; MENSINK; CAPUTO, 2016). Para

isso, NNO representa o modelo como um vetor de médias $M = [\mu_1, ..., \mu_k]$ e a classificação passa a ocorrer de acordo com um conjunto de funções $\varphi = \{\hat{f}_i(x)\}_{i=1}^{k}$ que determina a probabilidade do objeto x pertencer à classe i. A classe cuja função $\hat{f}_i(x)$ resulta no maior valor de probabilidade, determina a classificação da amostra x. Por outro lado, se $\hat{f}_i(x) \leq 0$ para todos os valores de i, então a amostra x é rotulada pelo NNO como algo desconhecido ou uma novidade. Considerando D o conjunto de novidades detectadas e, depois, rotuladas por um oráculo como uma nova classe $k+1$, o modelo NNO aprende incrementalmente a nova classe ao acrescentar no vetor M o valor da média μ_{k+1} obtido por meio da Equação 11 sobre D. Por fim, o conjunto φ é também atualizado, com o acréscimo de uma função $\hat{f}_{k+1}(x)$. Esse processo ocorre ciclicamente, de forma que o NNO é capaz então de aprender novos conceitos ao longo do tempo e aplicar esse conhecimento na inferência de novos dados. Em Bendale e Boult (2015), é provado que uma abordagem simples para detecção de novidades consiste na definição de um limiar τ de forma que para uma amostra x não ser rejeitada como algo desconhecido então $\hat{f}_{y^*}(x) > \tau$, em que $\hat{f}_{y^*}(x)$ é a $\hat{f}_i(x)$ de valor mais alto em φ.

Devido à indisponibilidade de código fonte para implementação do NNO descrito acima, foi implementada neste trabalho uma versão atualizada deste modelo com base no código fornecido em Fontanel et al. (2021)[4]. Aqui a classificação ocorre baseado no modelo DeepNCM (GUERRIERO; CAPUTO; MENSINK, 2018) que utiliza no lugar da transformada de Mahalanobis, uma representação dos dados geradas automaticamente via *Deep Learning*. No artigo é mostrado que a natureza altamente não-linear das representações geradas, elimina a necessidade de uma representação linear como a de Mahalanobis. Na execução dos experimentos, as representações apresentadas na Seção 4.3 foram aplicadas como representações não-lineares dos dados. As médias μ_k e o limiar τ foram computados através da versão online do NNO (ROSA; MENSINK; CAPUTO, 2016), em que um conjunto inicial de treinamento T_0 é utilizado para calcular τ. Neste trabalho, T_0 foi composto pelas amostras de treino das classes conhecidas pelo modelo desde o início, i.e., as classes *Solo* e *Planta Saudável*, sendo τ calculado na primeira iteração do procedimento proposto na Seção 3.1. Por fim, a função de probabilidade de classe $\hat{f}_i(x)$ foi definida de acordo com a Equação 13, juntamente com uma função $\hat{f}_{k+1}(x)$ na Equação 14 que denota a probabilidade do objeto x representar uma novidade.

$$\hat{f}_i(x) = \exp\left(-\frac{1}{2}||x - \mu_i||\right) \tag{13}$$

$$\hat{f}_{k+1}(x) = 1 - \sum_{i=1}^{k} \hat{f}_i(x) \tag{14}$$

Um detalhe importante de ser mencionado é que antes de calcular $\hat{f}_{k+1}(x)$, os valores assumidos por cada função em φ são comparados com o valor de τ de maneira que se

[4] <https://github.com/DarioFontanel/OWR-VisualDomains>

$\hat{f}_i(x) < \tau$, então $\hat{f}_i(x) = 0$ na Equação 14. No NNO, os objetos selecionados para receberem o rótulo de classe pelo oráculo são aqueles que maximizam $\hat{f}_{k+1}(x)$.

2.3.3.3 Incremental Classifier and Representation Learning (iCaRL)

O algoritmo iCaRL proposto por Rebuffi, Kolesnikov e Lampert (2016) é um modelo do tipo *Class-IL* (Seção 2.2.1) que utiliza a abordagem de *rehearsal* para lidar com o problema de *catastrophic forgetting*. O intuito do modelo é fornecer uma estratégia prática para o aprendizado simultâneo de classificadores e extratores de características, a medida que novas classes surgem para serem incorporadas ao modelo. Neste trabalho iCaRL foi utilizado em experimentos para demonstrar que o *framework* apresentado na Seção 3.1 permite, também, o uso de classificadores incrementais já existentes na formação de uma solução do tipo *Open World*, bastando apenas efetuar algumas adaptações (a depender do modelo).

Originalmente, o algoritmo iCaRL utiliza uma Rede Neural Convolucional como extrator de características em conjunto com um classificador baseado em NCM. Contudo, nos experimentos aqui conduzidos o extrator de características original foi substituído pelas representações elencadas na Seção 4.3, para que seja possível uma comparação justa com o modelo da literatura que se utiliza também destas representações dos dados. Assim, será dado foco nesta seção no procedimento de classificação utilizada pelo iCaRL. A metodologia empregada na atualização do extrator de características pode ser vista em detalhes no artigo original (REBUFFI; KOLESNIKOV; LAMPERT, 2016).

Anterior à classificação, o algoritmo iCaRL define um conjunto de exemplares P_k para cada uma das n classes conhecidas até o presente momento, onde cada conjunto é composto por uma mesma quantidade $m = B/n$ de amostras da classe k, sendo B o tamanho total da memória que armazena os exemplares. Através desta memória de dados, o algoritmo mantém o conhecimento de classes antigas quando novas surgem. Após definir $P_1, ..., P_n$ o classificador baseado em NCM chamado *Nearest-Mean-of-Exemplars* é executado. Este modelo inicialmente computa a média μ_k de cada classe com base no conjunto de exemplares P_k usando a Equação 11, onde agora $I_k = P_k$ e, por fim, a inferência de classe sobre uma amostra de teste com vetor de características x, se dá por meio da Equação 12.

Sempre que iCaRL encontra novas classes, é executada uma rotina para atualizar o conhecimento do modelo (conjunto de exemplares) a fim de assimilar os novos padrões. Como o intuito é realizar aprendizado incremental das classes, não se pode utilizar a média de classe real pois isso demanda o armazenamento de todos os dados de treinamento. Assim, utiliza-se uma quantidade limitada e flexível de exemplares de cada classe conhecida, escolhidos de maneira a gerar uma média μ_k mais próxima possível da média de classe verdadeira μ'_k. Para cada classe, as amostras p_i são selecionadas iterativamente do conjunto de treino até atingir a quantidade m através da Equação 15,

procurando sempre selecionar amostras que aproximem μ_k de μ'_k. Esse procedimento é realizado sobre novas classes e também sobre as classes inicialmente conhecidas.

$$p_i = \underset{x \in I_k}{\operatorname{argmin}} \left\| \mu'_k - \frac{1}{i} \left[x + \sum_{j=1}^{i-1} p_j \right] \right\| \tag{15}$$

Além disso, à medida que novos padrões precisam ser aprendidos, alguns exemplares de classes conhecidas precisam ser excluídos para abrir espaço na memória a fim de respeitar o limite B estabelecido. Em Rebuffi, Kolesnikov e Lampert (2016) os autores fazem isto através de um procedimento inspirado em *herding* (WELLING, 2009). A lógica é simples, ao construir os conjuntos P_k usando a Equação 15, os exemplares são ordenados de maneira que os primeiros são mais importantes que os demais. Dessa forma, para reduzir as amostras de uma quantidade m' para m basta manter as amostras $p_1, ..., p_m$ e descartar as instâncias $p_{m+1}, ..., p_{m'}$. Isso permite atualizar os exemplares sem a necessidade de revisitar as amostras de treino.

Esta estratégia é adequada para o contexto de *Class-IL*, mas apresenta problemas quando se trata de um cenário *Open World*. Pois parte-se do princípio de que depois de uma classe ser aprendida o modelo deixa de receber amostras de tal categoria ao longo das iterações seguintes. E isto pode não ser verdade em *Open World*, onde não se sabe *a priori* a natureza das instâncias selecionadas como novidade por um critério de detecção de novas classes. Portanto, neste trabalho o iCaRL adaptado não executa a estratégia baseada em *herding* como no algoritmo original. Ao invés disso, os objetos selecionados como novidades são adicionados à memoria de exemplares B (nos experimentos $B = 60$) e então os conjuntos P_k são redefinidos à partir de B usando a Equação 15.

Por fim, a última adaptação realizada no modelo foi referente ao uso da solução apresentada por Mensink et al. (2015) com o intuito de permitir o classificador gerar vetores de distribuição de probabilidade de classe $\{y_{ik}\}$ na sua saída, uma vez que o modelo baseado em NCM fornece como resultado apenas medidas de distância. Dessa forma, y_{ik} foi obtido através da Equação 16 onde $\mathbf{d}$ é uma medida de distância.

$$y_{ik} = \frac{\exp(-\mathbf{d}(\mu_k, i))}{\sum_{k'=1}^{c} \exp(-\mathbf{d}(\mu_{k'}, i))} \tag{16}$$

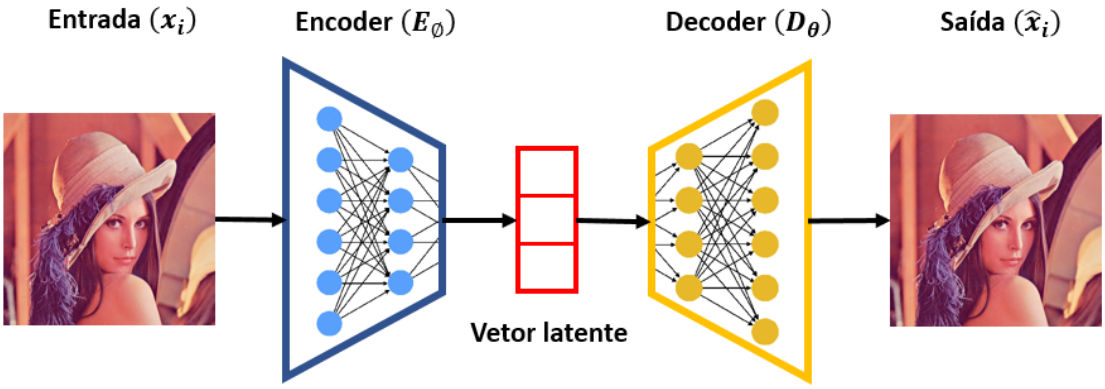
2.4 Representação de dados: Autoencoder Variacional

O desempenho de um modelo tem relação estreita com a qualidade dos dados utilizados ao longo do treinamento. À medida que a dimensionalidade de um conjunto de dados cresce, o volume do espaço de busca aumenta, dificultando a obtenção de um modelo que consiga aprender os padrões envolvidos na natureza do problema. Este obstáculo é conhecido como maldição da dimensionalidade (BELLMAN; CORPORATION; COLLECTION, 1957), situação que é típica em visão computacional, por conta das altas dimensões que

normalmente as imagens apresentam. Felizmente, é possível transformar dados de altas dimensões para uma representação com dimensões menores, preservando características importantes das amostras originais, o que reduz também o custo computacional na indução de um modelo. Os métodos para se fazer isso são divididos em lineares, como PCA (*Principal Component Analysis*), e não lineares que é o caso do *Autoencoder* que tem alcançado resultados no estado da arte em diferentes aplicações (KODIROV; XIANG; GONG, 2017; SHI et al., 2021; WANG, 2021).

Um *Autoencoder* (AE) é um modelo de rede neural não supervisionada treinada para codificar, de forma eficiente, os dados não rotulados de entrada empregando uma quantidade reduzida de dimensões (HINTON; SALAKHUTDINOV, 2006). O intuito principal do AE é fazer compressão de dados preservando suas principais características. Tal modelo é composto por duas partes, *encoder* e *decoder* que são acoplados de tal maneira a gerar um gargalo entre os componentes onde as codificações são geradas. A Figura 11 mostra um exemplo de AE.

Figura 11 – Estrutura de um Autoencoder clássico.



Fonte: Elaboração própria.

O processamento ocorre da seguinte forma: os dados de entrada x_i passam pelo *encoder* E_ϕ que gera uma representação comprimida dos dados, chamada de variáveis latentes. Então o vetor de variáveis produzido é inserido no *decoder* D_θ que tenta reconstruir a entrada x_i. Quanto menor o erro de reconstrução melhor a qualidade das variáveis latentes geradas. Esse erro é utilizado como função de custo para treinar a rede por meio de algoritmos de gradiente descendente, como o *backpropagation* (RUMELHART; HINTON; WILLIAMS, 1988). A Equação 17 é um exemplo de função de perda normalmente aplicada no treino de AE, se referindo ao erro médio quadrático entre a entrada e saída do modelo onde N é o total de amostras de treinamento.

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N} ||x_i - D_\theta(E_\phi(x_i))||_2^2 \tag{17}$$

Ao longo do treinamento o AE busca aprender uma função identidade I para conseguir minimizar o erro de reconstrução a zero. Em muitos casos isso pode ocorrer, gerando então *overfitting* e consequente perda de generalização (RUFF et al., 2018). Entretanto isso pode ser evitado, ao se configurar corretamente alguns parâmetros do modelo e então o AE passa a capturar estruturas interessantes nos dados.

A representação de conhecimento executado pelo AE se dá pelo uso de variáveis latentes. O termo latente vem do latim *lateo* que significa escondido, ou seja, a variável latente mede uma característica ou grandeza que não pode ser medida diretamente, devido a limitações físicas ou então por lidar com conceitos abstratos (SPIRTES, 2015). Contudo essas variáveis podem ser inferidas através de variáveis observáveis e então utilizadas para explicá-las. São largamente estudadas na estatística (EVERITT, 1984), mas encontram aplicações em outras áreas como em AM, na tarefa de redução de dimensionalidade (CARREIRA-PERPINAN, 2001), onde algumas variáveis latentes são empregadas para representar dados multidimensionais.

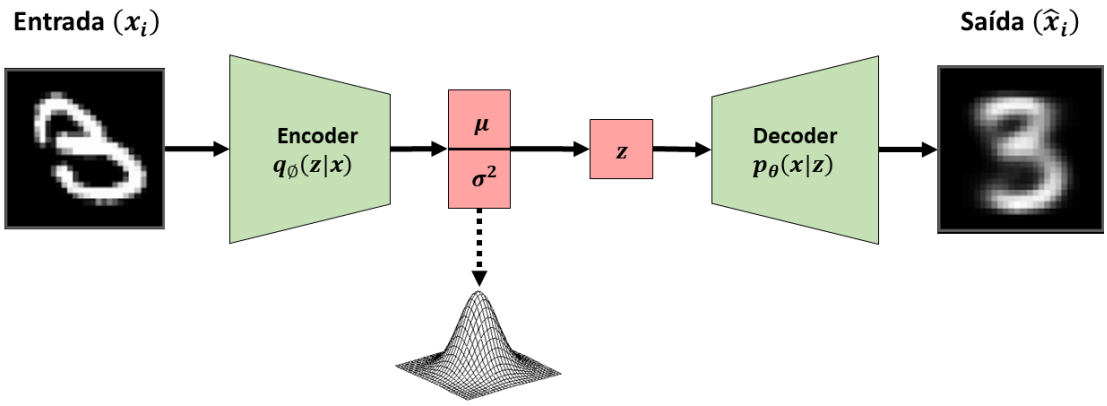
Existem variantes do AE convencional que permitem sua aplicação em outras tarefas além da compressão de dados. O *Denoising Autoencoder* (DAE) (VINCENT et al., 2010), por exemplo, é aplicado para tirar ruídos de imagens para melhorar sua qualidade enquanto o *Variational Autoencoder* (VAE) ou Autoencoder Variacional, permite que o modelo utilize as estruturas latentes para gerar novos dados, tornando o AE também generativo.

O VAE é um modelo apresentado pela primeira vez no artigo de Kingma e Welling (2014) que une *Deep Learning* e estatística Bayesiana para definir variáveis latentes em forma de distribuição de probabilidade. No AE básico o vetor latente produzido apresenta valores fixos para os dados de forma a gerar um espaço latente discreto, assim o novo espaço de características é pouco regularizado, não propiciando geração de novos objetos por amostragem das variáveis latentes. No VAE as variáveis latentes são contínuas, assumindo forma de distribuição normal com matriz de covariância diagonal. Com isso, a transição entre as classes ocorre de maneira suavizada no novo espaço de características e a amostragem sobre as variáveis latentes contínuas gera dados sintéticos (DOERSCH, 2016).

Nesta variante, modelos probabilísticos são aplicados na formulação matemática do algoritmo onde o *encoder* passa a ser tido como *encoder* probabilístico ou modelo de reconhecimento e o *decoder* como *decoder* probabilístico. A Figura 12 ilustra um VAE. O VAE aplica os princípios de modelos de variáveis latentes ao buscar modelar uma distribuição de probabilidade como a da Equação 18. A premissa é de que existem variáveis latentes z_i que geram os dados observados x_i. Tais modelos são treinados ao maximizar a probabilidade marginal dos dados observados x sob os parâmetros θ do

modelo (KINGMA; WELLING, 2014), expresso pela Equação 19.

Figura 12 – Estrutura de um Autoencoder Variacional.



Fonte: Elaboração própria.

$$p_\theta(x,z) = p_\theta(x|z)p_\theta(z) \tag{18}$$

$$p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz \tag{19}$$

Uma vez que o logaritmo é uma função monotonicamente crescente e simplifica a computação de alguns cálculos, a Equação 19 é modificada e então a maximização ocorre sobre o *log* da probabilidade marginal como dado por 20.

$$\begin{aligned} \log p_\theta(X) &= \log \prod_{i=1}^{N} p_\theta(x_i) \\ &= \sum_{i=1}^{N} \log p_\theta(x_i) \\ &= \sum_{i=1}^{N} \log \int p_\theta(x|z)p_\theta(z)dz \end{aligned} \tag{20}$$

Na Equação 20, X se refere a um conjunto de dados de N amostras. Na grande maioria das situações não é possível maximizar a probabilidade marginal usando diretamente essa equação devido a presença da integral, sendo então um problema intratável (KINGMA; WELLING, 2014). O mesmo ocorre com a probabilidade posterior das variáveis latentes da Equação 21. Uma das formas de superar esse problema é o uso de inferência variacional (BLEI; KUCUKELBIR; MCAULIFFE, 2017).

$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} \tag{21}$$

Considerando que a probabilidade $p_\theta(z|x)$ é intratável, pode-se utilizar uma nova distribuição $q_\phi(z)$ de formato qualquer, que será utilizada para tentar aproximar $p_\theta(z|x)$. Uma maneira eficiente de realizar tal aproximação é minimizar a divergência Kulback-Leibler (KL) entre as distribuições, como mostra a Equação 22.

$$KL[q_\phi(z)||p_\theta(x|z)] = -\sum_z q_\phi(z) \log \frac{p_\theta(x|z)}{q_\phi(z)} \tag{22}$$

Fazendo uso do Teorema de Bayes e rearranjando os termos tem-se a Equação 23.

$$\begin{aligned} -\sum_z q_\phi(z) \log \frac{p_\theta(x|z)}{q_\phi(z)} &= -\sum_z q_\phi(z) \log \left(\frac{p_\theta(x,z)}{q_\phi(z)} . \frac{1}{p_\theta(x)} \right) \\ &= -\sum_z q_\phi(z) \left(\log \frac{p_\theta(x,z)}{q_\phi(z)} - \log p_\theta(x) \right) \\ &= -\sum_z q_\phi(z) \log \frac{p_\theta(x,z)}{q_\phi(z)} + \sum_z q_\phi(z) \log p_\theta(x) \\ &= -\sum_z q_\phi(z) \log \frac{p_\theta(x,z)}{q_\phi(z)} + \log p_\theta(x) \end{aligned} \tag{23}$$

Ao isolar o segundo termo do lado direito da igualdade, chega-se a Equação 24.

$$\log p_\theta(x) = KL[q_\phi(z)||p_\theta(x|z)] + \sum_z q_\phi(z) \log \frac{p_\theta(x,z)}{q_\phi(z)} \tag{24}$$

Com isso obtêm-se outra maneira de se calcular o logaritmo da probabilidade marginal composta por dois termos. O primeiro termo é a divergência KL entre a distribuição variacional $q_\phi(z)$ e o posterior intratável $p_\theta(x|z)$, enquanto o segundo termo é chamado na literatura de *variational lower bound* ou *evidence lower bound* (ELBO), pois como a divergência KL é não negativa a ELBO funciona como limite inferior do logaritmo da probabilidade marginal (DOERSCH, 2016). Na inferência variacional busca-se maximizar a ELBO expressa na Equação 25 ao invés da equação de probabilidade marginal original, já que nesta nova representação o problema deixa de ser intratável (KINGMA; WELLING, 2014).

$$\mathscr{L} = \sum_z q_\phi(z) \log \frac{p_\theta(x,z)}{q_\phi(z)} \tag{25}$$

Da ELBO se deriva a função objetivo da Equação 26, que é de fato aplicado na obtenção de um VAE. Este resultado é obtido ao reescrever a ELBO em forma de valor esperado, fazer a substituição do Teorema de Bayes e gerar um termo relativo a divergência KL ao separar os logaritmos.

$$
\begin{aligned}
\mathscr{L} &= \sum_z q_\phi(z) \log \frac{p_\theta(x,z)}{q_\phi(z)} \\
&= \mathbb{E}_{q_\phi(z)} \log \frac{p_\theta(x,z)}{q_\phi(z)} \\
&= \mathbb{E}_{q_\phi(z)} \log \frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z)} \\
&= \mathbb{E}_{q_\phi(z)} \log p_\theta(x|z) + \mathbb{E}_{q_\phi(z)} \log \frac{p_\theta(z)}{q_\phi(z)} \\
&= \mathbb{E}_{q_\phi(z)} \log p_\theta(x|z) - KL[q_\phi(z)||p_\theta(z)] \\
&= \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) - KL[q_\phi(z|x)||p_\theta(z)]
\end{aligned}
\tag{26}
$$

No final, como a distribuição q é aplicada com o objetivo de se aproximar de $p_\phi(z|x)$, então define-se $q_\phi(z) = q_\phi(z|x)$.

