UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Marcondes Ricarte da Silva Júnior

Mapas Auto-organizáveis Multi-camadas Especialistas com Relevâncias Intra e Inter-categoria

Recife

2025

Marcondes Ricarte da Silva Júnior

Mapas Auto-organizáveis Multi-camadas Especialistas com Relevâncias Intra e Inter-categoria

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Doutorado em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Aluizio Fausto Ribeiro Araújo

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Silva Júnior, Marcondes Ricarte da.
Mapas Auto-organizáveis Multi-camadas Especialistas com Relevâncias Intra e Inter-categoria / Marcondes Ricarte da Silva Júnior. - Recife, 2024.
133f.: il.

Tese (Doutorado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-graduação em Ciência da Computação, 2024.
Orientação: Aluizio Fausto Ribeiro Araújo.
Inclui referências.

1. Mapas Auto-organizáveis; 2. Multi-camadas; 3. Relevâncias Intra e Inter-categoria; 4. Agrupamento por em subespaços; 5. Refinamento. I. Araújo, Aluizio Fausto Ribeiro. II. Título.

UFPE-Biblioteca Central

SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

A tese de doutorado intitulada **"Mapas Auto-organizáveis Multi-camadas Especialistas com Relevâncias Intra e Inter-categoria"**, apresentada por **Marcondes Ricarte da Silva Júnior** em 27/02/2024, foi aprovada pela Banca Examinadora. Como exigência regimental para a obtenção do grau de doutor, o aluno deverá entregar a versão final do trabalho com as devidas correções propostas pelos examinadores, no prazo de **90** dias.

Recife, 27 de fevereiro de 2024.

_____
Prof. Dr. Hansenclever de França Bassani
Centro de Informática / UFPE

_____
Prof. Dr. Adrien Joan Sylvain Durand-Petiteville
Departamento de Engenharia Mecânica / UFPE

_____
Prof. Dr. Zhao Liang
Departamento de Computação e Matemática / USP

_____
Prof. Dr. Guilherme de Alencar Barreto
Departamento de Engenharia de Teleinformática / UFC

_____
Prof. Dr. Bruno José Torres Fernandes
Escola Politécnica de Pernambuco / UPE

As correções solicitadas na defesa foram aprovadas pelo presidente da banca em:

____/___/____ _____
(Secretaria/Carimbo)

Dedico esta tese, com todo o meu carinho e gratidão, à minha família, que esteve ao meu lado em cada passo desta jornada.

AGRADECIMENTOS

Inicialmente, expresso minha gratidão a Deus por Seu constante cuidado ao longo da minha vida. Acredito que o amor Dele é infinito e incondicional, e sou profundamente grato por isso. Agradeço por todas as oportunidades e talentos que Ele me concedeu. Quero dedicar um especial agradecimento ao Prof. Aluizio Araújo por ser não apenas meu orientador, mas também meu companheiro, incentivador, mentor e mediador. Cada papel desempenhado por ele contribuiu para minha formação e o desenvolvimento deste trabalho. Agradeço por confiar em mim essa responsabilidade. À minha esposa, Marta Naiá, minha grande paixão, que expresso minha profunda gratidão pela compreensão, dedicação, carinho, paciência e companherismo. Cada página desta tese tem um pouco de você também.

Minhas palavras de gratidão não são suficientes para expressar o quanto sou grato aos meus pais, Marcondes e Raquel Ricarte, e à minha irmã Rebbeca Ricarte. Apoiar alguém nas crenças é fácil, mas apoiar algo que pode parecer uma loucura, demonstra respeito e confiança. Mãe, agradeço por confiar em mim mesmo quando a intuição talvez duvidasse. Pai, obrigado por sempre me acompanhar e incentivar em todas as minhas atividades. Rebbeca, apesar das brincadeiras ao longo da vida, você é parte fundamental do que sou hoje, e sempre serei grato por isso.