O intuito de treinar um VAE com a função objetivo 26 é encontrar uma distribuição $q_\phi(z|x)$ que modele o espaço latente de forma a maximizar a probabilidade dos dados observados x e ao mesmo tempo modelar uma distribuição $p_\theta(x|z)$ que seja capaz de gerar novos dados a partir de amostras z do espaço latente induzido.

O termo $\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z)$ da equação se destina a aprender duas distribuições q e p de tal forma que $p_\theta(x|z)$ seja capaz de reconstruir o dado original de entrada a partir da variável latente z gerada por $q_\phi(z|x)$ que mapeia o dado x na representação latente z. Em outras palavras, o primeiro termo da Equação 26 se trata do erro de reconstrução do VAE.

O segundo termo dado pela divergência KL, busca aproximar a distribuição q de uma $p_\theta(z)$. A distribuição $p_\theta(z)$ define a forma desejada para as variáveis latentes z e o VAE estabelece que estas variáveis devem assumir a forma de uma distribuição normal multivariável com matriz de covariância diagonal, assim $p_\theta(z) = N(0,I)$. O uso deste tipo de modelo se justifica pelo fato de uma distribuição gaussiana simples ser capaz de modelar distribuições mais complicadas se uma função complexa o suficiente for utilizada para fazer este mapeamento (DOERSCH, 2016), como uma rede neural artificial.

A presença do cálculo de divergência na função de custo funciona como um termo de regularização, procurando evitar a ocorrência de *overfitting* ao impor a presença de uma variância na distribuição $q_\phi(z|x)$. Assim o modelo aprende uma estrutura com informações importantes a respeito do espaço latente dos dados permitindo também construir codificações z que variem suavemente em um espaço latente contínuo, de tal forma que uma amostragem feita neste espaço gera dados novos e realistas ao passar por $p_\theta(x|z)$, no mesmo domínio dos dados originais. Sem a variância em $q_\phi(z|x)$ o modelo se comportaria como um AE convencional.

Para modelar o espaço latente de acordo com os dados, assume-se que a distribuição $q_\phi(z|x)$ é dada por uma mistura de gaussianas multivariadas cujos parâmetros $\mu_\phi(x)$ e $\sigma^2_\phi(x)$ são uma função da entrada x como apresentado na Equação 27.

$$q_\phi(z|x) = N(\mu_\phi(x), diag(\sigma^2_\phi(x))) \tag{27}$$

Os parâmetros da distribuição q são modelados por meio de uma rede neural que gera na sua saída um vetor de médias μ e um vetor de variâncias σ^2 para cada dado x_i do conjunto de dados X. Esta rede neural é o *encoder* probabilístico do VAE, definida por um conjunto de pesos ϕ (KINGMA; WELLING, 2014).

De forma semelhante a distribuição $p_\theta(x|z)$ é composta por uma mistura de distribuições diagonais que assume normalmente a forma de gaussiana ou Bernoulli, dependendo da natureza dos dados. A distribuição p é alcançada por meio de uma outra rede neural que constitui o *decoder* probabilístico do VAE, fornecendo como saída os parâmetros da distribuição escolhida e é definida por um conjunto de pesos θ.

Ambos *encoder* e *decoder* são treinados juntos por meio de técnicas de gradiente descendente como SGD (*Stochastic Gradient Descent*), que atualiza os pesos da rede neural buscando minimizar a função objetivo da Equação 26.

O processo geral de funcionamento do VAE ocorre da seguinte forma: um dado x_i entra no *encoder* que produz vetores de médias e variâncias definindo uma gaussiana multivariada diagonal, que representa os atributos latentes do dado. A variável latente z_i é amostrada de $q_\phi(z_i|x_i)$ e em seguida é passada ao *decoder* para fazer a reconstrução do dado, fornecendo como saída os parâmetros da distribuição escolhida (Gaussiana ou Bernolli) para $p_\theta(x|z)$.

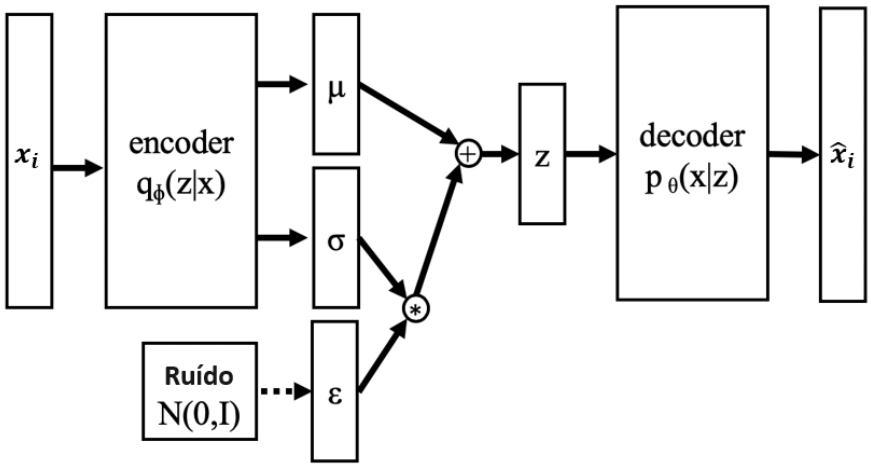
A amostragem da variável latente z_i não é feita diretamente sobre a saída do *encoder* pois isso gera um problema na retropopagação do erro no treinamento devido a geração de descontinuidades (DOERSCH, 2016). A solução encontrada para o problema foi representar z_i como uma função determinística do dado x_i e de um ruido aleatório ε chamado de truque de reparametrização (do inglês *reparameterization trick* (KINGMA; WELLING, 2014)), como expresso na Equação 28.

$$\begin{gathered} z_i = g_\phi(x_i, \varepsilon_i) = \mu_\phi(x_i) + diag(\sigma_\phi(x_i)).\varepsilon_i \\ \varepsilon_i \sim N(0, I) \end{gathered} \tag{28}$$

Essa definição permite que z siga a distribuição $q_\phi(z_i|x_i)$ e estabelece uma continuidade entre o *encoder* e *decoder*. O processo de funcionamento do VAE é ilustrado na Figura 13.

Um VAE treinado pode ser usado para várias aplicações. Amostrar um ponto da distribuição $p_\phi(z)$ e passá-lo ao *decoder* permite a geração de dados sintéticos do mesmo domínio dos dados de treino (HOU et al., 2017), mas com características que o tornam um novo objeto. É possível também realizar redução de dimensionalidade (MAHMUD; HUANG; FU, 2020) ou extração de *features* ao inserir os dados no *encoder* e extrair o vetor de médias geradas na saída que corresponde à variável latente mais provável. Por fim, o caráter probabilístico do VAE pode ser aproveitado para identificar anomalias

Figura 13 – Fluxo de processamento no VAE.



Fonte: Elaboração própria.

ao calcular para um dado x_i, o erro de reconstrução (BAUR et al., 2019) ou então a divergência KL (ZIMMERER et al., 2018).

2.5 Trabalhos Relacionados no Contexto Agrícola

Na literatura existem poucos trabalhos que se propõem a apresentar um modelo capaz de aprender, de forma contínua e incremental, novas ameaças nos cultivos agrícolas. Pelo que se tem conhecimento, apenas os trabalhos de Pantazi, Moshou e Bravo (2016) e Coletta et al. (2022) se alinham com esta proposta.

Em Pantazi, Moshou e Bravo (2016) um novo modelo de *Active Learning* é apresentado, combinando detecção de novidades com aprendizado incremental de classes. No trabalho é utilizado um conjunto de dados de imagens hiperespectrais de uma plantação de milho com dez espécies diferentes de daninhas. O intuito foi desenvolver um modelo que fosse capaz de aprender incrementalmente a classificar as espécies de daninhas que não foram apresentadas previamente durante o treinamento.

No artigo, os autores trabalharam com classificadores *one-class mixture of Gaussians* (MOG), *one-class self-organizing map* (SOM), *one-class* SVM e *one-class Autoencoder*. Inicialmente o modelo é treinado apenas com as amostras da classe milho e posteriormente, na fase de teste, as amostras de uma classe de daninha são apresentadas ao classificador *one-class* na expectativa de serem detectadas como um evento anormal ou *outlier*. Uma vez encontrada uma quantidade suficiente de *outliers*, um novo classificador *one-class* é treinado com os objetos da nova classe de daninha. Em seguida, amostras de uma outra classe de daninha são apresentadas durante a fase de teste para serem classificadas pelos modelos treinados até o momento e então o procedimento se repete,

gerando para cada nova daninha um classificador *one-class*. Assim o modelo aprende continuamente novas classes. Entre os classificadores testados, MOG e SOM geraram os melhores resultados.

Coletta et al. (2022) apresentam uma versão aprimorada de um modelo semi-supervisionado, capaz de identificar nos dados o surgimento de um novo padrão e também aprender incrementalmente a nova classe detectada. Este algoritmo é chamado pelos autores de IC-EDS (Seção 2.3.3.1). Para isso, foi combinada em uma única medida, os valores dos critérios de entropia e densidade para cada amostra, de forma que os objetos *i* que maximizam essa combinação são fortes candidatos a representarem uma nova classe.

Essas amostras são então rotuladas por um agente externo capaz de dizer o rótulo dos dados e em seguida os objetos são inseridos no conjunto de treino juntamente com as amostras já existentes, para retreinar o modelo com as novas informações e assim aprender incrementalmente a nova classe. O conjunto de dados foi obtido por um VANT sobre uma plantação de eucalipto com focos da doença *Ceratocystis wilt*. A classe relativa à doença foi apresentada apenas na fase de teste e foi o alvo a ser aprendido ao longo da execução do modelo. Este trabalho serviu como uma das inspirações para a realização da presente pesquisa e, por conta disso, o modelo IC-EDS foi usado como comparação nos experimentos.

Como será visto no Capitulo 3, o *framework* proposto permite a aplicação de diferentes técnicas para a detecção de novas classes, visando a construção de um classificador para problemas *Open World*. Nesse sentido, seguindo o trabalho, este classificador será avaliado em um cenário agrícola real, reconhecidamente dinâmico e ruidoso.

3 ABORDAGEM PROPOSTA

Este capítulo introduz o *framework* proposto neste trabalho, cujo objetivo é tornar a classificação de dados mais flexível e aderente à realidade com base no aprendizado incremental de novas classes, como será discutido na Seção 3.1. Mais especificamente, este trabalho busca melhor esquematizar os componentes envolvidos numa alternativa de classificador *Open World*, capaz de lidar com cenários abertos, em que novidades/mudanças podem acontecer necessitando a atualização do modelo. Por fim, na Seção 3.2, são destacados um conjunto de critérios numéricos que podem ser *plugados* na estrutura discutida, cujo objetivo é o de determinar a presença de novas classes, o que considera-se ser parte essencial para o sucesso de classificadores mais autônomos e capazes de serem operacionalizados no mundo real.

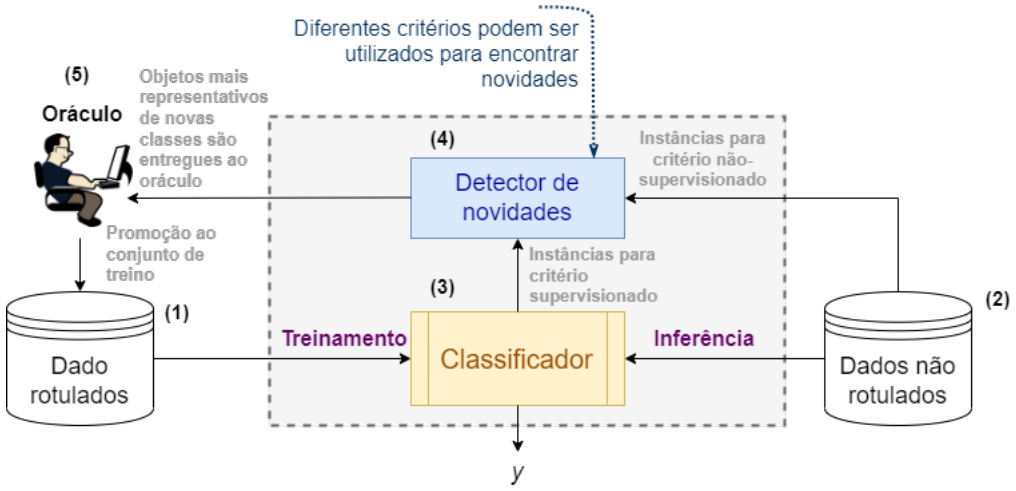
3.1 Procedimento de aprendizado incremental de classes

A metodologia proposta se refere a uma estrutura algorítmica que permite a implementação de classificadores capazes de atuar em cenários mais realistas de *Open World*, por meio do aprendizado contínuo e incremental de novas classes. Para atingir este fim, o modelo deve incorporar algumas das características dos algoritmos de Aprendizado Incremental (Seção 2.2), precisamente a habilidade de aprender a partir de novas amostras sem retreinar completamente o modelo e, além disso, preservar o conhecimento de classes já observadas. Contudo, o escopo de Aprendizado Incremental não se preocupa explicitamente em como podem ser obtidas essas novas amostras, de forma que sejam plenamente passíveis de serem servidas para o procedimento incremental. Em outras palavras, para que o sistema exiba um ciclo de vida, com mínima necessidade de interrupções e intervenções (humanas), um componente capaz de preparar as novas amostras para serem promovidas ao modelo é requerido. No *framework* proposto, este componente é o mais importante, tendo a responsabilidade de identificar novidades e fornecê-las para a rotulagem, momento em que se revelam novos padrões/classes, os quais podem ser então aprendidos pelo modelo.

A Figura 14 descreve a estrutura proposta englobando os principais componentes, que juntos permitem que um modelo se adapte às mudanças de forma iterativa. Durante esse processo o objetivo é encontrar novidades, para que instâncias mais prováveis de representar uma nova classe possam ser utilizadas para expandir o conhecimento do modelo. Assim, o foco de nossa metodologia está no componente denominado Detector de Novidades, no qual diferentes critérios para detecção de novidades podem ser empregados e combinados. Este componente, possui um mecanismo inspirado em Aprendizado Ativo, visto na Seção 2.1, para encontrar as novas classes de forma

parcimoniosa, ou seja, observando e selecionando apenas as instâncias mais frutíferas para o refinamento do modelo. Essas instâncias serão rotuladas por um agente externo (oráculo/especialista humano) quando então uma nova classe pode ser revelada. Então, essas instâncias são finalmente inseridas no *dataset* rotulado, aumentando-o para que o classificador possa ser treinado novamente (usando todos os dados disponíveis) ou apenas atualizado via *Class-Incremental Learning*.

Figura 14 – Esquema ilustrativo do funcionamento do *framework* para o reconhecimento de novas classes.



Fonte: Elaboração própria.

Formalmente, seja C^K o conjunto das K classes conhecidas em um problema e C^U o conjunto das novas classes desconhecidas que pode ter, hipoteticamente, tamanho infinito. Nossa estrutura para Reconhecimento de Novas Classes, F_{RNC}, exige que os seguintes componentes sejam aplicados: um conjunto de dados rotulados X^L, um conjunto de dados não rotulados X^U, um modelo de classificação $\hat{f}$, um mecanismo de detecção de novidades Δ, e um oráculo capaz de anotar corretamente as classes de um conjunto de instâncias selecionadas como sendo novidades, N. Embora N seja uma saída clara da estrutura, o classificador usado, $\hat{f}$, também fornecerá predições de forma iterativa, Y, para instâncias em X^U (esperando obter resultados cada vez melhores, mesmo que surjam novas classes). A Equação 29 define matematicamente o *framework* proposto:

$$F_{NCR}\left(\begin{array}{l}\hat{f}_t(X_t^L, X_t^U) \rightarrow Y_t \\ \Delta(X_t^L, X_t^U, Y_t) \rightarrow N_t^U\end{array}\right), \tag{29}$$

com $X_t^L = \langle X_t^L + N_{t-1}^L \rangle$ e $X_t^U = \langle X_t^U - N_{t-1}^U \rangle$, e o número de iteração $t > 1$, o que significa que as instâncias selecionadas, N^U, da iteração $t-1$ são removidas de X_t^U para serem inseri-

das em X_t^L. Mas para isso, é necessária uma etapa intermediária de rotulagem, tal que $\Omega(N^U) \rightarrow N^L$, sendo $\Omega(.)$ capaz de atribuir o rótulo de classe correto para as instâncias, detectando padrões antes inexistentes no conjunto de treino do modelo. Assim a execução do processo da Figura 14 ocorrerá iterativamente como descrito a seguir:

- **Passo 1:** Use os dados rotulados, X_t^L para treinar um classificador $\hat{f}$;
- **Passo 2:** Use $\hat{f}$ para prever rótulos de classe de instâncias em um conjunto de dados não rotulados, X_t^U, fornecendo Y_t, normalmente uma distribuição de probabilidades de classes;
- **Passo 3:** Um ou mais critérios de Δ são executados nos dados disponíveis, X_t^U e/ou X_t^L, e/ou saída do classificador, Y_t, para gerar um conjunto de instâncias pertencentes a uma nova classe, N_t^U;
- **Passo 4:** Remova as instâncias N_t^U do conjunto de dados não rotulados X_t^U e forneça-os para rotulagem, $\Omega(N^U) \rightarrow N^L$;
- **Passo 5:** Depois de rotular as instâncias, N_t^L, incremente os dados rotulados, X_t^L com eles, treine novamente o classificador, $\hat{f}$, e para uma próxima iteração do *framework*, vá para o Passo 2;

Convém salientar que deseja-se também nesta estrutura reduzir a complexidade em termos de *features*, almejando um espaço que favoreça a separabilidade e distinção de novas classes. Deste modo, adianta-se aqui que o algoritmo apresentado nesta seção poderá atuar sobre dados com características transformadas. Esta possibilidade é melhor discutida no Capítulo 4, referente à metodologia da pesquisa.

Por fim, para que essa solução possa ser aplicada em problemas *Open World*, requisitos apresentados na Seção 2.3.2 devem ser satisfeitos pelos componentes (3) e (4) do *framework*. Em relação ao classificador (3), o modelo deve ser capaz de se adaptar continuamente às novas classes encontradas, sem necessitar reaprender padrões apresentados anteriormente. Assim o modelo usado em (3) deve ser do tipo incremental. Sobre o Detector de Novidades (4), pode ser usado qualquer método capaz de destacar instâncias que não se encaixam na base de conhecimento do modelo, a partir tanto dos dados não rotulados (critério não-supervisionado) ou das saídas do classificador (critério supervisionado) ou ainda ambos ao mesmo tempo, através da combinação de dois ou mais critérios. Qualquer classificador e critério de detecção de novas classes que esteja de acordo com estes requisitos, pode ser implementado nos componentes (3) e (4) da Figura 14 para compor um modelo de aplicação em cenários *Open World*. Assim se tem uma estrutura flexível onde o usuário apenas precisa configurar o classificador e medida de seleção que sejam mais indicados para um dado problema de Cenário Aberto, uma vez que não existe uma solução que é aplicável para todos os problemas do mundo real. Cada contexto demandará uma solução específica.

3.2 Critérios para detecção de novas classes

A estrutura algorítmica estabelecida na Seção 3.1 deve ser equipada com um ou mais critérios, objetivando-se identificar amostras no conjunto de teste que potencialmente são representantes de um novo padrão/classe. Tais critérios atuam no componente (4) do esquema da Figura 14. Com o intuito de avaliar o *framework* apresentado, foram aplicados nos experimentos três critérios para detecção de novas classes: entropia (critério supervisionado), Silhueta Baseada em Centroide e Silhueta Baseada em Instâncias (ambas critérios não-supervisionados).

O uso da medida de silhueta para detecção de novidades não é algo novo, em Masud et al. (2011) esta ideia foi aplicada para integrar detecção de novas classes a classificadores tradicionais de *data stream*. Para isso, os autores computam a silhueta em uma implementação de aprendizado incremental, ao substituir as amostras brutas de treinamento por representações chamadas de *pseudoclusters*. Esses *pseudoclusters* armazenam as informações de centroide e raio de cada *cluster* gerado sobre o conjunto de treino via uso de *K-means* e, então, os dados não rotulados que estiverem fora dos limites de decisão, definidos pela extensão dos *pseudoclusters*, são denominados *outliers*. O valor de coesão entre os *outliers* é verificada e as q instâncias que ultrapassam um limiar definido são tidas como representantes de uma nova classe.

Para computar essa coesão, Masud et al. (2011) utilizam o cálculo da silhueta, que é um método não-supervisionado aplicado em algoritmos de agrupamentos (ex: *K-means*) para verificar a consistência dos *clusters* obtidos (ROUSSEEUW, 1987; YEUNG, 2001). Avalia quantitativamente quão próximos os objetos estão de outros do seu mesmo grupo (homogeneidade ou similaridade) comparado a objetos de grupos vizinhos (heterogeneidade ou dissimilaridade). Esse índice varia entre -1 e 1 e quanto maior o valor resultante, melhor a qualidade dos agrupamentos.

De maneira mais formal, considere um objeto $j \in \{1,2,...,N\}$ pertencente a um grupo $p \in \{1,...,c\}$. A silhueta deste objeto pode ser calculada pela Equação 30:

$$s_j = \frac{b_{pj} - a_{pj}}{max\{a_{pj}, b_{pj}\}}, \tag{30}$$

onde a_{pj} é a distância média do objeto j a todos os objetos de seu grupo p. Para calcular b_{pj}, considere a distância média d_{qj} do objeto j a todos os objetos de um outro grupo q. Pode-se definir b_{pj} como o menor valor d_{qj} calculado para $q \in \{1,...,c\}$, $q \neq p$, ou seja, $b_{pj} = min(d_{qj})$, $q \neq p$. Dessa forma, o valor de b_{pj} representa a dissimilaridade do objeto j ao grupo vizinho mais próximo. O denominador desta equação é usado apenas como um termo de normalização.

Porém essa forma de cálculo tem um alto custo computacional $N(O^2)$. Então Alves, Campello e Hruschka (2006) propõem uma simplificação, chamada de Silhueta Simpli-

ficada, que possui desempenho semelhante à forma padrão e demanda menor processamento, com complexidade linear *N(O)*. A Silhueta Simplificada envolve tornar b_{pj} a distância do objeto j ao centro do *cluster* vizinho mais próximo e a_{pj} a distância desse objeto até o centro de seu grupo.

Nossa abordagem de uso da silhueta como detector de novas classes se inspira nos trabalhos de Masud et al. (2010), Masud et al. (2011), porém utiliza como base matemática a versão da Silhueta Simplificada. Como será visto no Capítulo 4, um conjunto de experimentos foram conduzidos sobre a estrutura apresentada na Seção 3.1 em dois casos diferentes: primeiramente aplicando classificadores não-incrementais (caso 1) e posteriormente com modelos incrementais (caso 2). Para cada caso, os critérios de silhueta foram utilizados de forma diferente na detecção de novas classes sobre os dados não rotulados. A seguir é explicado como se deu o uso de cada um dos três critérios utilizados nos experimentos.

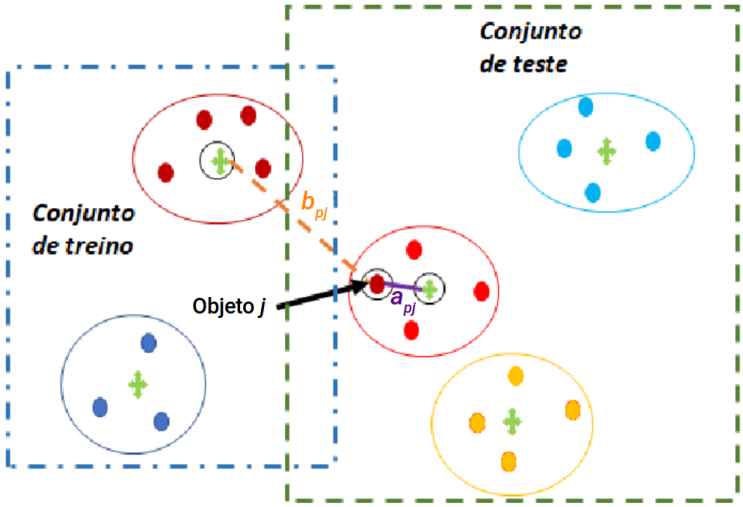
- **Entropia**: A entropia, calculada pela Equação 9, busca identificar amostras não rotuladas cujo vetores de probabilidade de classes apresentam maior incerteza de classificação, i.e., maior valor de entropia, destacando instâncias que fogem dos padrões conhecidos. Esse critério foi implementado de maneira similar ao que ocorre com o modelo IC-EDS (Seção 2.3.3.1).

- **Silhueta Baseada em Centroide**: Para o caso 1, a Figura 15 ilustra como a medida foi implementada e também indica como os parâmetros de cálculo da Equação 30 foram definidos. Duas partições de dados foram geradas: uma sobre o conjunto de treino e outra sobre o conjunto de teste. Nota-se na figura que, dado um objeto j no conjunto de teste, para o cálculo de b_{pj} considera-se aquele centro de grupo vizinho mais próximo contido no conjunto de treino.

  O termo a_{pj} se refere à similaridade do objeto j ao seu próprio grupo, i.e., o conjunto de teste. Este critério é aqui denominado como Silhueta Baseada em Centroide devido ao fato de tanto a_{pj} quanto b_{pj} considerarem a informação de centroides no cálculo das distâncias. Quanto maior o valor desta medida para um certo objeto j nos dados de teste, maiores são as chances deste objeto ser muito distinto daqueles já conhecidos (presentes no conjunto de treino). As amostras com os maiores valores de Silhueta Baseada em Centroide podem apontar a presença de novos padrões nos dados, ou seja, uma possível nova classe.

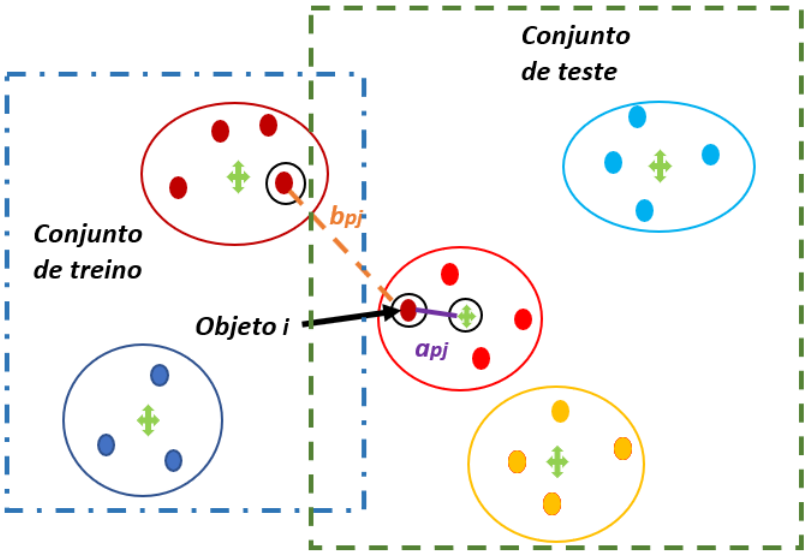
- **Silhueta Baseada em Instância**: A diferença deste critério em relação ao anterior é que o termo b_{pj} da Equação 30 passa a ser a distância do objeto j no conjunto de teste ao objeto mais próximo do *cluster* mais próximo no conjunto de treino, enquanto a_{pj} permanece igual ao do critério anterior. A Figura 16 mostra como atua esta outra versão.

Figura 15 – Abordagem da Silhueta Baseada em Centroide para a detecção de novidades.



Fonte: Elaboração própria.

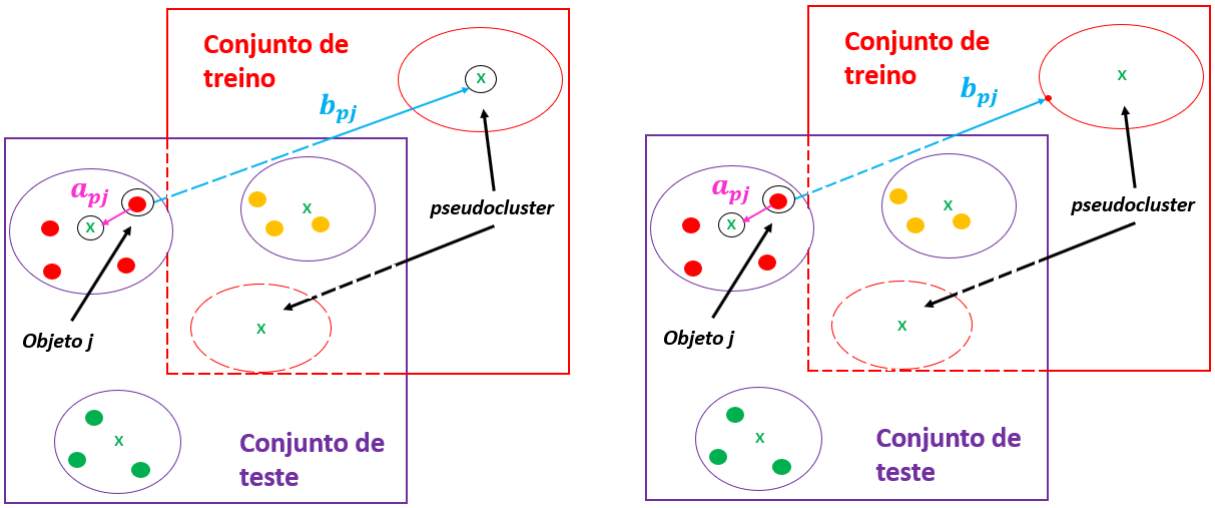
Figura 16 – Abordagem da Silhueta Baseada em Instância para a detecção de novidades.



Fonte: Elaboração própria.

A hipótese é que em casos onde um grupo p do conjunto de treino tem uma grande extensão do centro até a sua borda, o método da Figura 15 tende a gerar resultados inconsistentes, pois, mesmo que a distância até o centro de p seja grande, culminando em maior valor de s_{ij}, há chances do objeto j em avaliação pertencer ao grupo p devido a extensa área que ocupa no espaço de busca. Assim esta versão do critério de silhueta procura levar em consideração a extensão dos grupos da partição de treino no momento de calcular b_{pj} para evitar tal equívoco, com base

Figura 17 – Representação dos critérios de a) Silhueta Baseada em Centroide e b) Silhueta Baseada em Instância, adaptadas para o modelo iCaRL.



a) b)

Fonte: Elaboração própria.

em certos objetos (instâncias) dentro desses grupos. Daí sua denominação como Silhueta Baseada em Instância.

Entretanto, para o caso 2 (modelos incrementais), foi necessário realizar uma adaptação em ambos os critérios de silhueta, para que não seja necessário manter armazenado todas as amostras de treino para obtenção dos *clusters*, o que inviabiliza o aprendizado incremental (requerido para um algoritmo *Open World*). Para o modelo iCaRL (Seção 2.3.3.3), que foi utilizado no componente (3) da Figura 14 para a execução dos experimentos em contexto *Open World*, as adaptações foram realizadas baseadas (novamente) nos trabalhos de Masud et al. (2010), Masud et al. (2011).

Em termos práticos, o que ocorre é que os dados de treino passaram pelo seguinte procedimento nos experimentos do caso 2: antes de se definir os conjuntos de exemplares do iCaRL, logo na primeira iteração do processo discutido na Seção 3.1, o conjunto de treino é agrupado em diversos *clusters*, via algum modelo de agrupamento (aqui foi utilizado *K-means*). Em seguida, é armazenado em um *buffer* P_s uma tupla com as informações de centro e raio de cada *cluster* gerado, sendo estas tuplas chamadas de *pseudoclusters*. A informação de cada *pseudocluster* passa então a ser utilizada nos cálculos de ambas as versões do critério de silhueta, substituindo os objetos de treino originais. E quando uma classe nova é detectada, as instâncias da nova classe são utilizadas para formar um novo *pseudocluster*, sendo posteriormente adicionado em P_s. Assim, o critério de Silhueta Baseada em Centroide e Silhueta Baseada em Instância foram executadas conforme as Figuras 17 a) e 17 b), respectivamente.

Como se pode notar, para o critério de Silhueta Baseada em Centroide (em uma implementação incremental), as definições de a_{pj} e b_{pj} permaneceram iguais aos apresentados na Figura 15. Enquanto que para o critério de Silhueta Baseada em Instância o termo b_{pj}, neste outro contexto, passa a se referir à distância do objeto j até a borda do *pseudocluster* mais próximo F^* no conjunto de treino, dado pela Equação 31 onde R_{F^*} e C_{F^*} são o raio e centroide, respectivamente, de F^*.

$$b_{pj} = \mathbf{d}(C_{F^*}, j) - R_{F^*} \tag{31}$$

A partir dessas metodologias, ambos os critérios de silhueta foram avaliados na detecção de novas classes, considerando tanto classificadores convencionais (nos experimentos iniciais) quanto classificadores incrementais (nos experimentos de cenário *Open World*).

4 METODOLOGIA DA PESQUISA

Neste capítulo são brevemente descritos, na Seção 4.1, os materiais necessários para a realização desta pesquisa. O *dataset* usado nos experimentos e suas diferentes representações são, respectivamente, detalhados nas Seções 4.2 e 4.3. São também apresentadas, na Seção 4.4, as configurações utilizadas no *framework* proposto e, por fim, a Seção 4.5 define as medidas de avaliação de desempenho aplicadas nos experimentos.

4.1 Materiais

Em termos de equipamentos para a condução e avaliação dos procedimentos propostos no Capítulo 3, foi disponibilizado pela UNESP, um Servidor de experimentos de alto desempenho Dell PowerEdge R740 com Linux. Este equipamento, com 32Gb de memória RAM e 24 processadores Intel Xeon (Skylake IBRS) de 2.1GHz, foi principalmente usado para se realizar o treinamento e avaliação dos modelos tratados neste trabalho. Um Notebook também foi usado para a confecção de códigos e escrita de relatórios, artigos e demais artefatos científicos.

Os recursos de *software* utilizados envolvem a IDE PyCharm[1]. A partir da linguagem de programação Python[2] adotada, são empregadas diversas bibliotecas, tais como *tensorflow* (ABADI et al., 2015), *scikit-kearn* (BUITINCK et al., 2013). O código construído para implementar o *framework* proposto pode ser acessado em <https://github.com/luizcoletta/fce-open-world-crop-classification>.

4.2 Dataset

Os *datasets* utilizados nos experimentos foram obtidos a partir do uso de um VANT equipado com câmera RGB, realizando voo sobre uma cultura permanente de Eucaliptos com focos da doença *Ceratocystis wilt*. A doença se trata de um fungo que se aloja inicialmente nas raízes do vegetal e então avança em direção ao caule, alcançando regiões mais altas. À medida que o fungo se multiplica, são gerados bloqueios no sistema de transporte de água e, como consequência, sintomas de murcha, amarelecimento, seca de folhas, além de cancro e necrose do caule surgem com o tempo ocasionando, por fim, na morte da planta. A *Ceratocystis wilt* é considerada uma das doenças mais danosas em plantações de Eucalipto.

A Figura 18 apresenta a imagem aérea coletada e usada neste trabalho, a qual possui dimensão de 4608 × 3456 pixels. Esta imagem foi divida em subimagens de dimensão

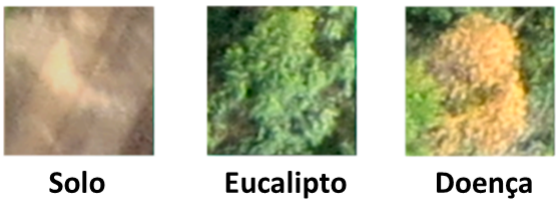
[1] https://www.jetbrains.com/pt-br/pycharm/download/
[2] https://www.python.org/

10 x 10 pixels, sendo rotuladas manualmente em uma das três classes: *Solo*, *Planta Saudável* (Eucalipto) e *Doença* (Eucalipto com *Ceratocystis wilt*). A Figura 19 apresenta exemplos para cada uma das classes. Com estas subimagens foi gerada uma coleção de *datasets*-base, com objetos selecionados aleatoriamente sem reposição, com tamanhos de 1%, 2%, 5%, 10% e 20% do conjunto original.

Figura 18 – Imagem capturada por drone em que os círculos vermelhos destacam algumas árvores de eucalipto doentes.

Fonte: Coletta et al. (2022).

Figura 19 – Classes que constituem o *dataset* em avaliação.



Fonte: Adaptado de Coletta et al. (2022).

A Tabela 1 apresenta a composição de cada *dataset*-base em relação ao número de amostras, além da distribuição dos objetos entre as classes.

Nota-se que a quantidade de objetos da classe 3 de interesse foi mantida para todos os *datasets*-base gerando desbalanceamento considerável em alguns casos. A intensidade deste desbalanceamento é apresentada na última coluna da tabela, a qual contém o Nível de Desbalanceamento de Classe (NDC) da categoria doença em relação as outras

Tabela 1 – Descrição de cada *dataset-base* obtido pelo processo de amostragem dos dados originais.

	Objetos	Classe-1 (Solo)	Classe-2 (Planta saudável)	Classe-3 (Doença)	NDC
dataset-1	689	247	247	195	1:2,5
dataset-2	1183	494	494	195	1:5,1
dataset-5	2667	1236	1236	195	1:12,7
dataset-10	4070	1402	2473	195	1:19,9
dataset-20	6543	1402	4946	195	1:32,6

Fonte: Adaptado de Coletta et al. (2022).

duas. Então, por exemplo, para o *dataset-5*, há um total de 2472 objetos das classes 1 e 2 enquanto 195 são da classe 3. Portanto, para cada amostra da classe 3 há aproximadamente 13 objetos pertencentes às classes 1 e 2.

Esta desproporção entre as classes procura simular o que ocorre no mundo real, onde a maior parte da imagem capturada é composta de plantas saudáveis com alguns pontos de foco da ameaça em questão (como doenças, plantas invasoras e pragas).

Ao aplicar os *datasets*-base nos experimentos, cada objeto visual (subimagem 10x10 pixels) é representado por um vetor numérico de características (*features*), o qual é submetido para treinamento do modelo e classificação. A extração de *features* a partir de uma imagem se deu de três formas: *hand-crafted*, *deep features* ou variáveis latentes extraídas com VAE. A forma de representação em cada um desses espaços é discutida em mais detalhes na Seção 4.3.

4.3 Espaço de Características

No artigo de Coletta et al. (2022), características foram extraídas dos dados com o intuito de reduzir a sua dimensionalidade e foi esta versão dos dados que serviu de insumo para os modelos aqui testados.

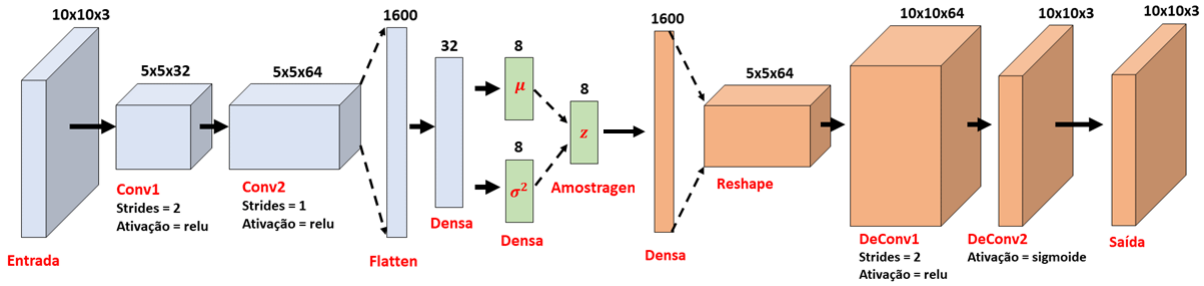
As *features* foram extraídas de três formas diferentes no total, gerando os seguintes conjuntos de dados:

- ***Hand-crafted features*** (SOUZA et al., 2015): este conjunto foi construído por meio de características de cor e textura extraídas manualmente das amostras, constituindo assim os dados com *hand-crafted features*. Para obtenção do formato final, estas *features* passaram por redução de dimensionalidade. Esta redução foi realizada por intermédio do filtro CFS (*Correlation-based Feature Selection*) e posteriormente os valores gerados foram normalizados entre 0 e 1.

- ***Deep features*:** neste caso aplicou-se o modelo de Redes Neurais Profundas VGGNet-16 treinada com ImageNet[3] para extrair automaticamente as características, formando representações chamadas de *deep features*. Para as *deep features*, a redução das dimensões foi alcançada com o emprego do método não-linear de mapeamento isométrico. Maiores detalhes a respeito das representações *deep features* podem ser consultados no trabalho original de Coletta et al. (2022).

- **Variáveis latentes:** diz respeito a um conjunto de dados gerado a partir de *features* extraídas via Autoencoder Variacional. Aqui as características geradas se tratam de variáveis latentes que possuem forma Gaussiana, mas o vetor final de características é composto apenas pela média de cada variável latente. Esta outra forma de representação das amostras foi escolhida uma vez que muitos trabalhos recentes (INáCIO; IZBICKI; GYIRES-TóTH, 2021; BERGAMIN et al., 2022) mostraram que o uso de VAE no processo de detecção de padrões desconhecidos nos dados, tende a gerar melhores resultados. Os vetores latentes gerados pelo VAE foram configurados para assumirem dimensão já reduzida, não demandando nenhum tratamento posterior.

Figura 20 – Estrutura do Autoencoder Variacional implementado, onde μ e σ^2 representam, respectivamente, a média e variância das variáveis latentes.



Fonte: Elaboração própria.

A estrutura do VAE foi implementada aqui com Redes Neurais Convolucionais, seguindo como modelo uma estrutura disponibilizada pelo livro Michelucci (2018). O código se encontra no ambiente *Google Colaboratory* (Colab) [4] para uso livre. A estrutura do *encoder* e *decoder* aplicados nos experimentos são apresentadas na Figura 20.

Para treinamento do VAE, as imagens coloridas 10x10, que compõe os *datasets* utilizados neste trabalho, tiveram seus valores de pixel normalizados para o intervalo

[3] https://www.image-net.org/

[4] <https://colab.research.google.com/github/toelt-llc/astroml-hackdays/blob/master/1%20-%20Autoencoders/code/Variational%20Autoencoders.ipynb>

[0,1] e foram organizadas em *batches* de 15 objetos para otimização dos pesos da rede. O modelo foi configurado para fornecer 8 variáveis latentes na saída do *encoder* (representando o vetor de características das amostras) e o treinamento se deu em 100 épocas.

Em todos os espaços de *features*, o vetor final de características assumiu tamanho 8. Com estas diferentes representações foi possível analisar qual espaço de *features* é mais interessante para encontrar a nova classe.

4.4 Configurações do Modelo

O *framework* proposto neste trabalho foi avaliado por meio de dois bancos de dados: MNIST e aqueles apresentados na Tabela 1. O conjunto MNIST é formado por 70.000 imagens de dígitos manuscritos de 0 a 9, sendo muito empregado na análise de algoritmos de visão computacional (UNGARBAYEV; DEMIREL; AKHTAR, 2022; RONY et al., 2019; ISLAM et al., 2017). Este *dataset* foi utilizado aqui para realizar uma análise inicial sobre o *framework*, permitindo uma inspeção visual dos objetos selecionados pelos critérios apresentados na Seção 3.2. Já os *datasets* da Tabela 1, permitiram verificar o desempenho do algoritmo sobre uma aplicação mais realista, uma vez que os dados apresentam desequilíbrio entre as classes, como normalmente ocorre em uma tarefa do mundo real.

Algumas configurações foram definidas igualmente em todos os experimentos, independentemente do banco de dados. Por exemplo, todos os *datasets* foram divididos em conjuntos de treinamento e teste na proporção de 20% e 80%, respectivamente. O conjunto de treinamento possui as amostras rotuladas responsáveis por construir o modelo de classificação, enquanto o conjunto de teste foi usado para avaliar o desempenho do algoritmo. Para obter resultados mais consistentes, o algoritmo foi treinado e avaliado em um mesmo *dataset* cinco vezes por meio do método de Validação Cruzada Estratificada *K-Fold* (BREIMAN et al., 1984), com $K = 5$, onde a cada vez um conjunto de treinamento inicial diferente é levado em consideração na indução do modelo.