À Vovó Iolanda, minhas tias, meus tios, primos e primas, meu profundo agradecimento. A Elizabeth Carvalho, que esteve ao meu lado nos momentos mais desafiadores, agradeço por suas observações, intervenções, conselhos e sugestões, todos ouvidos com atenção e apreço.

Aos amigos do Laboratório, André Tiba, Orivaldo Vieira, Flávia Araújo, Paulo Ferreira, Elaine Guerreiro, Felipe Duque, Hesdras Viana, Lucas Farias e Mário Melo, agradeço pelas conversas, sugestões e pela vivência compartilhada ao longo deste trabalho.

A Alexandre Mota e Augusto Sampaio, que lideraram um projeto de robótica e inteligência artificial, agradeço por proporcionarem uma experiência enriquecedora, contribuindo para meu crescimento como líder e pesquisador. Obrigado também à equipe técnica do projeto, meus companheiros de um trabalho inovador: Riei Rodrigues, Cecília Virgínia, Jefferson Norberto, Jairton Falcão, Jonas Ferreira e Matheus Hopper.

"No tecido da visão, cada pixel é um fio que entrelaça a tapeçaria da compreensão, transformando luz em conhecimento."(Autor desconhecido).

RESUMO

O crescimento exponencial na quantidade e acessibilidade de dados provenientes de diversas fontes tem gerado uma demanda crescente por modelos e ferramentas capazes de capturar, armazenar, analisar e visualizar esses dados. A habilidade de categorizar cenas é crucial para a construção de mapas semânticos por robôs, proporcionando-lhes a capacidade de compreender o contexto do ambiente com conceitos de alto nível. A categorização de cenas enfrenta diversos desafios, incluindo a extração eficiente de características e a categorização de dados complexos. A complexidade do agrupamento aumenta proporcionalmente ao número de amostras, categorias e dimensionalidade dos dados, levando as métricas de distância a perderem poder de discriminação. Tarefas de categorização para mineração de dados ou categorização de objetos também são muito relevantes e demandadas no contexto da categorização. Nesse cenário, parte-se de um algoritmo de agrupamento por subespaço, denominado Mapas Auto-organizáveis para Agrupamento em Subespaço (SC-SOM), que aprende uma categoria por vez em um Mapa Auto-organizável (SOM) individual. A extensão desse modelo inicialmente proposto se chama Múltiplos Mapas Auto-organizáveis Especialistas em uma Categoria (MOCE-SOM), composto por vários módulos que possuem alguns SC-SOM dispostos serialmente, em camadas. O MOCE-SOM categoriza enquanto aprende um coeficiente de relevância intra-categoria para cada dimensão de agrupamento, enfatizando os atributos mais importantes para uma dada categoria. Ao final de cada treinamento do SC-SOM, cada camada incorpora também a relevância inter-categoria, diferenciando as dimensões mais importantes para distinção entre categorias. As amostras são propagadas entre as camadas por meio de um processo de composição de amostras com protótipos treinados, modificando assim a representação das amostras durante sua propagação entre as camadas. Ao final de todas as camadas, é realizado um refinamento por meio de inferência variacional. O MOCE-SOM foi avaliado utilizando conjuntos de dados para mineração, reconhecimento de objetos e cenas. Os resultados foram comparados em termos de acurácia e tempo com modelos baseados em protótipos, além de famílias de modelos derivados do k-NN e do LVQ. Uma análise detalhada de cada etapa relevante do MOCE-SOM foi realizada para entender sua contribuição na solução da categorização. Os experimentos indicam que o modelo é competitivo e escalável.

Palavras-chaves: Mapas Auto-organizáveis; Multi-camadas; Relevâncias Intra e Inter-categoria, Agrupamento por em subespaços; Refinamento.