Como visto na Seção 3.1, o *framework* atua ciclicamente sobre os dados, i.e, formando iterações. Nos experimentos, o número de iterações do algoritmo foi limitado a 10. Por fim, juntamente com os critérios de entropia e silhueta, foi empregado um método de seleção aleatória como *baseline* nos experimentos, a fim de verificar se os critérios aqui testados realmente apresentam alguma vantagem ou inteligência sobre uma amostragem puramente estocástica dos objetos.

4.4.1 Procedimento experimental para o conjunto MNIST

Especificamente para o MNIST, o modelo *manifold t-Distributed Stochastic Neighbor Embedding* (t-SNE) (MAATEN; HINTON, 2008) foi aplicado com o objetivo de obter um espaço de *features* em que as classes ficassem bem separadas umas das outras. Em seguida, uma classe foi escolhida para ser a novidade a ser aprendida e, então, essa classe foi excluída do conjunto de treino enquanto permaneceu no conjunto de teste. O *framework* foi aplicado sobre este *dataset* modificado e, para cada uma das dez iterações, foram obtidas imagens do espaço de *features* mostrando os objetos que foram selecionados como potenciais amostras de uma nova classe, por cada um dos critérios apresentados na Seção 3.2. Este procedimento foi executado para cada classe do MNIST, onde uma quantidade de 100 amostras por iteração foram escolhidas (via critérios de entropia e silhueta) para receberem a informação de rótulo (simulando o oráculo).

Nestes experimentos o *framework* foi equipado com um classificador *Support Vector Machine* (SVM) não-linear com *kernel* RBF e parâmetro C igual a 1, devido a sua simplicidade de implementação e eficácia já demonstrada em vários problemas explorados em artigos da literatura e por ser capaz de fornecer como saída, uma distribuição de probabilidade de classe que é necessária para computação da entropia. Além disso, modelos de agrupamento *K-means* construíram as partições para os cálculos envolvendo os critérios de silhueta (Seção 3.2). O número de agrupamentos em cada partição foi igual ao número de classes presentes em cada conjunto de amostras e todas as distâncias envolvidas para definição dos *clusters* e da silhueta, foram calculadas por meio da distância euclidiana. A formação dos grupos ocorreu a cada iteração do processo de aprendizado incremental, utilizando todos os dados rotulados e não rotulados do problema.

4.4.2 Procedimento experimental para os *datasets* gerados com imagens de VANT

Em relação aos *datasets* da Tabela 1, dois conjuntos de experimentos foram realizados: um com classificadores não-incrementais e outro com classificadores incrementais. Em ambos os casos, a cada iteração, os 5 objetos do conjunto de dados não rotulados com maior potencial de indicar um novo padrão, são selecionados via critérios apresentados na Seção 3.2 para receber a informação de classe. Dessa forma, ao todo, os 50 objetos mais representativos de novas classes foram rotulados e inseridos no conjunto de treino para atualização dos modelos. Além disso, a representação das amostras foi feita considerando os três espaços de *features* apresentados na Seção 4.3.

Com o intuito de tornar o comportamento dos dados não rotulados mais próximo daqueles vindos do mundo real, o novo padrão (classe 3) foi apresentado ao algoritmo da segunda iteração em diante, sendo previamente excluído do conjunto de treino no começo da execução do *framework*. Assim, no início do processo, o modelo desconhece

a existência da doença, o que causa 100% de erro de classificação nesta classe e, com o tempo, espera-se que as amostras da nova classe sejam incrementadas à base de conhecimento do modelo, buscando reduzir gradativamente esse nível de erro.

No primeiro conjunto de experimentos (com modelos não-incrementais), o foco foi analisar os critérios de detecção de novas classes (silhueta e entropia) considerando um contexto mais realista e, portanto, um classificador SVM não-incremental (o mesmo utilizado na Seção 4.4.1) foi aplicado no *framework* nestes experimentos. Os resultados obtidos com esta implementação foram comparados com o algoritmo IC-EDS (Seção 2.3.3) da literatura. O classificador SVM foi também empregado na composição do IC-EDS. Aqui os classificadores são retreinados por completo a cada iteração.

Já no segundo conjunto de experimentos (com modelos incrementais), o intuito foi avaliar o *framework* em contexto *Open World*. Para isso, aplicou-se o modelo incremental iCaRL da Seção 2.3.3.3, como classificador, e tanto o critério de silhueta quanto de entropia foram empregados sobre o iCaRL, conforme apresentado na Seção 3.2. Para estes experimentos, o classificador *Open World* NNO da literatura (Seção 2.3.3.2) foi utilizado para efetuar a comparação dos resultados, sendo necessário normalizar as representações dos dados entre [0,1] para todos os espaços de *features*, antes de ser utilizado.

Na próxima seção são apresentadas as métricas de desempenho aplicadas na avaliação dos experimentos e também a forma de análise dos resultados.

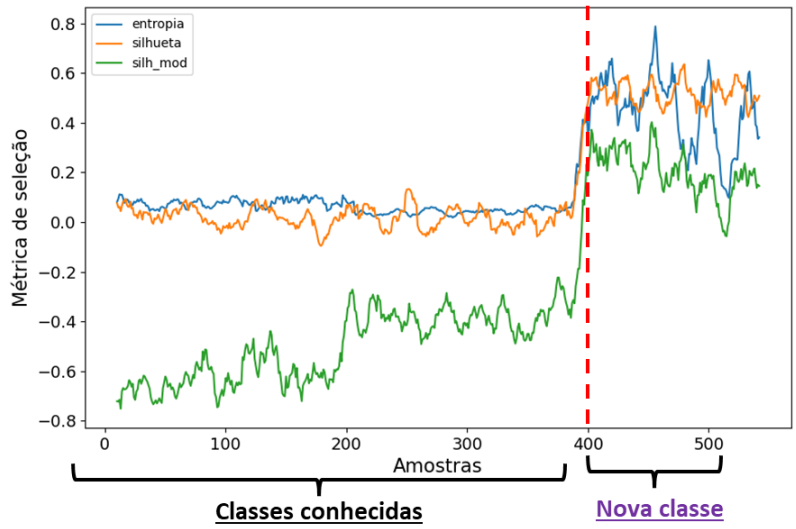
4.5 Métricas de avaliação dos resultados

A análise dos resultados dos experimentos foi conduzida de forma qualitativa e também quantitativa. Aplicando os dois métodos obteve-se resultados para cada *dataset* e também resultados gerais envolvendo todos os experimentos. Um experimento consistiu no uso de um *dataset* (MNIST ou aqueles da Tabela 1), com os objetos representados em um dos espaços de *features* (t-SNE, *hand-crafted*, *deep features* ou variáveis latentes), sobre o qual foi aplicado o procedimento de aprendizagem incremental da Seção 3.1, seguindo as configurações apresentadas na Seção 4.4 para cada caso.

Entre os resultados qualitativos que serão apresentados no Capítulo 5, um gráfico foi gerado exibindo o valor das medidas de entropia e silhueta para cada objeto não rotulado no conjunto de teste. A hipótese é que para as amostras de uma nova classe a curva sofre uma mudança de perfil, assumindo amplitudes distintas daquelas relacionadas a objetos de categorias já conhecidas. A Figura 21 demonstra o raciocínio apresentado. Para gerar os gráficos, tal como o da Figura 21, foi necessário ordenar as amostras de acordo com a classe.

Em relação aos resultados quantitativos, foram calculadas as medidas de acurácia, pela Equação 32, e *F1-score*, pela Equação 35, para o problema multiclasse. Este último é frequentemente visto como uma combinação das métricas de precisão dada pela

Figura 21 – Gráfico destacando a variação dos critérios de detecção de novas classes quando um novo padrão surge nos dados.



Fonte: Elaboração própria.

Equação 33 e *recall* dada pela Equação 34.

$$\text{Acurácia}(y,\hat{y}) = \frac{1}{N_{amostras}} \sum_{i=0}^{N_{amostras}-1} 1(\text{se } y_i = \hat{y}_i) \tag{32}$$

$$\text{Precisão} = \frac{tp}{tp+fp} \tag{33}$$

$$\text{Recall} = \frac{tp}{tp+fn} \tag{34}$$

$$\text{F1-Score} = 2 \times \frac{\text{precisão} \times recall}{\text{precisão} + recall} \tag{35}$$

Onde y e $\hat{y}$ correspondem ao rótulo verdadeiro e rótulo predito, respectivamente, em um cenário multiclasse; tp são os verdadeiros positivos; fp os falsos positivos e fn os falsos negativos.

Para o problema multiclasse, os valores de *F1-score* são calculados para cada classe de forma independente e depois, então, combinados para gerar o resultado final. Essa combinação pode ser feita de diferentes maneiras, aqui foi aplicado o método *"weighted"* da biblioteca *scikit-learn* que calcula o valor final ao encontrar o valor médio entre os resultados para cada classe ponderado por um valor de suporte (número de instâncias verdadeiras) (SCIKIT-LEARN, 2011). Este procedimento leva em conta o desbalanceamento das classes.

No Capítulo 5 a seguir são apresentados os resultados desta pesquisa.

5 RESULTADOS E DISCUSSÃO

Nas próximas seções são apresentados e discutidos os principais resultados obtidos com os experimentos descritos na Seção 4.4. A Seção 5.1 é dedicada aos experimentos conduzidos com o conjunto MNIST, enquanto a Seção 5.2 aborda a atuação dos critérios de seleção em uma tarefa mais realista considerando, inicialmente, o uso de classificadores não-incrementais. E por fim, os resultados da aplicação do *framework* proposto em contexto *Open World* é mostrado na Seção 5.3.

5.1 Experimentos Ilustrativos Sobre o Conjunto MNIST

Na Figura 22 é apresentado o comportamento dos critérios de detecção de novas classes aplicados em comparação com um método de seleção aleatória, usado como *baseline*, sobre o conjunto de dados MNIST onde a classe do dígito 1 é a novidade a ser aprendida (região laranja). Os pontos pretos nas imagens representam as amostras selecionadas pelos critérios no conjunto de teste, para receberem os rótulos verdadeiros (simulando o oráculo) sendo, posteriormente, utilizadas para atualizar o classificador conforme explicado na Seção 3.1.

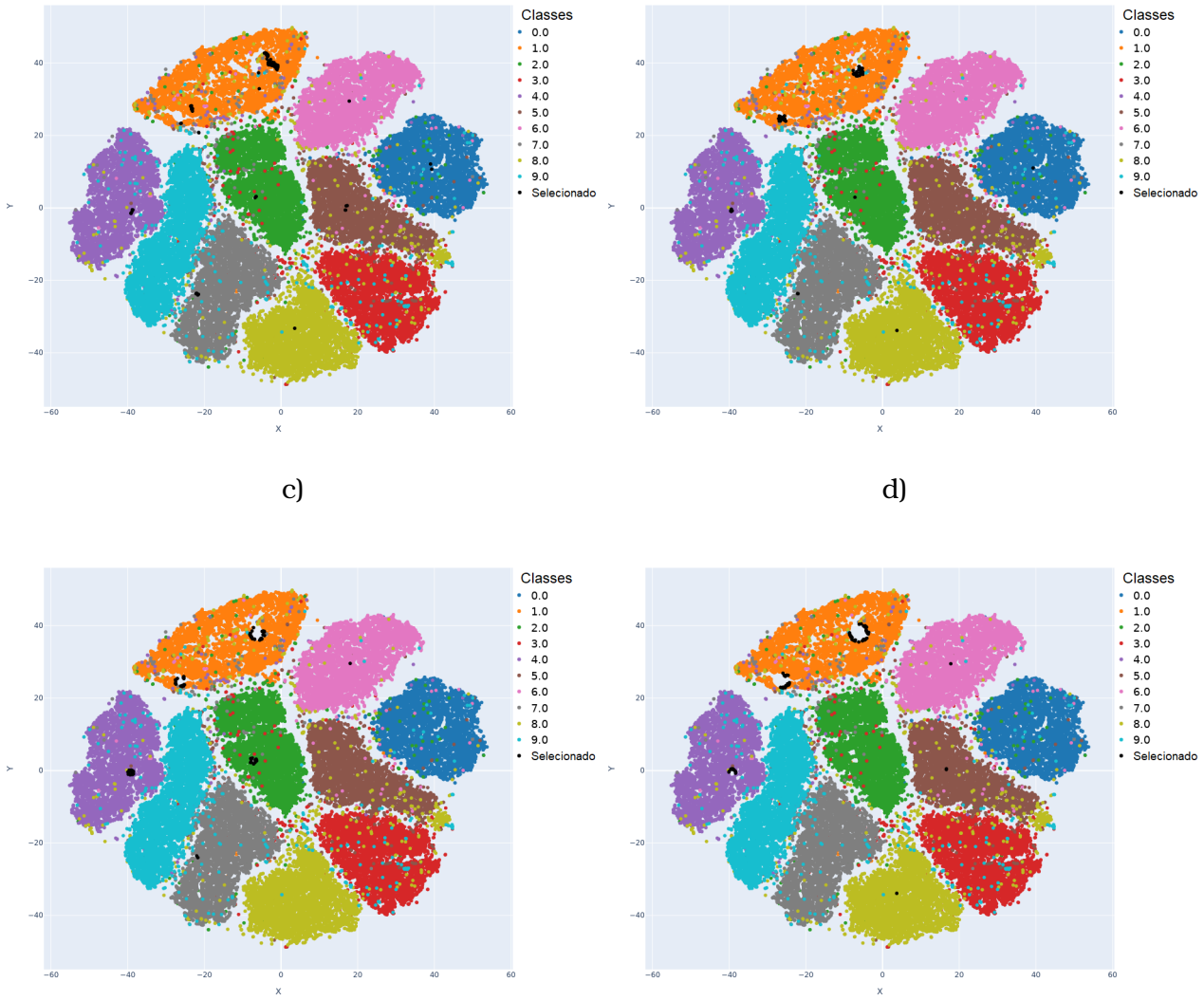
Na Figura 22 a) é mostrado o *baseline* adotado, onde nota-se que não há preferência (viés) na seleção das instâncias, i.e, o conjunto escolhido tende a ser balanceado para as classes existentes. Na Figura 22 b) é mostrado como atua o critério de entropia, onde a maioria dos objetos selecionados residem próximos ao limite de decisão entre as classes, uma vez que é a região com maior incerteza na classificação. Assim, a entropia seleciona amostras na borda da região onde há uma nova classe. Esse mesmo comportamento também ocorreu para todas as outras classes, quando foram retiradas do conjunto de treinamento para realização dos experimentos, exceto para as classes de dígitos 3 e 4, onde a entropia encontrou maior dificuldade para detectá-las como a nova classe.

Por outro lado, a Silhueta Baseada em Instância na Figura 22 c) seleciona amostras de forma dispersa dentro da região da nova classe, enquanto o critério de Silhueta Baseada em Centroide na Figura 22 d), tende a selecionar objetos próximos à região central da nova classe. Esse comportamento pode ser visto novamente nas Figuras 22 e) e 22 f) que mostram a progressão da etapa da Figura 22 d). Assim como em um processo de mineração, instâncias relevantes da nova classe são retiradas e levadas para o reconhecimento do modelo causando aberturas que, a cada iteração, aumentam no interior da classe. Este mesmo comportamento também foi observado para todas as outras classes, exceto para a classe 9 onde ambos os métodos de silhueta encontraram maiores dificuldades para detectá-la como um novo padrão.

O comportamento observado para ambas as formas de silhueta pode ser explicado através das suas respectivas definições matemáticas. Assim, para a Silhueta Baseada em Instância, a seleção do objeto leva em consideração o contorno externo dos *clusters* na partição de treinamento, de modo que quando esta borda for irregular as amostras escolhidas parecerão espalhadas aos olhos do observador. Já no caso da Silhueta Baseada em Centroide, quanto mais próximo o objeto estiver do centro de seu *cluster* na partição de teste, menor será o valor do termo a_{pj} na Equação 30 e, portanto, o resultado s_j é maximizado. Esta é a razão do comportamento visto nas imagens das Figuras 22 d) a 22 f).

Figura 22 – Comportamento dos critérios de detecção de novas classes no conjunto de dados MNIST, onde o dígito 1 é a novidade a ser aprendida. O *baseline* é apresentado em a), o critério de entropia em b), a Silhueta Baseada em Instância em c) e de d) a f) são apresentados o critério de Silhueta Baseada em Centroide nas iterações 1,2 e 4, respectivamente.

a) b)



c) d)

e) f)

Fonte: Elaboração própria.

Dessa forma, vê-se que com exceção do *baseline* na Figura 22 a), nas demais imagens há uma preferência por selecionar objetos da região laranja, onde se encontra a nova classe. Esse contraste nos mostra, visualmente, que os critérios empregados de fato direcionam o *framework* para detectar e aprender novidades no conjunto alvo, permitindo, assim, acompanhar as mudanças na distribuição.

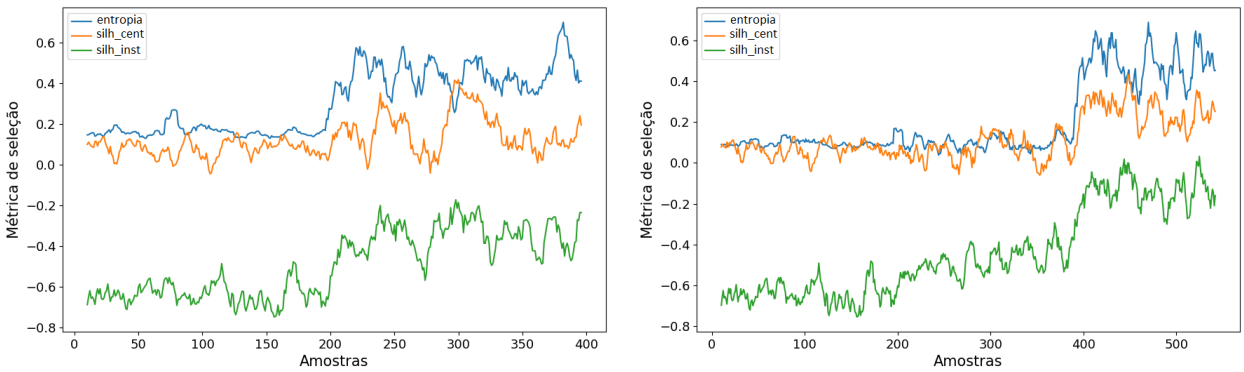
5.2 Avaliação dos Critérios de Detecção de Novidades Sobre Uma Tarefa do Mundo Real

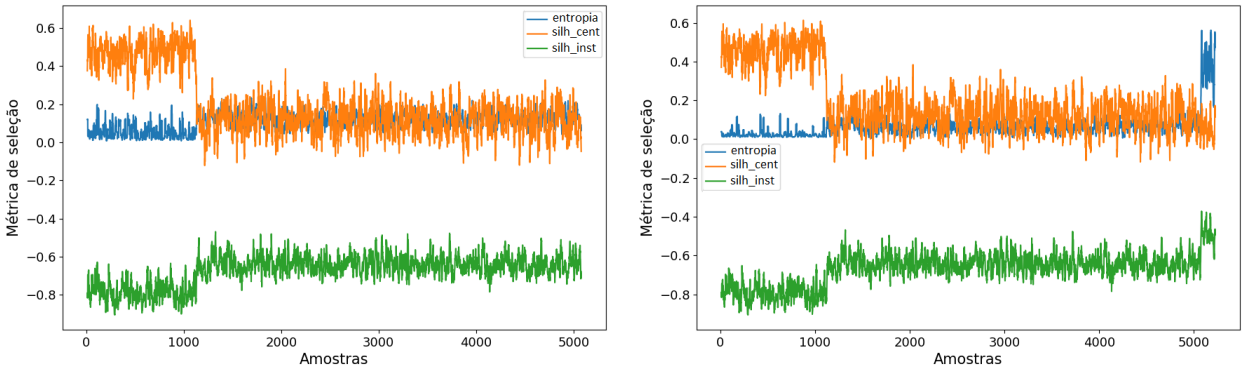
Nesta seção e na próxima, as análises foram conduzidas sobre os *datasets* da Tabela 1. Com o intuito de organizar a nomenclatura dos diferentes *datasets* utilizados nos experimentos, o seguinte padrão foi estabelecido: o *dataset-1*, *dataset-2* e outros da Tabela 1 que foram codificados via *deep features* passaram a ser chamados, respectivamente, de *dp_ceratocystis1*, *dp_ceratocystis2* e assim por diante. Da mesma maneira, quando representados com *hand-crafted features* serão chamados de *hc_ceratocystis1*, etc. Aqueles sob uso de variáveis latentes receberão as denominações *vae_ceratocystis1*, etc. Além disso, a Silhueta Baseada em Centroide, proposta na Seção 3.2, será referida nos gráficos e tabelas pelo nome *silh_cent*, enquanto que a Silhueta Baseada em Instância será nomeada como *silh_inst*.

Dito isso, os gráficos das Figuras 23 a 25 apresentam os valores obtidos para cada critério de detecção de novas classes sobre o conjunto de teste, ao buscar a existência de novas classes, seguindo o que foi descrito na Seção 4.5 (Figura 21). As curvas mostram o comportamento destes critérios, imediatamente antes e depois do aparecimento do novo padrão que ocorre na iteração 2. A Figura 23 apresenta estes gráficos para os *datasets* dp_ceratocystis1 e dp_ceratocystis20, a Figura 24 diz respeito ao hc_ceratocystis1 e hc_ceratocystis20 e a Figura 25 aos *datasets* vae_ceratocystis1 e vae_ceratocystis20. Os *datasets-1/20* representam, respectivamente, as situações de menor e maior desbalanceamento entre as classes, i.e., os casos mais extremos.

Nos experimentos relacionados ao *dataset-1*, nota-se que os critérios assumiram valores mais altos na presença de uma nova classe (porção final da curva a partir do valor

Figura 23 – Perfil dos critérios de detecção de novos padrões sobre os dados de teste: a) na iteração 1 e b) na iteração 3 do conjunto dp_cerotocystis1; c) na iteração 1 e d) na iteração 3 do conjunto dp_cerotocystis20.

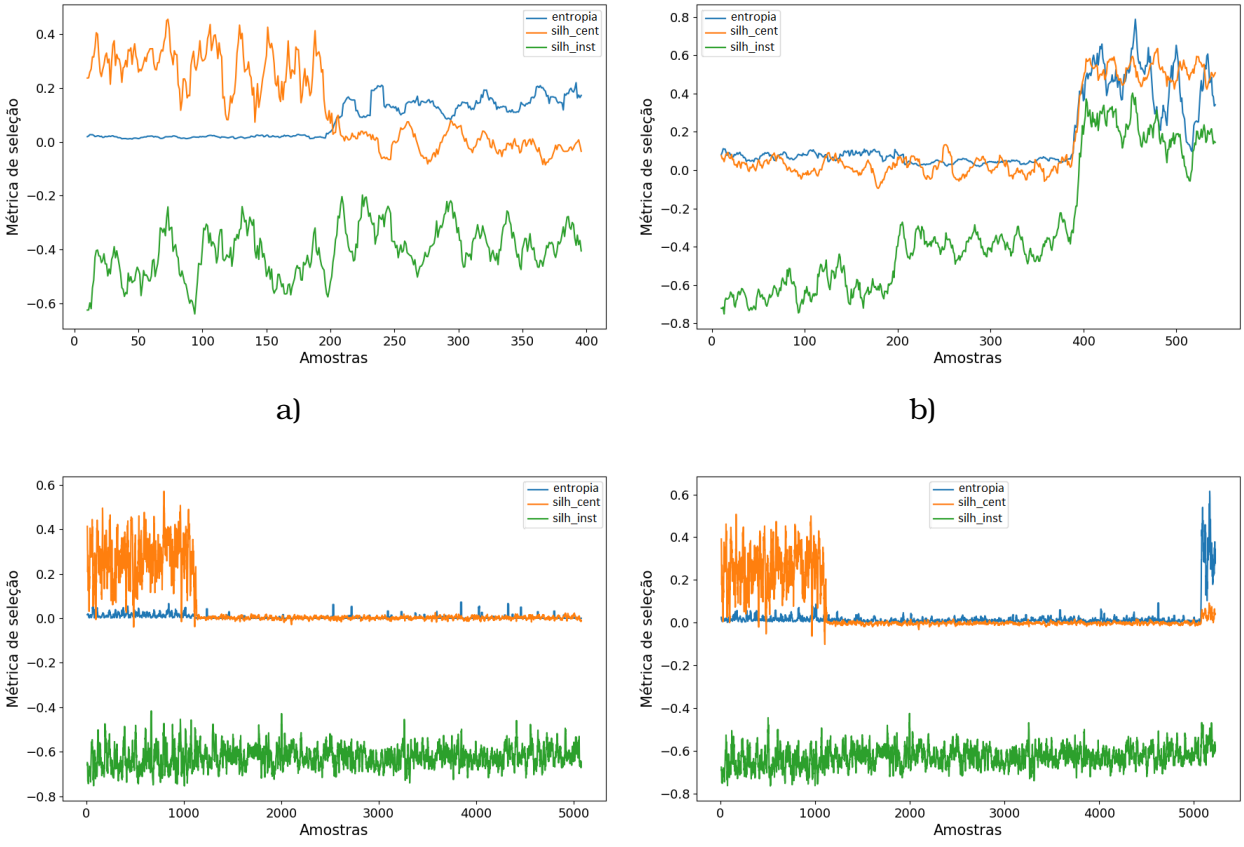




c) d)

Fonte: Elaboração própria.

Figura 24 – Perfil dos critérios de detecção de novos padrões sobre os dados de teste: a) na iteração 1 e b) na iteração 3 do conjunto hc_cerotocystis1; c) na iteração 1 e d) na iteração 3 do conjunto hc_cerotocystis20.

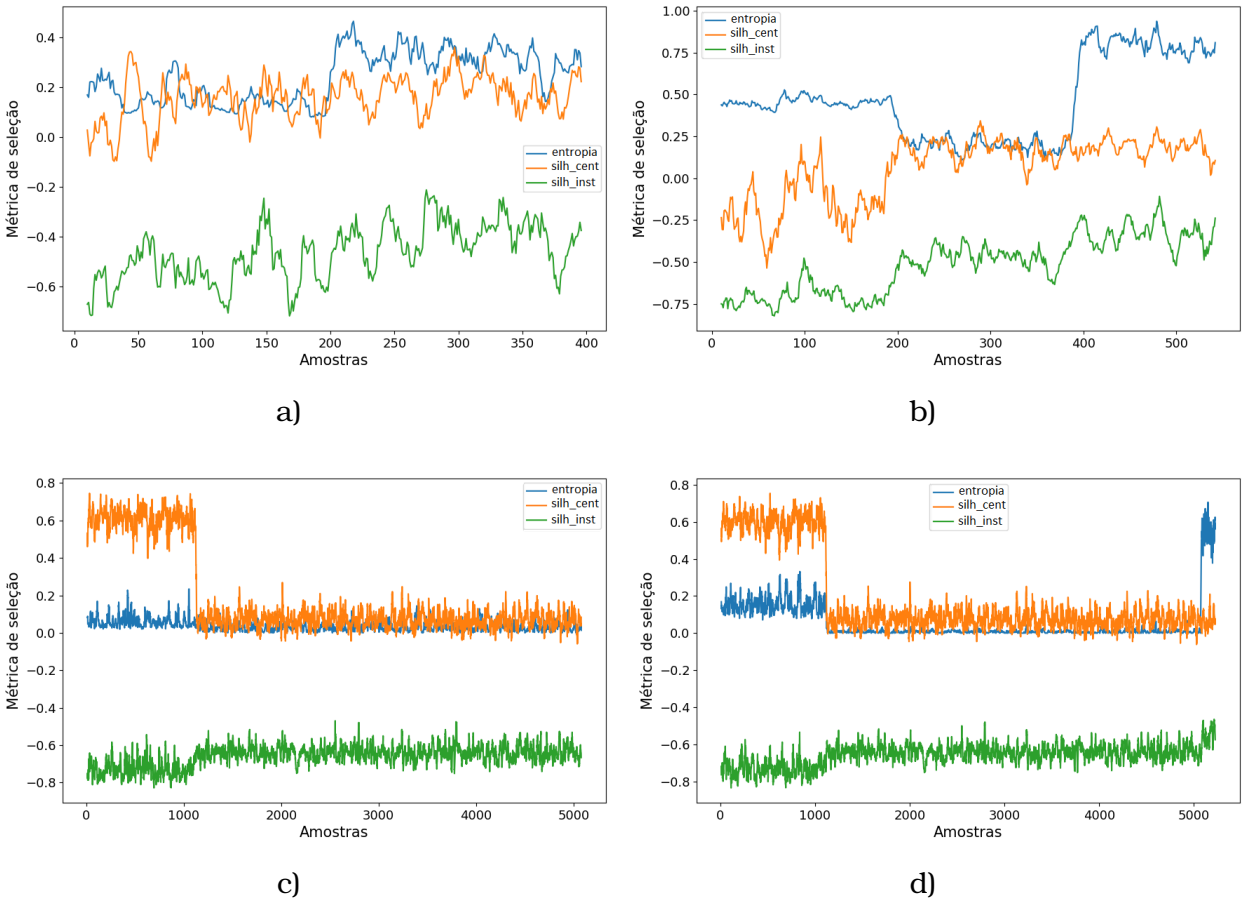


a) b)

c) d)

Fonte: Elaboração própria.

Figura 25 – Perfil dos critérios de detecção de novos padrões sobre os dados de teste: a) na iteração 1 e b) na iteração 3 do conjunto vae_cerotocystis1; c) na iteração 1 e d) na iteração 3 do conjunto vae_cerotocystis20.



Fonte: Elaboração própria.

400 do eixo x, na terceira iteração). Em especial as *features hand-crafted* pareceram demonstrar maior sensibilidade a novos padrões em todos os critérios testados, enquanto que para as demais *features* a entropia gerou melhores resultados, principalmente na representação em variáveis latentes por VAE onde ambos os critérios de silhueta exibiram dificuldade para detectar a chegada do novo padrão.

Sobre o *dataset-20* (classe nova surge a partir de $x = 5000$) observa-se que a entropia apresenta também uma boa performance em todos os casos, enquanto que a Silhueta Baseada em Instância apresenta alguma variação na presença do novo padrão nas Figuras 23 d) e 25 d). Já a Silhueta Baseada em Centroide não demonstra variação significativa. Para o *dataset-20* vê-se que o espaço de características fez pouca diferença, gerando perfis muito similares em todos os casos com destaque à entropia que mais uma vez apresentou melhores resultados. Por fim, os gráficos mostram também que o uso de

critérios baseados em distância para detectar novos padrões, tem seu desempenho deteriorado à medida que o desbalanceamento aumenta.

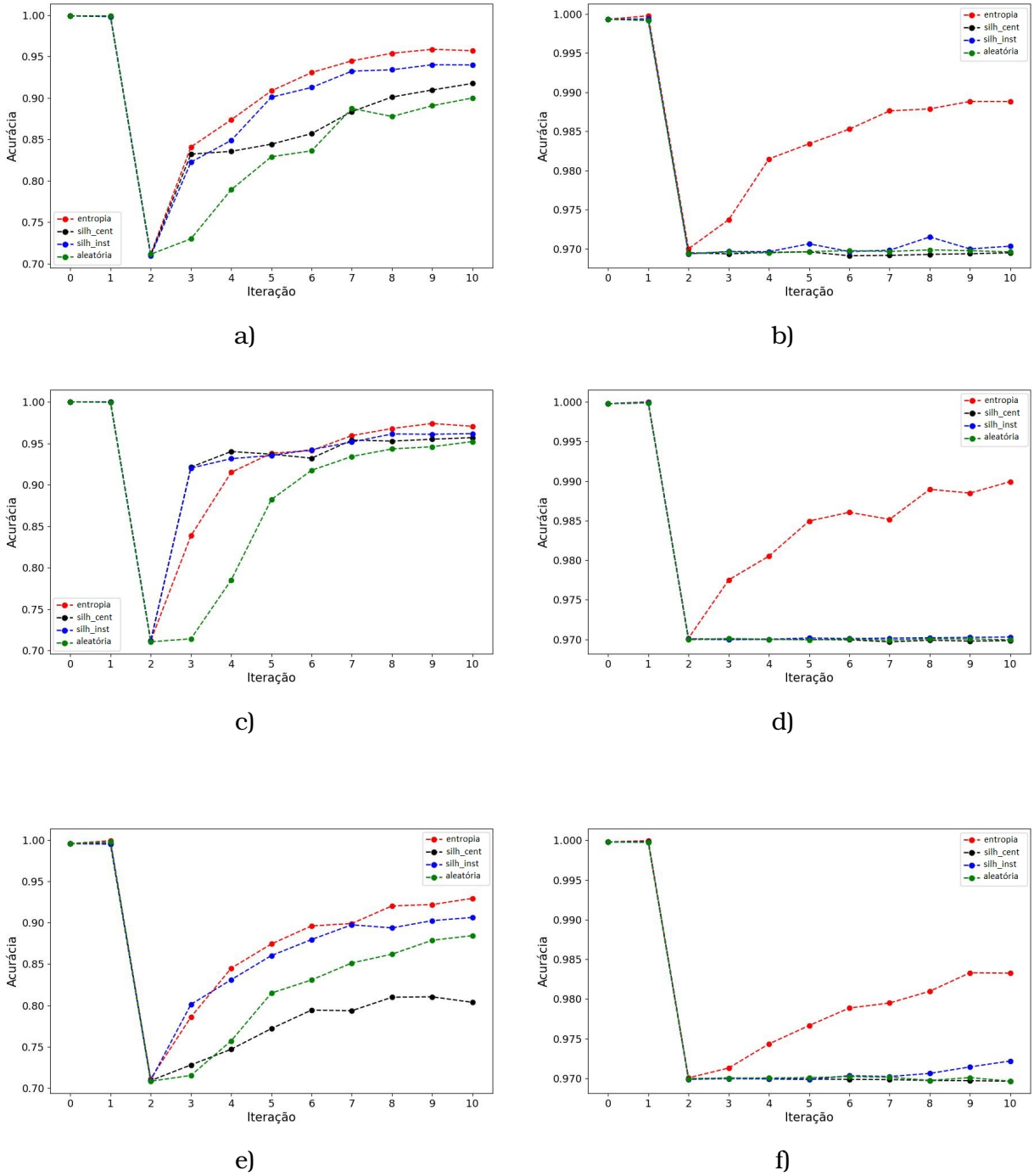
Uma crítica ao critério de entropia é que apesar da sua sensibilidade em detectar objetos fora dos padrões conhecidos pelo modelo permitir o encontro de novas classes, essa mesma habilidade pode por engano considerar os ruídos, i.e. *outliers*, como objetos que devem ser avaliados pelo oráculo. Assim, com o intuito de prevenir tal comportamento indesejado é importante um pré-processamento dos dados no sentido de eliminar o máximo de ruídos possível.

Na Figura 26 são apresentadas as curvas de acurácia do *framework* ao longo das 10 iterações, sobre todas as classes, para os mesmos *datasets* considerados nas Figuras 23, 24 e 25. Em todos os casos percebe-se que inicialmente a medida atinge valor próximo de 100%, porém na iteração 2 sofre uma queda brusca. Isso ocorreu pois neste momento surge o novo padrão nos dados (classe 3 - Doença), antes nunca apresentado no conjunto de treino e por isso então o classificador comete erros na predição, degradando a acurácia. A partir da terceira iteração a nova classe é identificada e amostras dela são inseridas no conjunto de treino com seu devido rótulo para retreinar o classificador SVM. Consequentemente, o modelo aprende o novo padrão e gradativamente a acurácia cresce.

Para o *dataset-1* (Figura 26 a), 26 c) e 26 e)), os critérios implementados se mostraram mais eficientes do que uma simples seleção aleatória, com destaque para a entropia que alcançou níveis maiores de acurácia seguido da Silhueta Baseada em Instância. Com o uso de variáveis latentes como espaço de *features*, a Silhueta Baseada em Instância obteve resultados melhores que a seleção aleatória enquanto que a Silhueta Baseada em Centroide teve desempenho pior do que as demais.

Em relação ao *dataset-20* (Figura 26 b), 26 d) e 26 f)), os resultados foram semelhantes entre todos os espaços de *features*, onde apenas a entropia foi capaz de recuperar parte da acurácia perdida ao surgir uma nova classe enquanto os outros critérios falharam. Isso demonstra novamente a maior robustez da medida de entropia sobre dados desbalanceados, sendo o *dataset-20* o caso mais extremo desse desbalanceamento entre os conjuntos considerados. O mau desempenho das medidas de silhueta pode ter ocorrido por conta da existência de sobreposição entre as classes no espaço de *features*, gerando confusão nos critérios.

Figura 26 – Acurácia sobre todos os dados não-rotulados para a) dp_ceratocystis1, b) dp_ceratocystis20, c) hc_ceratocystis1, d) hc_ceratocystis20, e) vae_ceratocystis1 e f) vae_ceratocystis20.



Fonte: Elaboração própria.

A partir deste ponto serão discutidos os resultados gerais envolvendo todos os experimentos executados. Nas Figuras 27 e 28 são apresentados, respectivamente, os mapas

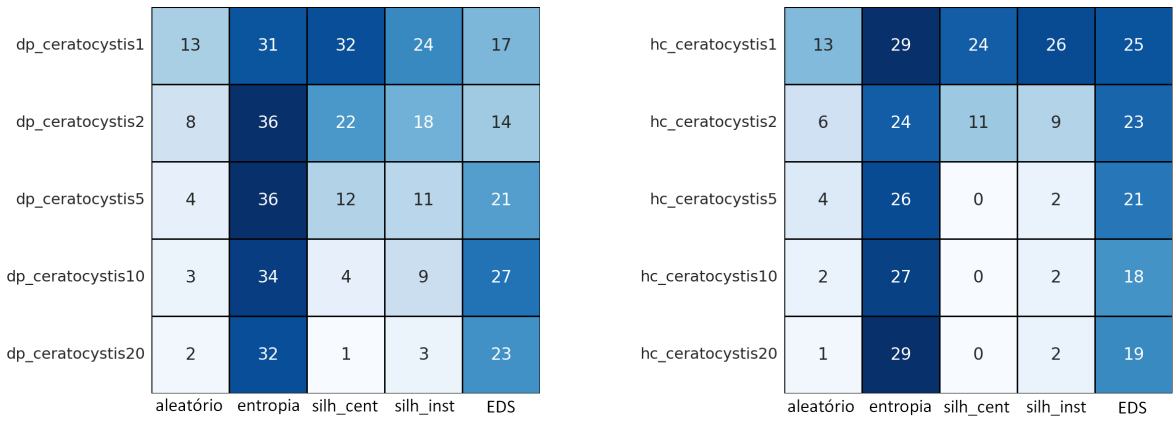
de calor da quantidade de amostras selecionadas e da acurácia de classificação, apenas sobre a nova classe, ao final das 10 iterações para todos os experimentos onde diferentes comportamentos foram observados entre os três espaços de *features* aqui considerados.

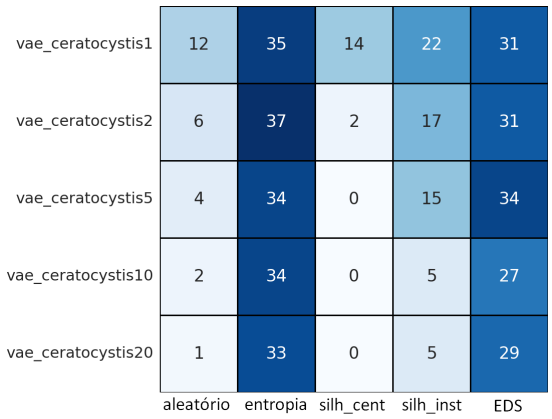
Especificamente para *deep features*, os mapas de calor das Figuras 27 a) e 28 a) destacam o método da entropia como o melhor critério testado para o *framework* proposto, uma vez que seleciona a maior quantidade de instâncias da nova classe entre as 50 amostras coletadas nos experimentos (para atualização do modelo) e também porque obteve maior taxa de acerto na nova classe. Ambos os resultados superaram os obtidos com a seleção aleatória e o método EDS da literatura.

Para os dois critérios de silhueta, os resultados superaram os obtidos com a seleção aleatória em quase todos os *datasets*, mas notou-se que o desempenho de ambos é cada vez mais deteriorado à medida que o desbalanceamento aumenta. Considerando a variação de cor na Figura 28 a) para o critério de Silhueta Baseada em Centroide, pode-se ver que seu comportamento foi semelhante ao *baseline* de seleção aleatória, enquanto a Silhueta Baseada em Instância foi melhor principalmente nos *datasets-5/20*. Em comparação com EDS, apenas a Silhueta Baseada em Instância obteve resultados competitivos, exceto para *dp_ceratocystis-20*.

Para *hand-crafted features*, nas Figuras 27 b) e 28 b) o critério de entropia foi novamente o melhor entre os elencados na Seção 3.2, mas apenas superou o *baseline* de seleção aleatória. Enquanto que a Silhueta Baseada em Centroide e a Silhueta Baseada em Instância foram piores do que a seleção aleatória e o EDS em quase todos os *datasets*, exceto para os *datasets-1/2* comparando com a seleção aleatória. Assim, este espaço de *features* foi apenas favorável à entropia, no entanto, o IC-EDS apresentou melhor desempenho (Figura 28 b)).

Figura 27 – Mapas de calor da quantidade de amostras selecionadas da nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes.





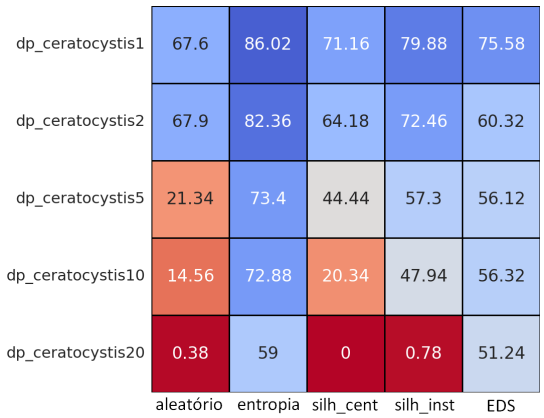
c)

Fonte: Elaboração própria.

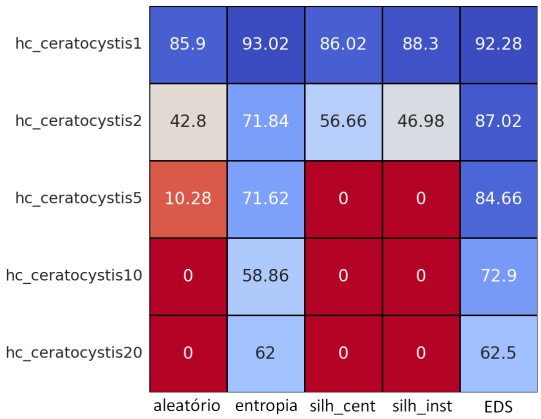
Por fim, nas Figuras 27 c) e 28 c), a entropia teve melhor desempenho do que a seleção aleatória e o IC-EDS para o espaço de variáveis latentes, enquanto o critério de Silhueta Baseada em Centroide, desta vez, foi o pior em ambos os casos, tanto para o número de novidades selecionadas quanto para a acurácia na nova classe, onde na maioria dos casos nenhuma amostra do novo padrão foi selecionada. Por outro lado, a Silhueta Baseada em Instância selecionou uma quantidade considerável de objetos vindos da nova classe, apresentando um desempenho melhor do que o *baseline* de seleção aleatória. Mas, não superou o método EDS da literatura. E como observado nos casos anteriores, seu desempenho foi se deteriorando à medida que o desequilíbrio aumentava.

A Tabela 2 abaixo mostra os resultados de *F1-score* de cada experimento, ao final das 10 iterações, considerando todas as classes do problema juntas.

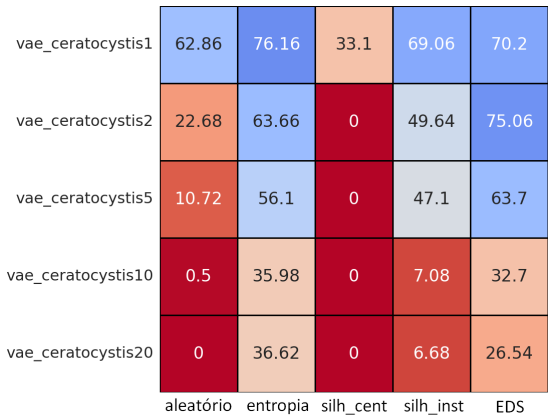
Figura 28 – Mapas de calor da acurácia de classificação (%) apenas sobre a nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes.



a)



b)



c)

Fonte: Elaboração própria.