ABSTRACT

The exponential growth in the quantity and accessibility of data from various sources has led to an increasing demand for models and tools capable of capturing, storing, analyzing, and visualizing this data. The ability to categorize scenes is crucial for the construction of semantic maps by robots, providing them with the capability to understand the context of the environment with high-level concepts. Scene categorization faces various challenges, including the efficient extraction of features and the categorization of complex data. The complexity of clustering increases proportionally with the number of samples, categories, and data dimensionality, causing distance metrics to lose discrimination power. Categorization tasks for data mining or object categorization are also highly relevant and demanded in the context of categorization. In this scenario, we start with a subspace clustering algorithm, named Subspace Clustering Self-Organizing Maps, which learns one category at a time on an individual Self-Organizing Map (SOM). The extension of this initially proposed model is called Multiple One-Category Expert Self-Organing Maps (MOCE-SOM), consisting of multiple modules with several SC-SOMs arranged serially in layers. The MOCE-SOM categorizes while learning an intra-category relevance coefficient for each clustering dimension, emphasizing the most important attributes of a given category. At the end of each SC-SOM training, each layer incorporates inter-category relevance, differentiating the most important dimensions for category distinction. Samples are propagated between layers by composing samples with trained prototypes, thus modifying the representation of samples during their propagation between layers. At the end of all layers, refinement is performed through variational inference. The MOCE-SOM was evaluated using datasets for data mining, object recognition, and scenes. Results were compared in terms of accuracy and time with prototype-based models, model families derived from k-NN and LVQ. A detailed analysis of each relevant step of the MOCE-SOM was conducted to understand its contribution to the categorization solution. The experiments indicate that the model is competitive and scalable.

Keywords: Self-Organizing Maps; Multi-Layers; Intra and Inter-Category Relevancies, Subspace Clustering; Refinement.

LISTA DE FIGURAS

Figura 1 – Ilustrações de grande variação intra-categórica e ambiguidade semântica. Topo (grande variação intra-categoria). Os shoppings são bem diferentes devido à iluminação e conteúdo geral. Abaixo (ambigüidade semântica). Fonte: (ZENG et al., 2021). 31

Figura 2 – Uma arquitetura Rede Neural Convolucional (CNN) simples, composta por apenas cinco camadas. (O'SHEA; NASH, 2015). 39

Figura 3 – Uma representação visual de uma camada convolucional. O elemento central do *kernel* é colocado sobre o vetor de entrada, do qual é então calculado e substituído com uma soma ponderada de si mesmo e quaisquer pixels próximos. (O'SHEA; NASH, 2015). 40

Figura 4 – Ilustração esquemática do LMNN. 56

Figura 5 – Taxonomia dos mais relevantes classificadores *Learning Vector Quantization* (LVQ) desde o trabalho inicial de Teuvo Kohonen nos anos 80. Fonte: (NOVA; ESTÉVEZ, 2014). 60

Figura 6 – Esquema da regra de aprendizado para o LVQ2.1. Os círculos representam os protótipos; o quadrado representa uma amostra de entrada; e as cores indicam rótulos de categorias diferentes. Neste exemplo, a amostra é categorizada incorretamente e o protótipo mais próximo da categoria correta se move em direção à amostra, enquanto o protótipo mais próximo da categotia incorreta se afasta da amostra (NOVA; ESTÉVEZ, 2014). 63

Figura 7 – Esquema da regra de aprendizado GLVQ que analisa o comportamento dos dois protótipos pertencentes a diferentes categorias. Os círculos representam os protótipos; o quadrado representa a amostra de entrada; e as cores indicam os diferentes rótulos de categorias (regra aprendizado quando uma amostra de entrada com um rótulo de categoria vermelha é apresentada). (b) Regra de aprendizado quando uma amostra de entrada com um rótulo de categoria azul é apresentada. (NOVA; ESTÉVEZ, 2014). 68

Figura 8 – (a) Grade Linear. (b) Grade Retangular. (c) Grade Hexagonal (ASTUDILLO; OOMMEN, 2014). 75

Figura 9 – Arquitetura do modelo proposto. 90

Figura 10 – Análise da influência do β na acurácia do conjunto de treinamento e teste. (a) Base de dados: Iris (b) Base de dados: Scenes_15. Fonte: Autor 94
Figura 11 – Imagens de exemplo do conjunto de dados CIFAR_10 (KRIZHEVSKY; HINTON et al., 2009). 102
Figura 12 – Imagens de exemplo do conjunto de dados Caltech_101 (FEI-FEI; FERGUS; PERONA, 2004). 103
Figura 13 – Imagens de exemplo do conjunto de dados Scenes_15 (LAZEBNIK; SCHMID; PONCE, 2006). 104
Figura 14 – Imagens de exemplo do conjunto de dados Indoors_67 (QUATTONI; TORRALBA, 2009). 105
Figura 15 – Imagens de exemplo do conjunto de dados Indoors_67 (XIAO et al., 2010). . 106
Figura 16 – Acurácia média para os conjuntos de mineração de dados. 113
Figura 17 – Tempo de treinamento em escala logarítmica para mineração de dados. . . 113
Figura 18 – Acurácia média para os conjuntos de categorização de objetos. 114
Figura 19 – Tempo de treinamento em escala logarítmica para categorização de objetos. 114
Figura 20 – Acurácia média para os conjuntos de categorização de cenas. 115
Figura 21 – Tempo de treinamento em escala logarítmica para categorização de cenas. . 116
Figura 22 – Análise de ablation: (a) Minaração de dados, (b) categorização de objetos e (c) categorização de cenas. 118

LISTA DE TABELAS

Tabela 1 – Tabela com as características dos principais métodos da família LVQ (NOVA; ESTÉVEZ, 2014). 61

Tabela 2 – Descrição dos conjuntos de dados: número de amostras, atributos, categorias e tipo de problema: Mineração de Dados - MD, Categorização de Objetos - CO e Categorização de Cenas - CC. 101

Tabela 3 – Tabela com os intervalos de valores de parametrização para o *Multiple One-Category Expert Self-Organing Maps* (MOCE-SOM). 108

Tabela 4 – Tabela com os resultados para todas as bases de dados comparando os métodos. São apresentados todas as médias e o desvios padrões de acurácia. Comparação ao final da tabela: $+/-/\approx$. 110

Tabela 5 – Tabela com os resultados para todas as bases de dados comparando os métodos. São apresentados melhor acurácia para cada modelo. Comparação ao final da tabela: $+/-/\approx$. 111

Tabela 6 – Tabela com os resultados para todas as bases de dados comparando os métodos. São apresentados todas as médias e desvios padrões dos tempos treinamento (em segundos) para os melhores resultados. Comparação ao final da tabela: $+/-/\approx$. 112

Tabela 7 – Tabela com os resultados para todas as bases de dados comparando a principais etapas do MOCE-SOM. São apresentados todas as médias e desvios padrões da acurácia em cada etapa principal do modelo e na linha final é comparado o ganho médio entre as configurações. 118

LISTA DE ABREVIATURAS E SIGLAS

k-**NN**	k-vizinhos mais próximos
AMP	*Approximate Message Passing*
BMU	*Best Matching Unit*
BoVW	*Bag-of-Visual Word*
CFA	*Contextual Features in Appearance*
CFV	*Convolution Fisher Vector*
CNN	Rede Neural Convolucional
CRF	*Conditional Random Field*
DL-CNN	*Dictionary Learning CNN*
DSSOM	*Dimension Selective Self-organizing Map*
GAP	*Global Average Pooling*
GAP-CNN	*Global Average Pooling Convolutional Neural Networks*
GLVQ	*Generalized Learning Vector Quantization*
GMLVQ	*Generalized Matrix Learning Vector Quantization*
GMM	*Gaussian Mixture Model*
GRLVQ	*Generalized Relevance Learning Vector Quantization*
HoG	*Histogram of Oriented Gradients*
HSC	*Hard Subspace Clustering*
ICANN	*International Conference on Artificial Neural Networks*
IFV	*Improved Fisher Vector*
JIRS	*Journal of Intelligent and Robotic Systems*
LANN	Locally Adaptive Nearest Neighbors
LARFDSSOM	*Local Adaptive Receptive Field Dimension Selective Self-organizing Map*
LARFDSSOM2	*Local Adaptive Receptive Field Dimension Selective Self-organizing Map 2*
LARS	*Latin American Robotics Symposium*