Tabela 2 – Valores de *F1-score* alcançado sobre todas as classes juntas em todos os experimentos.

Dataset	Classificador /Critério	Deep features (%)	Hand-crafted features (%)	Variáveis latentes (%)
Dataset-1	SVM/entropia	**95,63**	97,04	**92,61**
	SVM/silh_cent	91,35	95,62	**76,79**
	SVM/silh_inst	93,81	96,13	90,22
	SVM/aleatória	**89,40**	**95,13**	87,61
	IC_EDS/EDS	92,43	**97,11**	91,14
Dataset-2	SVM/entropia	**96,67**	94,77	94,19
	SVM/silh_cent	93,58	91,59	**74,84**
	SVM/silh_inst	94,97	88,68	90,12
	SVM/aleatória	93,76	**87,13**	82,66
	IC_EDS/EDS	**91,43**	**97,13**	**95,48**
Dataset-5	SVM/entropia	**98,11**	97,88	96,80
	SVM/silh_cent	95,30	88,99	**88,90**
	SVM/silh_inst	96,42	**88,75**	95,42
	SVM/aleatória	**91,85**	90,51	90,57
	IC_EDS/EDS	96,68	**98,53**	**97,25**
Dataset-10	SVM/entropia	**98,45**	98,05	**96,84**
	SVM/silh_cent	94,68	**92,75**	**92,76**
	SVM/silh_inst	97,08	92,88	93,68
	SVM/aleatória	**94,22**	92,85	92,91
	IC_EDS/EDS	97,88	**98,42**	96,40
Dataset-20	SVM/entropia	**98,76**	**98,88**	**98,01**
	SVM/silh_cent	**95,47**	**95,50**	**95,51**
	SVM/silh_inst	95,62	95,57	96,08
	SVM/aleatória	95,52	95,53	95,53
	IC_EDS/EDS	98,46	98,77	97,62

Fonte: Elaboração própria.

As cores azul e vermelha destacam os melhores e piores resultados, respectivamente, para cada *dataset* de acordo com a representação utilizada. Conclusões semelhantes às das Figuras 27 e 28 podem ser tiradas da tabela acima, onde os valores em vermelho nos mostram como o critério de Silhueta Baseada em Centroide tende a gerar os piores resultados à medida que a dificuldade do problema aumenta. O critério de entropia obteve melhores resultados na maioria dos experimentos, sendo robusto a diferentes níveis de desbalanceamento de classe, bem como a diferentes espaços de *features*. Já a Silhueta Baseada em Instância foi mais robusta ao desbalanceamento do que o critério de Silhueta Baseada em Centroide. Mas ambas mostraram sensibilidade à representação utilizada, onde a última mostrou-se pior que ambos os métodos usados para comparação como se pode ver, por exemplo, na quarta e quinta coluna para os *datasets-10/20*.

No que diz respeito às representações, o uso de *hand-crafted features* gerou melhores resultados de acurácia (Figura 28) e *F1-score* (Tabela 2) via IC-EDS, mas acabou por ser muito restritivo para a detecção de novidades com bom desempenho apenas para critérios baseados em entropia. Portanto, as características geradas com *Deep Learning* foram mais favoráveis aos critérios aqui testados, permitindo a detecção de mais ocorrências de *Ceratocystis wilt* (Figura 27). O uso do espaço de variáveis latentes gerados com VAE, resultou nos menores índices de acurácia e *F1-score* entre todos os experimentos, enquanto que a representação com *deep features* atingiu desempenho competitivo em relação aos melhores observados com *hand-crafted features*, com a vantagem de serem gerados automaticamente por redes neurais, evitando trabalho manual.

5.3 Avaliação do *framework* em cenário *Open World*

A Tabela 3 apresenta o valor final de *F1-score* obtido nos experimentos envolvendo os modelos NNO e iCaRL (Seção 4.4.2), após as 10 iterações sobre todas as classes do problema. Os valores em azul e vermelho destacam, respectivamente, os melhores e piores resultados para cada *dataset* em cada tipo de representação. Analisando estes resultados verificou-se que, de forma geral, o desempenho das implementações em cada *dataset* ficaram muito próximas uma das outras onde as maiores diferenças entre o melhor e o pior caso ocorreram com mais frequência nos *datasets* que são, originalmente, menos desbalanceados.

Um aspecto importante desses experimentos que vale a pena relembrar, é que os modelos NNO e iCaRL são do tipo incrementais, ou seja, ao longo das iterações os classificadores não revisitam os dados de treinamento. Ao invés disso, cada um adota uma forma de representação dos conhecimentos passados (o NNO utiliza médias de classe e o iCaRL conjuntos de exemplares de cada classe), que é usado em segundo plano pelo modelo enquanto é dado enfoque no aprendizado de novidades. Por consequência, o desbalanceamento das classes permanece apenas nos dados de teste enquanto que

nos dados de treinamento essa característica deixa de existir neste outro contexto.

Tabela 3 – Valores de *F1-score* alcançado sobre todas as classes juntas em todos os experimentos.

Dataset	Classificador /Critério	Deep features (%)	Hand-crafted features (%)	Variáveis latentes (%)
Dataset-1	iCaRL/entropia	**89,65**	**97,52**	86,68
	iCaRL/silh_cent	88,67	95,59	83,96
	iCaRL/silh_inst	86,60	96,24	**89,60**
	iCaRL/aleatória	86,55	96,06	87,81
	NNO	**84,57**	**95,89**	**74,15**
Dataset-2	iCaRL/entropia	91,42	**97,60**	91,60
	iCaRL/silh_cent	91,21	96,93	92,04
	iCaRL/silh_inst	**92,47**	97,49	**92,26**
	iCaRL/aleatória	90,02	96,69	89,32
	NNO	**86,90**	**95,73**	**85,65**
Dataset-5	iCaRL/entropia	92,37	**97,79**	91,00
	iCaRL/silh_cent	94,19	96,94	92,83
	iCaRL/silh_inst	**96,81**	97,44	**93,77**
	iCaRL/aleatória	95,36	96,80	**90,94**
	NNO	**92,06**	**96,76**	92,58
Dataset-10	iCaRL/entropia	91,31	97,18	90,73
	iCaRL/silh_cent	**91,15**	97,08	92,46
	iCaRL/silh_inst	92,60	**97,25**	92,84
	iCaRL/aleatória	91,30	**96,73**	**90,65**
	NNO	**92,74**	96,82	**92,98**
Dataset-20	iCaRL/entropia	**90,39**	96,81	**93,01**
	iCaRL/silh_cent	90,90	97,05	93,55
	iCaRL/silh_inst	92,79	97,01	94,65
	iCaRL/aleatória	90,71	**96,10**	95,40
	NNO	**96,22**	**98,48**	**97,13**

Fonte: Elaboração própria.

Isso pode explicar a semelhança dos valores de *F1-score* entre as diferentes implementações nos experimentos, já que as classes são representadas de forma mais clara na base de conhecimento dos modelos. Apesar disso, na Tabela 3 vê-se que o modelo NNO apresentou desempenho ligeiramente inferior ao iCaRL nos *datasets-1* a *5* enquanto alcançou resultados ligeiramente melhores nos *datasets-10/20* que apresentam maior desbalanceamento. Em relação às implementações envolvendo iCaRL, o uso de diferentes critérios de detecção de novas classes fizeram pouca diferença significativa, resultando em desempenhos próximos ao *baseline* de seleção aleatória.

Pela Tabela 3 é possível avaliar apenas o resultado final do processo de aprendizagem incremental, mas não é possível verificar como se deu a recuperação do desempenho perdido após o aparecimento da nova classe. Por conta disso, as Figuras 29, 30 e 31 foram

geradas e exibem o comportamento dos modelos incrementais em todos os experimentos ao longo das 10 iterações, apresentando em cada uma delas a acurácia de classificação atingida sobre todas as classes no conjunto de teste.

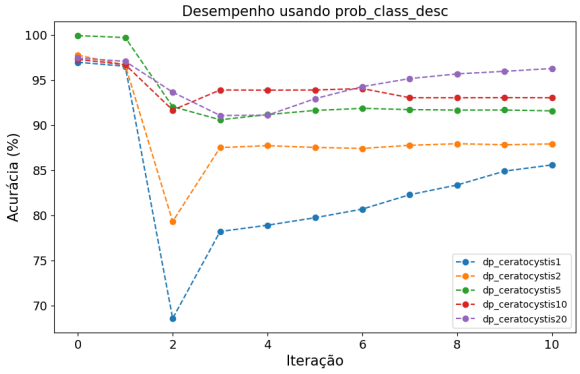
Na Figura 29, para a representação em *deep features*, todas as implementações conseguiram melhorar o desempenho do modelo após a iteração 2 nos casos *dp_ceratocystis1* e *dp_ceratocystis2* sendo que o modelo NNO (Figura 29 a)) resultou nos piores índices de acurácia. Para os demais *datasets*, apenas a implementação envolvendo iCaRL e Silhueta Baseada em Instância (Figura 29 d)) apresentou desempenho melhor que o NNO e a seleção aleatória (Figuras 29 a) e 29 b)), enquanto que a implementação de iCaRL e entropia (Figura 29 c)) foi a que encontrou maiores dificuldades na recuperação do desempenho. Por fim, o desempenho do critério de Silhueta Baseada em Centroide na Figura 29 e) se aproximou daquele observado com o uso de seleção aleatória.

Por outro lado, para *hand-crafted features* na Figura 30 os resultados envolvendo uso de entropia (Figura 30 c)) e silhueta (Figuras 30 d) e 30 e)) foram muito similares entre si, onde ambas apresentaram desempenho melhor que iCaRL com seleção aleatória (Figura 30 b)) e ligeiramente superiores ao NNO na Figura 30 a), exceto para *hc_ceratocystis20*.

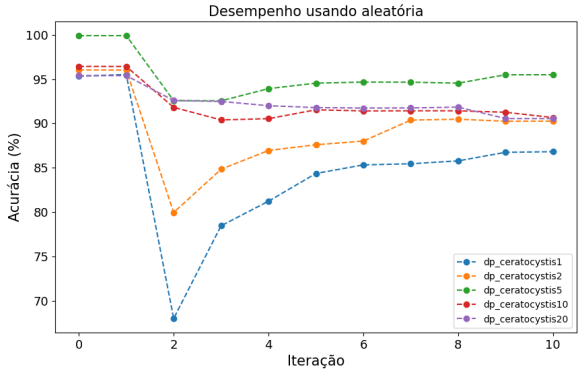
Por fim na Figura 31, o modelo NNO (Figura 31 a)) demonstrou pior desempenho na recuperação da acurácia perdida entre os experimentos com variáveis latentes para *vae_ceratocystis1* e *vae_ceratocystis2*, ao passo que apresentou o melhor desempenho para os demais *datasets* superando os resultados atingidos com iCaRL. Já as implementações com iCaRL e entropia/silhueta (Figuras 31 c) a 31 e)) apresentaram comportamento semelhante ao *baseline* de seleção aleatória (Figura 31 b)), onde apenas foram capazes de recuperar a acurácia nos casos *vae_ceratocystis1* e *vae_ceratocystis2*. Assim de forma geral, o *framework* proposto foi capaz de sintetizar o novo padrão de maneira a recuperar parte da acurácia perdida quando o modelo sofre um distúrbio com o aparecimento da classe 3, de maneira a apresentar desempenho similar ou até mesmo melhor que a seleção aleatória e o NNO.

O último aspecto analisado foi o desempenho unicamente sobre a classe nova a ser aprendida. Com intuito de realizar isso, foram obtidos os mapas de calor das Figuras 32 e 33 que apresentam os resultados de desempenho sobre a classe 3 (*Ceratotystis wilt*) ao final das 10 iterações. Mais especificamente, a quantidade de objetos selecionados da nova classe e a acurácia de classificação sobre a classe 3, respectivamente, para todos os experimentos.

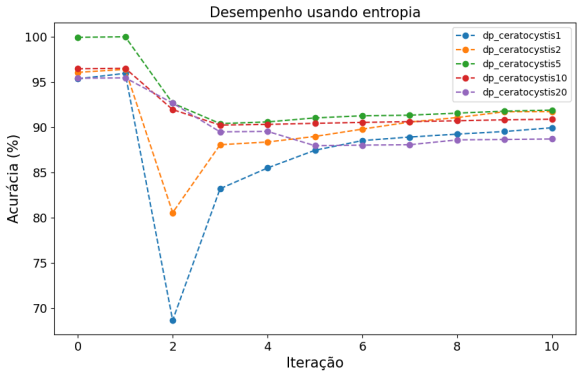
Figura 29 – Gráficos da acurácia sobre todas as classes à cada iteração, para a representação em *deep features* considerando os modelos: a) NNO, b) iCaRL com seleção aleatória, c) iCaRL com entropia, d) iCaRL com Silhueta Baseada em Instância e e) iCaRL com Silhueta Baseada em Centroide.



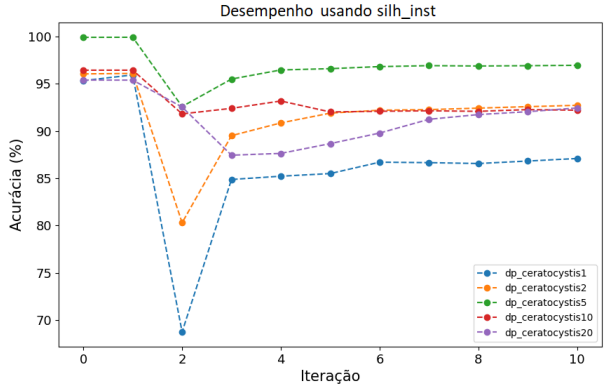
a)



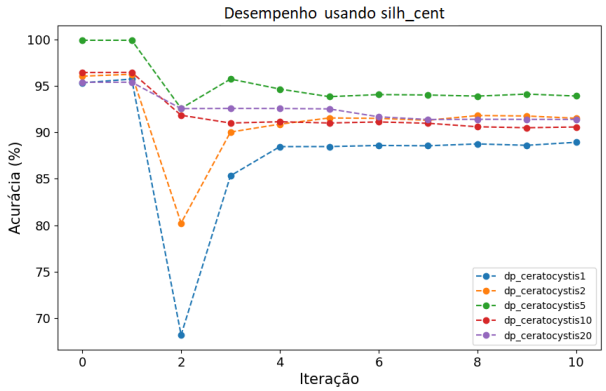
b)



c)



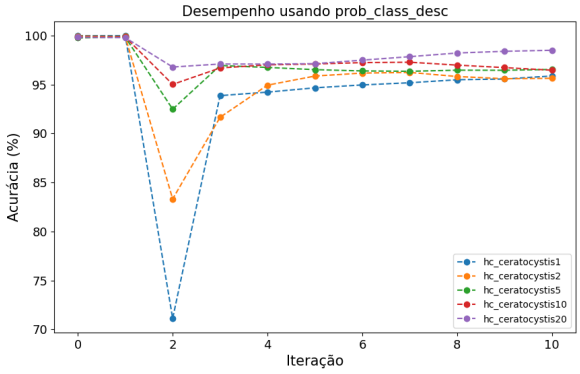
d)



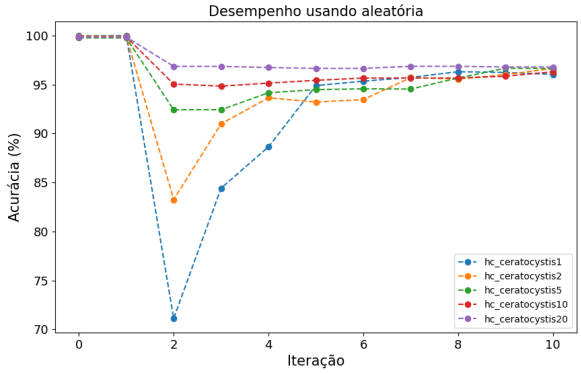
e)

Fonte: Elaboração própria.

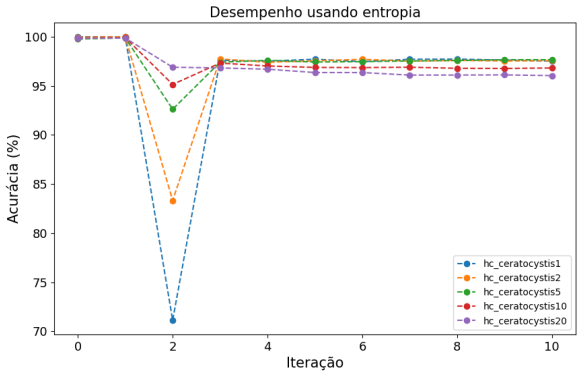
Figura 30 – Gráficos da acurácia sobre todas as classes à cada iteração, para a representação em *hand-crafted features* considerando os modelos: a) NNO, b) iCaRL com seleção aleatória, c) iCaRL com entropia, d) iCaRL com Silhueta Baseada em Instância e e) iCaRL com Silhueta Baseada em Centroide.



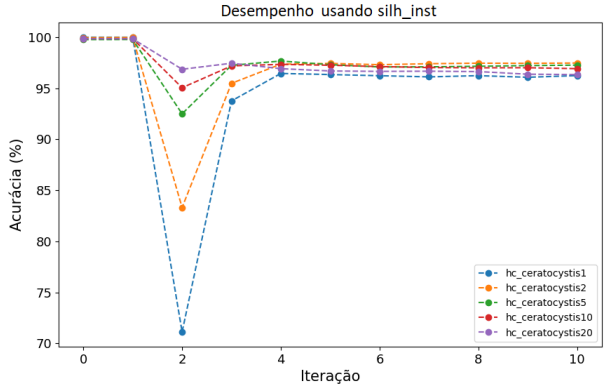
a)



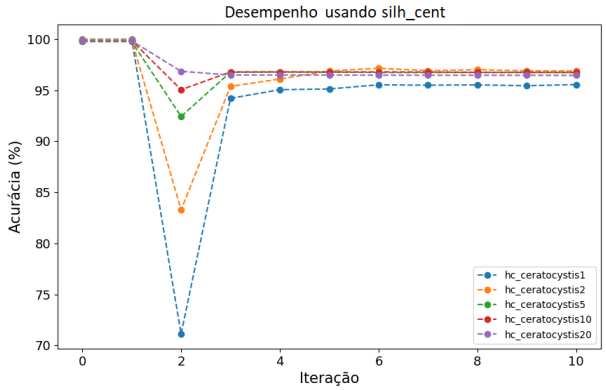
b)



c)



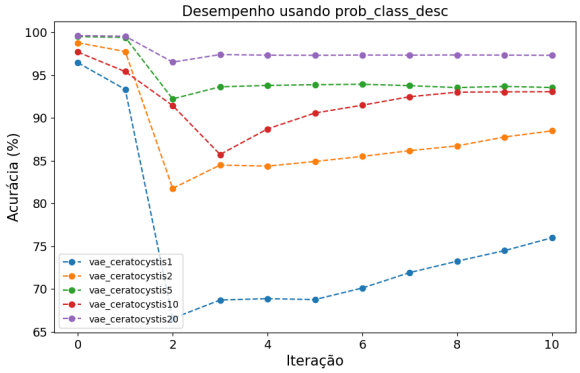
d)



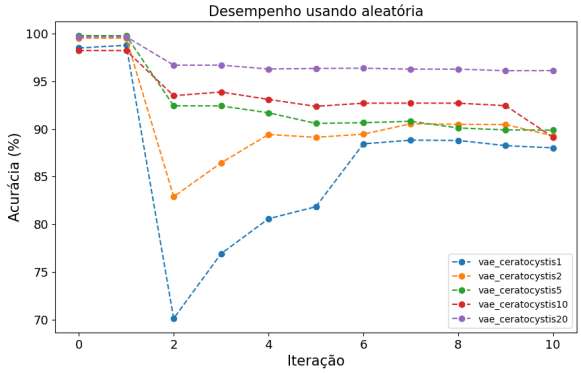
e)

Fonte: Elaboração própria.

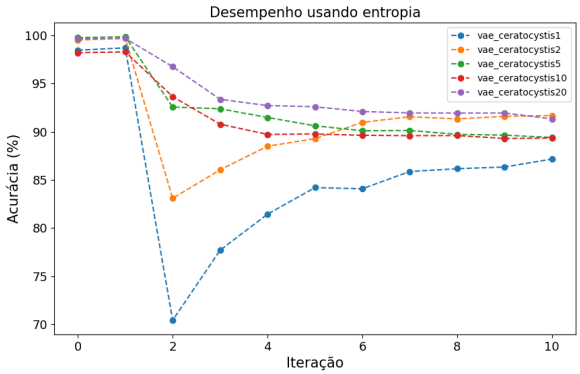
Figura 31 – Gráficos da acurácia sobre todas as classes à cada iteração, para a representação em variáveis latentes com VAE considerando os modelos: a) NNO, b) iCaRL com seleção aleatória, c) iCaRL com entropia, d) iCaRL com Silhueta Baseada em Instância e e) iCaRL com Silhueta Baseada em Centroide.