LCDA	*Louvain Community Detection Algorithm*
LGMLVQ	*Local Generalized Matrix Learning Vector Quantization*
LHS	Latin Hyper-cube Sampling
LMNN	*Large Margin Nearest Neighbor*
LSTM	*Long short-term memory*
LVQ	*Learning Vector Quantization*
MLR	*Mid-level Local Discriminative Representation*
MOCE-SOM	*Multiple One-Category Expert Self-Organing Maps*
MR-CNN	Multi Resolution Convolutional Neural Networks
OLARFDSSOM	*Online Local Adaptive Receptive Field Dimension Selective Self-Organizing Map*
OLVQ	*Optimized-learning-rate Learning Vector Quantization*
ResNets	Redes Residuais
RLVQ	*Relevance Learning Vector Quantization*
S2G-Stream	Soft Subspace Neural Gas for Data Stream Clustering
SC	Agrupamento de Subespaço
SC-MLSOM	*Subspace Clustering Multi-Layer Self-organizing Map*
SC-SOM	*Subspace Clustering Self-organizing Map*
SC-MuSOM	*Subspace Clustering Multi-module Self-organizing Map*
SIFT	*Scale-Invariant Feature Transform*
SLAM	*Simultaneous Localization and Mapping*
SOM	Mapa Auto-Organizável
SSC	*Soft Subspace Clustering*
STDA	Associação de Dados Espaço-Temporais
SURF	*Speeded up robust features*
SVM	*Support Vector Machine*
UCI	*Machine Learning Repository*

VLAD *Vector of Locally Aggregated Descriptors*

LISTA DE SÍMBOLOS

$\mathbf{x}$	Amostra.
$\mathbf{X}$	Amostras.
ν	Fator de penalização do LMNN.
$\mathbf{w}$	Peso do centro ou propótipo.
η_0	Taxa de aprendizado inicial.
η_f	Taxa de aprendizado final.
η	Taxa de aprendizado.
n	Época.
N	Quantidade de épocas de um treinamento.
ϵ	Fator de estabilização.
d^+	Distância para o protótipo correto mais próximo.
d^-	Distância para o protótipo incorreto mais próximo.
$\mathbf{w}^+$	Protótipo ou centro correto mais próximo.
$\mathbf{w}^-$	Protótipo ou centro incorreto mais próximo.
D	Dimensão das amostras.
$r_{j/norm}$	Vetor de relevância normalizado do bloco j.
$r_{j/insta}$	Vetor de relevância instantâneo do bloco j.
$r_{j/inter}$	Vetor de relevância inter-categoria.
d_r	Distância com relevância.
h_w	Vizinhança do peso w.
σ	Fator de decaimento da vizinhaça.
λ	Fator de ponderação da operação de composição.

ρ	Fator de ponderação das amostras com protótipos.
β	Fator de ponderação entre as relevâncias intra e inter.
N_{inf}	Número de épocas máxima da etapa de inferência bayesiana.
δ	Erro mínimo de parada para inferência bayesiana.
τ_λ	Taxa de decaimento do parâmetro λ.
τ_η	Taxa de decaimento do parâmetro η.