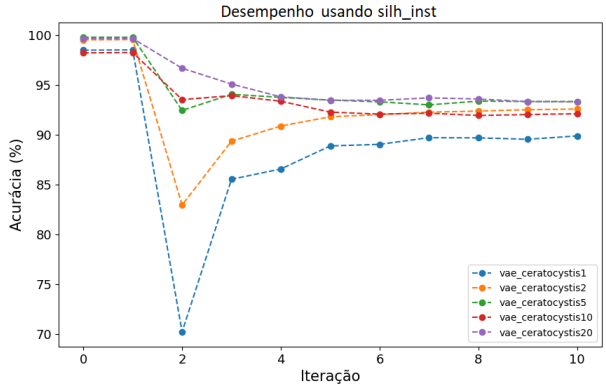
a)



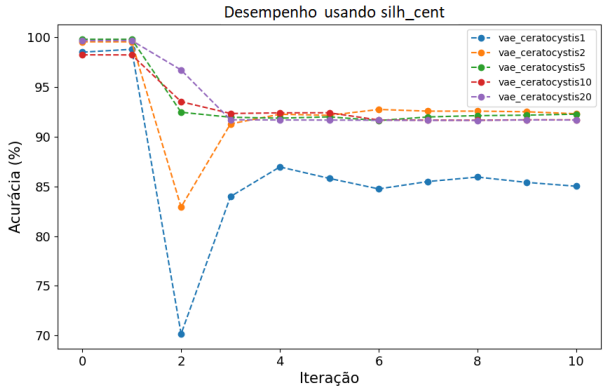
b)



c)



d)



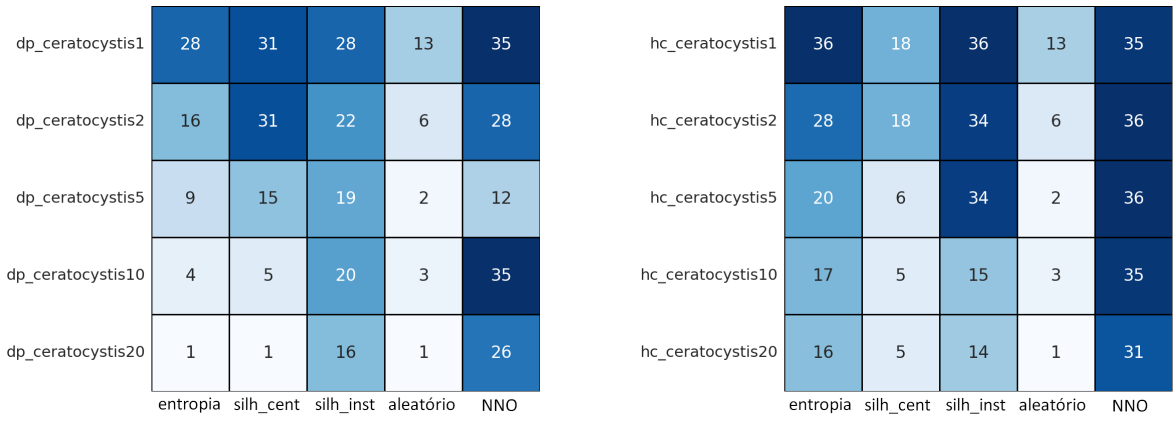
e)

Fonte: Elaboração própria.

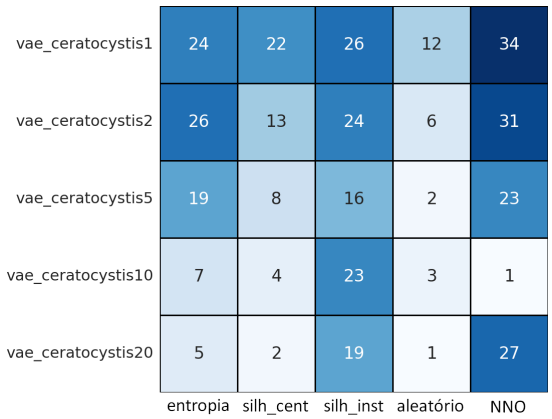
Na Figura 32, nota-se que o NNO foi capaz de detectar a maior quantidade de amostras da novidade em todos os espaços de *features*, enquanto que a implementação de iCaRL com seleção aleatória selecionou a menor quantidade de objetos em todas as representações, sendo que à medida que o desbalanceamento foi aumentando ao longo dos *datasets*, menor foi essa quantidade. No caso da entropia, um comportamento diferente ao observado na Seção 5.2 ocorreu, o critério não foi robusto ao crescimento de desbalanceamento de classes no conjunto de teste para as representações obtidas com Redes Neurais (Figuras 32 a) e 32 c)). Uma possível explicação para isso é que nos classificadores incrementais é aplicada uma forma de representação de conhecimento que busca ser próxima ao conjunto de treino original. Mas na prática essa representação possui limitações de forma que, diante de dados desbalanceados, instâncias de padrões conhecidos podem parecer algo novo para o classificador que não possui uma base de conhecimento complexa o suficiente para reconhecer o objeto, categorizando-o (erroneamente) como novidade. A única exceção ocorre na Figura 32 b), onde a representação *hand-crafted features* aparenta (por algum motivo) ser mais favorável à entropia.

Já a Silhueta Baseada em Centroide, em todos os diagramas, aproximou-se da seleção aleatória (com exceção dos *datasets-1/2*). A maior dificuldade de detecção de novas classes em conjuntos desbalanceados pode estar relacionado às grandes extensões que algumas classes ocupam no espaço de *features* do conjunto de teste, de forma que na Equação 30 a distância b_{pj} até o centro do pseudocluster é muito alta para um certo objeto j nestas classes e, por consequência, o resultado de silhueta também é maior identificando j erroneamente como algo diferente do se conhece. Esse problema é mitigado na Silhueta Baseada em Instância ao considerar a extensão dos pseudoclusters no cálculo do critério. Isso explica a maior robustez observada em todos os mapas de calor superando o *baseline* de seleção aleatória, em termos de quantidade.

Figura 32 – Mapas de calor da quantidade de amostras selecionadas da nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes.



a) b)



c)

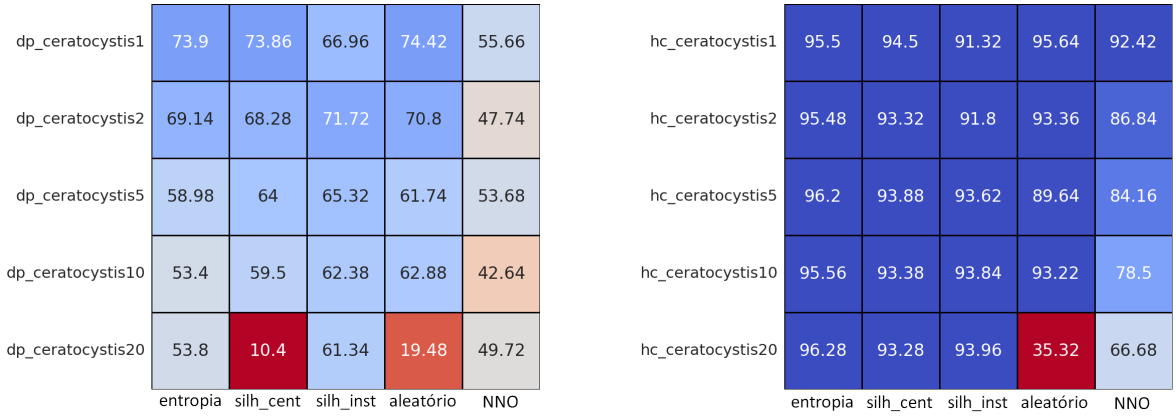
Fonte: Elaboração própria.

Pela Figura 33 foi possível constatar a influência da quantidade de amostras selecionadas da nova classe, sobre o desempenho de classificação do novo padrão. Pelas imagens verificou-se que o modelo NNO da literatura apresentou, de forma geral, os menores índices de acurácia em todos os tipos de representação dos dados. Isso indica que apesar do modelo ter selecionado a maior quantidade de amostras na Figura 32, esses objetos não possibilitaram uma representação de qualidade da nova classe na estrutura interna do modelo.

Por outro lado, o uso de conjuntos de exemplares no iCaRL permitiu que o modelo atingisse um bom nível acurácia na nova classe, mesmo utilizando poucas amostras. Assim os critérios de entropia e de silhueta acabaram apresentando desempenho semelhante à seleção aleatória, sendo que a entropia demonstrou resultados normalmente inferiores ao *baseline* (com exceção do caso da Figura 33 b)). Entretanto para o *dataset* mais desbalanceado (*dataset-20*) fica visível a vantagem dos critérios testados sobre uma mera seleção estocástica.

De forma geral, o uso do *framework* proposto se mostrou superior ao modelo NNO da literatura para o aprendizado de novas classes em cenário *Open World*, em que os critérios aqui testados e propostos na Seção 3.2 são mais eficientes do que uma seleção aleatória, para *datasets* com alto desbalanceamento. Por fim, a representação *handcrafted* dos dados gerou melhor desempenho na detecção e aprendizagem da nova classe com valores de acurácia muitas vezes acima de 90% e com o uso do *framework* foi possível adaptar um modelo incremental já existente (iCaRL) para a construção de um modelo *Open World* eficiente.

Figura 33 – Mapas de calor da acurácia de classificação (%) apenas sobre a nova classe, ao final das 10 iterações para: a) *deep features*, b) *hand-crafted features* e c) variáveis latentes.



Fonte: Elaboração própria.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi proposto um *framework* com componentes customizáveis, para o desenvolvimento de algoritmos capazes de aprender novas classes incrementalmente, em contexto *Open World*, buscando atender as necessidades geradas pelo comportamento natural dos dados, em um problema real do contexto agrícola.

Os resultados obtidos mostraram que o *framework* apresentado é uma alternativa válida para o desenvolvimento de modelos capazes de aprender novos padrões, exibindo desempenho competitivo ou, então, até mesmo melhores do que algoritmos da literatura (NNO e IC-EDS). Em relação aos critérios de detecção de novas classes, foi observado que os métodos de entropia e silhueta introduzem no modelo um viés que o direciona no sentido de encontrar novas classes, sendo que em contexto *Open World*, esse efeito foi mais evidente sobre conjuntos de dados mais desbalanceados, com destaque ao critério proposto de Silhueta Baseada em Instância, que apresentou maior robustez a este tipo de comportamento indesejado.

Ao implementar o modelo iCaRL da literatura como uma solução *Open World* (a partir do *framework* proposto), obteve-se um modelo escalável mais eficaz do que o bem-conhecido NNO, no aprendizado da nova classe, principalmente pelo fato do modelo atingir níveis de acurácia satisfatórios com o uso de apenas algumas amostras rotuladas do novo padrão. Para aplicações do contexto agrícola este é um resultado interessante, pois neste tipo de problema os dados passam por um tratamento manual de rotulação executado por profissionais da área, sendo então um processo custoso em termos de tempo e trabalho. Portanto, é desejável que modelos tenham bom desempenho ao mesmo tempo que dependam da menor quantidade de dados rotulados possível, ainda mais considerando o fato do ambiente agrícola ser muito suscetível ao surgimento de novidades, i.e., demandando continuamente amostras rotuladas para manter o modelo atualizado. Além disso, essa característica se alinha a paradigmas em voga atualmente como *Green AI* (SCHWARTZ et al., 2020), que se preocupa em obter algoritmos menos custosos computacionalmente para reduzir emissões de carbono.

Como principais contribuições do trabalho, pode-se destacar:

- Definição de um *framework* simples para o desenvolvimento de modelos de classificação para problemas *Open World*;
- Demonstração de seu uso na adaptação de modelos incrementais já existentes, para a construção de um modelo *Open World* eficaz (essa demonstração foi realizada através de iCaRL).

Apesar dos resultados obtidos na pesquisa, maiores investigações precisam ser conduzidas a fim de verificar o uso do *framework* em outras configurações e preencher lacunas ainda abertas. Como exemplo, pesquisas podem ser feitas investigando a combinação de dois ou mais critérios de detecção de novas classes, além disso, estudos podem ser feitos analisando o comportamento do *framework* quando mais de uma nova classe aparece ao mesmo tempo nos dados e, por fim, experimentos poderiam ser realizados considerando outros tipos de problemas, como reconhecimento de objetos e segmentação semântica.

BIBLIOGRAFIA

ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANé, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIéGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; ZHENG, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <https://www.tensorflow.org/>.

ACHARYA, A.; HRUSCHKA, E. R.; GHOSH, J.; ACHARYYA, S. An optimization framework for combining ensembles of classifiers and clusterers with applications to nontransductive semisupervised learning and transfer learning. *ACM Trans. Knowl. Discov. Data*, Association for Computing Machinery, New York, NY, USA, v. 9, n. 1, aug 2014. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/2601435>.

ADE, M. R. R.; DESHMUKH, P. R. Methods for incremental learning : A survey. *International Journal of Data Mining Knowledge Management Process*, v. 3, p. 119–125, 07 2013.

ALVES, V.; CAMPELLO, R.; HRUSCHKA, E. Towards a fast evolutionary algorithm for clustering. In: *2006 IEEE International Conference on Evolutionary Computation*. [S.l.: s.n.], 2006. p. 1776–1783.

ANGLUIN, D. Queries and concept learning. *Machine Learning*, v. 2, p. 319–342, 1988.

ARJOVSKY, M.; BOTTOU, L. *Towards Principled Methods for Training Generative Adversarial Networks*. arXiv, 2017. Disponível em: <https://arxiv.org/abs/1701.04862>.

ASAVALERTPALAKORN, S.; SINGHATANADGID, P.; ARDSOMANG, T. Novelty detection of a rolling bearing using long short-term memory autoencoder. In: *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*. [S.l.: s.n.], 2022. p. 1–4.

BAUR, C.; WIESTLER, B.; ALBARQOUNI, S.; NAVAB, N. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In: CRIMI, A.; BAKAS, S.; KUIJF, H.; KEYVAN, F.; REYES, M.; WALSUM, T. van (Ed.). *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Cham: Springer International Publishing, 2019. p. 161–169. ISBN 978-3-030-11723-8.

BELLMAN, R.; CORPORATION, R.; COLLECTION, K. M. R. *Dynamic Programming*. Princeton University Press, 1957. (Rand Corporation research study). ISBN 9780691079516. Disponível em: <https://books.google.com.br/books?id=wdtoPwAACAAJ>.

BENDALE, A.; BOULT, T. Towards open world recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 1893–1902.

BENDALE, A.; BOULT, T. E. Towards open world recognition. *CoRR*, abs/1412.5687, 2014. Disponível em: <http://arxiv.org/abs/1412.5687>.

BENDALE, A.; BOULT, T. E. Towards open set deep networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 1563–1572.

BENOS, L.; TAGARAKIS, A. C.; DOLIAS, G.; BERRUTO, R.; KATERIS, D.; BOCHTIS, D. Machine learning in agriculture: A comprehensive updated review. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 21, n. 11, p. 3758, 2021.

BERGAMIN, L.; CARRARO, T.; POLATO, M.; AIOLLI, F. *Novel Applications for VAE-based Anomaly Detection Systems*. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2204.12577>.

BERNARD, E. *Introduction to Machine Learning*. Wolfram Media, Incorporated, 2021. ISBN 9781579550455. Disponível em: <https://books.google.com.br/books?id=x6K7zgEACAAJ>.

BLEI, D. M.; KUCUKELBIR, A.; MCAULIFFE, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, Informa UK Limited, v. 112, n. 518, p. 859–877, apr 2017. Disponível em: <https://doi.org/10.1080%2F01621459.2017.1285773>.

BOULT, T.; GRABOWICZ, P.; PRIJATELJ, D.; STERN, R.; HOLDER, L.; ALSPECTOR, J.; JAFARZADEH, M.; AHMAD, T.; DHAMIJA, A.; LI, C.; CRUZ, S.; SHRIVASTAVA, A.; VONDRICK, C.; SCHEIRER, W. Towards a unifying framework for formal theories of novelty. In: . [S.l.: s.n.], 2021.

BREIMAN, L.; FRIEDMAN, J.; STONE, C.; OLSHEN, R. *Classification and Regression Trees*. Taylor & Francis, 1984. ISBN 9780412048418. Disponível em: <https://books.google.com.br/books?id=JwQx-WOmSyQC>.

BRUZZONE, L.; FERNÁNDEZ-PRIETO, D. An incremental-learning neural network for the classification of remote-sensing images. *Pattern Recognit. Lett.*, v. 20, p. 1241–1248, 1999.

BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B.; VAROQUAUX, G. API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. [S.l.: s.n.], 2013. p. 108–122.

BUONO, F. D.; CALABRESE, F.; BARALDI, A.; PAGANELLI, M.; GUERRA, F. Novelty detection with autoencoders for system health monitoring in industrial environments. *Applied Sciences*, v. 12, p. 4931, 05 2022.

CARREIRA-PERPINAN, M. Continuous latent variable models for dimensionality reduction and sequential data reconstruction. 01 2001.

CARSCOOPS. *This Is What Tesla's Autopilot Sees On The Road*. 2020. Acesso em: 27 de julho de 2022. Disponível em: <https://www.youtube.com/watch?v=fKXztwtXaGo>.

CARVALHO, V. d. S.; GUEDES, E. B.; SALAME, M. F. A. Classification of weeds in agricultural crops with ensembles of convolutional neural networks. In: IN: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL, 16, 2019 . *Embrapa Amazônia Ocidental-Artigo em anais de congresso (ALICE)*. [S.l.], 2020.

CASTEELS, W.; HELLINCKX, P. Exploiting non-i.i.d. data towards more robust machine learning algorithms. *CoRR*, abs/2010.03429, 2020. Disponível em: <https://arxiv.org/abs/2010.03429>.

CEPEA. *Mensuração econômica da incidência de pragas e doenças no Brasil: uma aplicação para as culturas de soja, milho e algodão.* Available at <https://www.cepea.esalq.usp.br/upload/kceditor/files/Cepea_EstudoPragaseDoencas_Parte%201.pdf>, Acessado em: 22/06/2022, 2019.

CHAPELLE, O.; SCHöLKOPF, B.; ZIEN, A. A discussion of semi-supervised learning and transduction. In: _____. *Semi-Supervised Learning*. [S.l.: s.n.], 2006. p. 473–478.

CHAUDHRY, A.; DOKANIA, P. K.; AJANTHAN, T.; TORR, P. H. S. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *CoRR*, abs/1801.10112, 2018. Disponível em: <http://arxiv.org/abs/1801.10112>.

CHAUHAN, R.; GHANSHALA, K. K.; JOSHI, R. Convolutional neural network (cnn) for image detection and recognition. In: *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. [S.l.: s.n.], 2018. p. 278–282.

COHN, D.; LADNER, R.; WAIBEL, A. Improving generalization with active learning. In: *Machine Learning*. [S.l.: s.n.], 1994. p. 201–221.

COLETTA, L.; HRUSCHKA, E.; ACHARYA, A.; GHOSH, J. Using metaheuristics to optimize the combination of classifier and cluster ensembles. *Integrated Computer Aided Engineering*, v. 22, p. 229–242, 06 2015.

COLETTA, L. F.; de Almeida, D. C.; SOUZA, J. R.; MANZIONE, R. L. Novelty detection in uav images to identify emerging threats in eucalyptus crops. *Computers and Electronics in Agriculture*, v. 196, p. 106901, 2022. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169922002186>.

COLETTA, L. F.; PONTI, M.; HRUSCHKA, E. R.; ACHARYA, A.; GHOSH, J. Combining clustering and active learning for the detection and learning of new image classes. *Neurocomputing*, v. 358, p. 150–165, 2019. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231219306605>.

CRUPI, V.; GUGLIELMINO, E.; MILAZZO, G. Neural-network-based system for novel fault detection in rotating machinery. *Journal of Vibration and Control - J VIB CONTROL*, v. 10, p. 1137–1150, 08 2004.

DAGAN, I.; ENGELSON, S. P. Committee-based sampling for training probabilistic classifiers. In: PRIEDITIS, A.; RUSSELL, S. (Ed.). *Machine Learning Proceedings 1995*. San Francisco (CA): Morgan Kaufmann, 1995. p. 150–157. ISBN 978-1-55860-377-6. Disponível em: <https://www.sciencedirect.com/science/article/pii/B978155860377650027X>.

DIEHL, C.; CAUWENBERGHS, G. Svm incremental learning, adaptation and optimization. In: *Proceedings of the International Joint Conference on Neural Networks, 2003.* [S.l.: s.n.], 2003. v. 4, p. 2685–2690 vol.4.

DIEHL, C.; HAMPSHIRE, J. Real-time object classification and novelty detection for collaborative video surveillance. In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290).* [S.l.: s.n.], 2002. v. 3, p. 2620–2625 vol.3.

DOERSCH, C. *Tutorial on Variational Autoencoders*. arXiv, 2016. Disponível em: <https://arxiv.org/abs/1606.05908>.

EMBRAPA. *Controle biológico de pragas da agricultura*. Available at <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/212490/1/CBdocument.pdf>, Acessado em: 22/06/2022, 2020.

EMBRAPA. *Plantas daninhas*. Available at <https://www.embrapa.br/tema-plantas-daninhas/sobre-o-tema>, Acessado em: 22/06/2022, 2022.

EVERITT, B. S. *An Introduction to Latent Variables Models*. [S.l.]: Chapman Hall, 1984.

FONTANEL, D.; CERMELLI, F.; MANCINI, M.; BULÒ, S. R.; RICCI, E.; CAPUTO, B. Boosting deep open world recognition by clustering. *CoRR*, abs/2004.13849, 2020. Disponível em: <https://arxiv.org/abs/2004.13849>.

FONTANEL, D.; CERMELLI, F.; MANCINI, M.; CAPUTO, B. On the challenges of open world recognition under shifting visual domains. *IEEE Robotics and Automation Letters*, v. 6, n. 2, p. 604–611, 2021.

FUKUNAGA, K. *Introduction to statistical pattern recognition*. [S.l.]: Elsevier, 2013.

GARETH, J.; DANIELA, W.; TREVOR, H.; ROBERT, T. *An introduction to statistical learning: with applications in R*. [S.l.]: Spinger, 2013.

GARG, A.; MAGO, V. Role of machine learning in medical research: A survey. *Computer Science Review*, v. 40, p. 100370, 2021. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013721000101>.

GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural networks and the bias/variance dilemma. *Neural Computation*, v. 4, n. 1, p. 1–58, 1992.

GENG, X.; SMITH-MILES, K. Incremental learning. In: _____. *Encyclopedia of Biometrics*. Boston, MA: Springer US, 2009. p. 731–735. ISBN 978-0-387-73003-5. Disponível em: <https://doi.org/10.1007/978-0-387-73003-5_304>.

GEORGE, L.-C. C. Y. Z.; SCHROFF, P. F.; ADAM, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. *ECCV. Liang-Chieh Chen Yukun Zhu George Papandreou Florian Schroff and Hartwig Adam*, 2018.

GEPPERTH, A.; HAMMER, B. Incremental learning algorithms and applications. In: *European Symposium on Artificial Neural Networks (ESANN)*. Bruges, Belgium: [s.n.], 2016. Disponível em: <https://hal.archives-ouvertes.fr/hal-01418129>.

GÉRON, A.; SAFARI, a. O. M. C. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Incorporated, 2019. Disponível em: <https://books.google.com.br/books?id=O2VJzQEACAAJ>.

GIDAHATAR. *Conteo geoespacial de cultivos a partir de ortofotos de drones con Python, Scikit Learn y Scikit Image - Tutorial*. 2020. Acesso em: 27 de julho de 2022. Disponível em: <https://gidahatari.com/ih-es/conteo-geoespacial-de-cultivos-a-partir-de-ortofotos-de-drones/-con-python-scikit-learn-y-scikit-image-tutorial>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GUERRIERO, S.; CAPUTO, B.; MENSINK, T. *DeepNCM: Deep Nearest Class Mean Classifiers*. 2018. Disponível em: <https://openreview.net/forum?id=rkPLZ4JPM>.

GUTSTEIN, S.; STUMP, E. Reduction of catastrophic forgetting with transfer learning and ternary output codes. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2015. p. 1–8.

HAN, J.; KAMBER, M. *Data Mining. Concepts and Techniques*. 2. ed. [S.l.]: Morgan Kaufmann, 2006. ISBN 1558609016.

HE, H.; CHEN, S.; LI, K.; XU, X. Incremental learning from stream data. *IEEE Transactions on Neural Networks*, v. 22, n. 12, p. 1901–1914, 2011.

HE, Z. Deep learning in image classification: A survey report. In: *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. [S.l.: s.n.], 2020. p. 174–177.

HINTON, G.; SALAKHUTDINOV, R. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, v. 313, p. 504–7, 08 2006.

HO, Y.-C.; AGRAWALA, A. On pattern classification algorithms introduction and survey. *Proceedings of the IEEE*, v. 56, n. 12, p. 2101–2114, 1968.

HOSSAIN, M. S.; SAHA, P.; CHOWDHURY, T. F.; RAHMAN, S.; RAHMAN, F.; MOHAMMED, N. *Rethinking Task-Incremental Learning Baselines*. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2205.11367>.

HOU, X.; SHEN, L.; SUN, K.; QIU, G. Deep feature consistent variational autoencoder. In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2017. p. 1133–1141.

HU, H.; LIAO, M.; ZHANG, C.; JING, Y. Text classification based recurrent neural network. In: *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. [S.l.: s.n.], 2020. p. 652–655.

HUANG, G. B.; RAMESH, M.; BERG, T.; LEARNED-MILLER, E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. [S.l.], 2007.

INáCIO, M.; IZBICKI, R.; GYIRES-TóTH, B. Distance assessment and analysis of high-dimensional samples using variational autoencoders. *Information Sciences*, v. 557, p. 407–420, 2021. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025520306538>.

ISLAM, K. T.; MUJTABA, G.; RAJ, R. G.; NWEKE, H. F. Handwritten digits recognition with artificial neural network. In: *2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)*. [S.l.: s.n.], 2017. p. 1–4.

JOSEPH, K. J.; KHAN, S.; KHAN, F. S.; BALASUBRAMANIAN, V. N. Towards open world object detection. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2021. p. 5826–5836.

JÚNIOR, P. C. P.; MONTEIRO, A.; RIBEIRO, R. da L.; SOBIERANSKI, A. C.; WANGENHEIM, A. von. Comparison of classical computer vision vs. convolutional neural networks for weed mapping in aerial images. *Revista de Informática Teórica e Aplicada*, v. 27, n. 4, p. 20–33, 2020.

JÚNIOR, P. R. M.; BOULT, T. E.; WAINER, J.; ROCHA, A. Open-set support vector machines. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, IEEE, 2021.

KIM, J.; KO, Y.; SEO, J. Construction of machine-labeled data for improving named entity recognition by transfer learning. *IEEE Access*, v. 8, p. 59684–59693, 2020.

KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

KIRILLOV, A.; MINTUN, E.; RAVI, N.; MAO, H.; ROLLAND, C.; GUSTAFSON, L.; XIAO, T.; WHITEHEAD, S.; BERG, A. C.; LO, W.-Y.; DOLLáR, P.; GIRSHICK, R. *Segment Anything*. 2023.

KIRKPATRICK, J.; PASCANU, R.; RABINOWITZ, N. C.; VENESS, J.; DESJARDINS, G.; RUSU, A. A.; MILAN, K.; QUAN, J.; RAMALHO, T.; GRABSKA-BARWINSKA, A.; HASSABIS, D.; CLOPATH, C.; KUMARAN, D.; HADSELL, R. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. Disponível em: <http://arxiv.org/abs/1612.00796>.

KODIROV, E.; XIANG, T.; GONG, S. Semantic autoencoder for zero-shot learning. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 4447–4456.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: . [S.l.]: Morgan Kaufmann, 1995. p. 1137–1143.

KORTGE, C. A. Episodic memory in connectionist networks. In: DEMPSTER, F. N.; BRAINERD, C. J.; BRAINERD, C. J. (Ed.). *The Twelfth Annual Conference of the Cognitive Science Society*. [S.l.]: Academic Press, 1990. p. 764–771.

KUBAT, M. *An Introduction to Machine Learning*. Springer International Publishing, 2015. ISBN 9783319200095. Disponível em: <https://books.google.com.br/books?id=5dWCrgEACAAJ>.

KULKARNI, P.; ADE, R. Incremental learning from unbalanced data with concept class, concept drift and missing features : A review. *International Journal of Data Mining Knowledge Management Process*, v. 4, p. 15–29, 11 2014.

KUNCHEVA, L. Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. *Proc. Eur. Conf. Artif. Intell.*, 01 2008.

LANGE, M. D.; ALJUNDI, R.; MASANA, M.; PARISOT, S.; JIA, X.; LEONARDIS, A.; SLABAUGH, G. G.; TUYTELAARS, T. Continual learning: A comparative study on how to defy forgetting in classification tasks. *CoRR*, abs/1909.08383, 2019. Disponível em: <http://arxiv.org/abs/1909.08383>.

LECUN, Y.; CORTES, C.; BURGES, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, v. 2, 2010.

LEE, J.; YOON, J.; YANG, E.; HWANG, S. J. Lifelong learning with dynamically expandable networks. *CoRR*, abs/1708.01547, 2017. Disponível em: <http://arxiv.org/abs/1708.01547>.

LEWANDOWSKY, S.; LI, S.-C. 10 - catastrophic interference in neural networks: Causes, solutions, and data. In: DEMPSTER, F. N.; BRAINERD, C. J.; BRAINERD, C. J. (Ed.). *Interference and Inhibition in Cognition*. San Diego: Academic Press, 1995. p. 329–361. ISBN 978-0-12-208930-5. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780122089305500118>.

LEWIS, D. D.; GALE, W. A. *A Sequential Algorithm for Training Text Classifiers*. arXiv, 1994. Disponível em: <https://arxiv.org/abs/cmp-lg/9407020>.

LOPEZ-PAZ, D.; RANZATO, M. Gradient episodic memory for continuum learning. *CoRR*, abs/1706.08840, 2017. Disponível em: <http://arxiv.org/abs/1706.08840>.

MA, K.; BEN-ARIE, J. Compound exemplar based object detection by incremental random forest. In: *2014 22nd International Conference on Pattern Recognition*. [S.l.: s.n.], 2014. p. 2407–2412.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. 11, 2008.

MAHMUD, M.; HUANG, J.; FU, X. Variational autoencoder-based dimensionality reduction for high-dimensional small-sample data classification. *International Journal of Computational Intelligence and Applications*, v. 19, p. 2050002, 04 2020.

MANCINI, M.; KARAOGUZ, H.; RICCI, E.; JENSFELT, P.; CAPUTO, B. Knowledge is never enough: Towards web aided deep open world recognition. *CoRR*, abs/1906.01258, 2019. Disponível em: <http://arxiv.org/abs/1906.01258>.

MASANA, M.; LIU, X.; TWARDOWSKI, B.; MENTA, M.; BAGDANOV, A. D.; WEIJER, J. van de. Class-incremental learning: survey and performance evaluation. *CoRR*, abs/2010.15277, 2020. Disponível em: <https://arxiv.org/abs/2010.15277>.

MASUD, M.; GAO, J.; KHAN, L.; HAN, J.; THURAISINGHAM, B. Classification and novel class detection in data streams with active mining. In: . [S.l.: s.n.], 2010. p. 311–324. ISBN 978-3-642-13671-9.

MASUD, M.; GAO, J.; KHAN, L.; HAN, J.; THURAISINGHAM, B. M. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering*, v. 23, n. 6, p. 859–874, 2011.

MCCALLUM, A.; NIGAM, K. Employing em and pool-based active learning for text classification. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. (ICML '98), p. 350–358. ISBN 1558605568.

MCCLOSKEY, M.; COHEN, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In: BOWER, G. H. (Ed.). Academic Press, 1989, (Psychology of Learning and Motivation, v. 24). p. 109–165. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0079742108605368>.

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

MENSINK, T.; VERBEEK, J.; CSURKA, G.; PERRONNIN, F. *Metric learning for nearest class mean classifiers*. [S.l.]: Google Patents, 2015. US Patent 9,031,331.

MENSINK, T.; VERBEEK, J.; PERRONNIN, F.; CSURKA, G. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 11, p. 2624–2637, 2013.

MERENTITIS, A.; DEBES, C.; HEREMANS, R. Ensemble learning in hyperspectral image classification: Toward selecting a favorable bias-variance tradeoff. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, v. 7, n. 4, p. 1089–1102, 2014.

MESHRAM, V.; PATIL, K.; MESHRAM, V.; HANCHATE, D.; RAMKTEKE, S. Machine learning in agriculture domain: A state-of-art survey. *Artificial Intelligence in the Life Sciences*, v. 1, p. 100010, 2021. ISSN 2667-3185. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667318521000106>.

MICHELUCCI, U. *Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks*. Apress, 2018. ISBN 9781484237908. Disponível em: <https://books.google.com.br/books?id=z1ptDwAAQBAJ>.

MILJKOVIć, D. Review of novelty detection methods. In: . [S.l.: s.n.], 2010. p. 593–598. ISBN 978-1-4244-7763-0.

MILLER, D.; SÜNDERHAUF, N.; MILFORD, M.; DAYOUB, F. Class anchor clustering: a distance-based loss for training open set classifiers. *CoRR*, abs/2004.02434, 2020. Disponível em: <https://arxiv.org/abs/2004.02434>.

MINSKY, M. Steps toward artificial intelligence. *Proceedings of the IRE*, v. 49, n. 1, p. 8–30, 1961.

MIRAFTABZADEH, S. M.; FOIADELLI, F.; LONGO, M.; PASETTI, M. A survey of machine learning applications for power system analytics. In: . [S.l.: s.n.], 2019.

MITCHELL, T. M. et al. *Machine learning. 1997*. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997.

MO, Y.; WU, Y.; YANG, X.; LIU, F.; LIAO, Y. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, v. 493, p. 626–646, 2022. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231222000054>.

MONTELEONI, C. Learning with online constraints: Shifting concepts and active learning. 08 2007.

OZA, N. C.; TUMER, K. Classifier ensembles: Select real-world applications. *Information Fusion*, v. 9, n. 1, p. 4–20, 2008. ISSN 1566-2535. Special Issue on Applications of Ensemble Methods. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1566253507000620>.

PANTAZI, X.-E.; MOSHOU, D.; BRAVO, C. Active learning system for weed species recognition based on hyperspectral sensing. *Biosystems Engineering*, v. 146, p. 193–202, 2016. ISSN 1537-5110. Special Issue: Advances in Robotic Agriculture for Crops. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1537511016000143>.

PARISI, G. I.; LOMONACO, V. Online continual learning on sequences. *CoRR*, abs/2003.09114, 2020. Disponível em: <https://arxiv.org/abs/2003.09114>.

PARK, C.; HUANG, J.; DING, Y. A computable plug-in estimator of minimum volume sets for novelty detection. *Operations Research*, v. 58, p. 1469–1480, 10 2010.

RAKHSITH, L.; S, A. K.; E, K. B.; D, A. N.; V, K. K. A survey on object detection methods in deep learning. In: *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*. [S.l.: s.n.], 2021. p. 1619–1626.

REBUFFI, S.; KOLESNIKOV, A.; LAMPERT, C. H. icarl: Incremental classifier and representation learning. *CoRR*, abs/1611.07725, 2016. Disponível em: <http://arxiv.org/abs/1611.07725>.

RISTIN, M.; GUILLAUMIN, M.; GALL, J.; GOOL, L. V. Incremental learning of ncm forests for large-scale image classification. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 3654–3661.

ROBINS, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, Taylor Francis, v. 7, n. 2, p. 123–146, 1995. Disponível em: <https://doi.org/10.1080/09540099550039318>.

ROH, Y.; HEO, G.; WHANG, S. E. A survey on data collection for machine learning: A big data - ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, v. 33, n. 4, p. 1328–1347, 2021.

RONY, J.; HAFEMANN, L. G.; OLIVEIRA, L. S.; AYED, I. B.; SABOURIN, R.; GRANGER, E. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. p. 4317–4325.

ROSA, R. D.; MENSINK, T.; CAPUTO, B. Online open world recognition. *CoRR*, abs/1604.02275, 2016. Disponível em: <http://arxiv.org/abs/1604.02275>.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, p. 53–65, 1987. ISSN 0377-0427. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377042787901257>.

RUFF, L.; VANDERMEULEN, R.; GOERNITZ, N.; DEECKE, L.; SIDDIQUI, S. A.; BINDER, A.; MÜLLER, E.; KLOFT, M. Deep one-class classification. In: DY, J.; KRAUSE, A. (Ed.). *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018. (Proceedings of Machine Learning Research, v. 80), p. 4393–4402. Disponível em: <https://proceedings.mlr.press/v80/ruff18a.html>.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. In: _____. *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988. p. 696–699. ISBN 0262010976.

RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 2002.

RUSU, A. A.; RABINOWITZ, N. C.; DESJARDINS, G.; SOYER, H.; KIRKPATRICK, J.; KAVUKCUOGLU, K.; PASCANU, R.; HADSELL, R. Progressive neural networks. *CoRR*, abs/1606.04671, 2016. Disponível em: <http://arxiv.org/abs/1606.04671>.

SALEHI, M.; ARYA, A.; PAJOUM, B.; OTOOFI, M.; SHAEIRI, A.; ROHBAN, M. H.; RABIEE, H. R. ARAE: adversarially robust training of autoencoders improves novelty detection. *CoRR*, abs/2003.05669, 2020. Disponível em: <https://arxiv.org/abs/2003.05669>.

SALEHI, M.; MIRZAEI, H.; HENDRYCKS, D.; LI, Y.; ROHBAN, M. H.; SABOKROU, M. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *CoRR*, abs/2110.14051, 2021. Disponível em: <https://arxiv.org/abs/2110.14051>.

SCHEIRER, W.; JAIN, L.; BOULT, T. Probability models for open set recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 36, n. 11, p. 2317–2324, Nov 2014. ISSN 0162-8828.

SCHEIRER, W. J.; ROCHA, A. de R.; SAPKOTA, A.; BOULT, T. E. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 7, p. 1757–1772, 2013.

SCHUMANN, R.; REHBEIN, I. Active learning via membership query synthesis for semi-supervised sentence classification. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 472–481. Disponível em: <https://aclanthology.org/K19-1044>.

SCHWARTZ, R.; DODGE, J.; SMITH, N. A.; ETZIONI, O. Green ai. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 63, n. 12, p. 54–63, nov 2020. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/3381831>.

SCIKIT-LEARN. Metrics and scoring: quantifying the quality of predictions. 2011. Disponível em: <https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics>.

SERRA, L. S.; MENDES, M. R. F.; SOARES, M.; MONTEIRO, I. P. Revolução verde: reflexões acerca da questão dos agrotóxicos. *Revista Científica do Centro de Estudos em Desenvolvimento Sustentável da UNDB*, v. 1, n. 4, p. 2–25, 2016.

SETTLES, B. Active Learning Literature Survey. In: *Relatório Técnico N. 1648*. University of Wisconsin–Madison: [s.n.], 2009.

SETTLES, B.; CRAVEN, M. An analysis of active learning strategies for sequence labeling tasks. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. USA: Association for Computational Linguistics, 2008. (EMNLP '08), p. 1070–1079.

SETTLES, B.; CRAVEN, M.; RAY, S. Multiple-instance active learning. In: PLATT, J.; KOLLER, D.; SINGER, Y.; ROWEIS, S. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2007. v. 20. Disponível em: <https://proceedings.neurips.cc/paper/2007/file/a1519de5b5d44b31a01de013b9b51a80-Paper.pdf>.

SEUNG, H. S.; OPPER, M.; SOMPOLINSKY, H. *Query by Committee*. 1992.

SHAHEEN, K.; HANIF, M. A.; HASAN, O.; SHAFIQUE, M. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *CoRR*, abs/2105.12374, 2021. Disponível em: <https://arxiv.org/abs/2105.12374>.

SHALEV-SHWARTZ, S. Online learning and online convex optimization. Now Publishers Inc., Hanover, MA, USA, v. 4, n. 2, p. 107–194, feb 2012. ISSN 1935-8237. Disponível em: <https://doi.org/10.1561/2200000018>.

SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, n. 3, p. 379–423, 1948.

SHARP, M.; AK, R.; HEDBERG, T. A survey of the advancing use and development of machine learning in smart manufacturing. *Journal of Manufacturing Systems*, v. 48, 03 2018.

SHEIKH, R.; MILIOTO, A.; LOTTES, P.; STACHNISS, C.; BENNEWITZ, M.; SCHULTZ, T. Gradient and log-based active learning for semantic segmentation of crop and weed for agricultural robots. In: . [S.l.: s.n.], 2020.

SHI, Y.; WANG, K.; LI, J.; LI, Y. Hyperspectral target detection with hierarchical denoising autoencoder and subspace projection. In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. [S.l.: s.n.], 2021. p. 4404–4407.

SHIN, H.; LEE, J. K.; KIM, J.; KIM, J. Continual learning with deep generative replay. *CoRR*, abs/1705.08690, 2017. Disponível em: <http://arxiv.org/abs/1705.08690>.

SOUZA, J. R.; MENDES, C. C. T.; GUIZILINI, V.; VIVALDINI, K. C. T.; COLTURATO, A.; RAMOS, F.; WOLF, D. F. Automatic detection of ceratocystis wilt in eucalyptus crops from aerial images. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. [S.l.: s.n.], 2015. p. 3443–3448.

SPIRTES, P. Latent structure and causal variables. In: WRIGHT, J. D. (Ed.). *International Encyclopedia of the Social Behavioral Sciences (Second Edition)*. Second edition. Oxford: Elsevier, 2015. p. 394–397. ISBN 978-0-08-097087-5. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780080970868421367>.

SUN, X.; YANG, Z.; ZHANG, C.; PENG, G.; LING, K. V. Conditional gaussian distribution learning for open set recognition. *CoRR*, abs/2003.08823, 2020. Disponível em: <https://arxiv.org/abs/2003.08823>.

TACK, J.; MO, S.; JEONG, J.; SHIN, J. CSI: novelty detection via contrastive learning on distributionally shifted instances. *CoRR*, abs/2007.08176, 2020. Disponível em: <https://arxiv.org/abs/2007.08176>.

TERVEN, J.; CORDOVA-ESPARZA, D. *A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond*. 2023.

UNGARBAYEV, D.; DEMIREL, O.; AKHTAR, M. T. Automatic data augmentation method with improved interpretability for image classification in computer vision applications. In: *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. [S.l.: s.n.], 2022. p. 1356–1361.

VEN, G. M. van de; TOLIAS, A. S. Three scenarios for continual learning. *CoRR*, abs/1904.07734, 2019. Disponível em: <http://arxiv.org/abs/1904.07734>.

VIGNOTTO, E.; ENGELKE, S. Extreme value theory for open set classification - gpd and gev classifiers. *ArXiv*, abs/1808.09902, 2018.

VINCENT, P.; LAROCHELLE, H.; LAJOIE, I.; BENGIO, Y.; MANZAGOL, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, v. 11, n. 110, p. 3371–3408, 2010. Disponível em: <http://jmlr.org/papers/v11/vincent10a.html>.

WANG, C.; YEH, I.; LIAO, H. M. You only learn one representation: Unified network for multiple tasks. *CoRR*, abs/2105.04206, 2021. Disponível em: <https://arxiv.org/abs/2105.04206>.

WANG, K. Variational autoencoder based approach for imbalance process fault detection. In: *2021 International Conference on Electronics, Circuits and Information Engineering (ECIE)*. [S.l.: s.n.], 2021. p. 149–152.

WELLING, M. Herding dynamical weights to learn. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2009. (ICML '09), p. 1121–1128. ISBN 9781605585161. Disponível em: <https://doi.org/10.1145/1553374.1553517>.

YANG, J.; ZHOU, K.; LI, Y.; LIU, Z. Generalized out-of-distribution detection: A survey. *CoRR*, abs/2110.11334, 2021. Disponível em: <https://arxiv.org/abs/2110.11334>.

YANG, X.; PENG, K.; AN, L.; HUANG, P.; FENG, P. Gmblof: A machine learning algorithm of novelty detection based on local outlier factor. In: *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. [S.l.: s.n.], 2022. p. 20–25.

YEUNG, K. Y. *Cluster analysis of gene expression data*. [S.l.]: University of Washington, 2001.

YU, J.; WANG, Z.; VASUDEVAN, V.; YEUNG, L.; SEYEDHOSSEINI, M.; WU, Y. *CoCa: Contrastive Captioners are Image-Text Foundation Models*. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2205.01917>.

ZAHEER, M. Z.; LEE, J. H.; ASTRID, M.; LEE, S. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. *CoRR*, abs/2004.07657, 2020. Disponível em: <https://arxiv.org/abs/2004.07657>.

ZHU, X.; GOLDBERG, A. B. *Introduction to Semi-Supervised Learning*. [S.l.]: Morgan & Claypool, 2009. (Synthesis Lectures on Artificial Intelligence and Machine Learning).

ZIMMERER, D.; KOHL, S. A. A.; PETERSEN, J.; ISENSEE, F.; MAIER-HEIN, K. H. *Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection*. arXiv, 2018. Disponível em: <https://arxiv.org/abs/1812.05941>.