SUMÁRIO

1 INTRODUÇÃO 20

1.1 OBJETIVOS 26

1.2 HIPÓTESE DE PESQUISA 26

1.3 PUBLICAÇÕES 27

1.4 ORGANIZAÇÃO DO DOCUMENTO 28

2 CATEGORIZAÇÃO DE CENAS E LUGARES: FORMALIZAÇÃO DO PROBLEMA E EXTRAÇÃO DE CARACTERÍSTICAS 29

2.1 CATEGORIZAÇÃO DE CENAS: DEFINIÇÃO DO PROBLEMA 29

2.1.1 Mapeamento Semântico e Categorização de Lugares 31

2.1.2 Categorização de Lugares e Cenas 33

2.2 EXTRAÇÃO DE CARACTERÍSTICAS DE CENAS 35

2.3 MÉTODOS DE EXTRAÇÃO DE CARACTERÍSTICAS *HANDCRAFTED* 37

2.4 MÉTODOS BASEADOS EM APRENDIZAGEM PROFUNDA 38

2.4.1 Fundamentos da CNN 38

2.4.2 Arquitetura da CNN 40

2.4.2.1 Modelo CNN Pré-treinado 41

2.4.2.2 Modelo CNN Ajustado 42

2.4.2.3 Modelo Específico de CNN 43

2.4.3 Representação de Cena Baseada em CNN 44

2.4.3.1 Método Baseado em Características de CNN Global 44

2.4.3.2 Características Invariantes ao Espaço 45

2.4.3.3 Características Semânticas 46

2.4.3.4 Característiscas Multi-camadas 47

2.4.3.5 Características Combinadas 48

2.4.4 Visual Geometry Group - VGG_19 49

2.5 DISCUSSÃO 51

3 CATEGORIZAÇÃO POR MODELOS QUE UTILIZAM A DISTÂNCIA POR VETORES DE REFERÊNCIA 53

3.1 MÉTODOS BASEADOS EM k-NN 53

3.1.1 Large Margin Nearest Neighbor (LMNN) 55

3.1.2 Locally Adaptive Nearest Neighbors (LANN) . . . 57
3.2 MODELOS LVQ E SUAS VARIAÇÕES COM RELEVÂNCIAS . . . 59
3.2.1 Learning Vector Quantization 1 (LVQ1) . . . 60
3.2.2 Learning Vector Quantization 2.1 (LVQ2.1) . . . 64
3.2.3 Learning Vector Quantization 3 (LVQ3) . . . 65
3.2.4 Optimized Learning Rate Learning Vector Quantization (OLVQ) . . . 66
3.2.5 Generalized Learning Vector Quantization (GLVQ) . . . 66
3.2.6 Generalized Relevance Learning Vector Quantization (GRLVQ) . . . 69
3.2.7 Generalized Matrix Learning Vector Quantization (GMLVQ) e Local Generalized Matrix Learning Vector Quantization (LGMLVQ) . . . 71
3.3 MODELO SOM . . . 73
3.4 LVQ VERSUS SOM . . . 76
3.4.1 SOM para Alta Dimensionalidade (*Subspace Clustering*) . . . 77
3.4.2 SOM para Categorização de Cenas e Lugares . . . 79
3.4.3 Refinamento . . . 80
3.4.3.1 Inferência Variacional . . . 81
3.5 DISCUSSÃO . . . 84
4 MÚLTIPLOS MAPAS AUTO-ORGANIZÁVEIS ESPECIALISTAS EM UMA CATEGORIA (MOCE-SOM) . . . 85
4.1 MOTIVAÇÃO E CARACTERÍSTICAS DO MODELO . . . 86
4.2 MÚLTIPLOS MAPAS AUTO-ORGANIZÁVEIS ESPECIALISTAS EM UMA CATEGORIA - MOCESOM . . . 88
4.2.1 Papel do SC-SOM no MOCE-SOM . . . 89
4.2.1.1 Treinamento de cada SC-SOM . . . 91
4.2.1.2 Inserção da Influência entre Grupos nas Relevâncias . . . 93
4.2.2 Composição . . . 95
4.2.3 Refinamento através de Inferência Variacional . . . 96
5 EXPERIMENTOS . . . 100
5.1 BASE DE DADOS . . . 100
5.2 METODOLOGIA DOS EXPERIMENTOS . . . 106
5.3 RESULTADOS . . . 108
5.3.1 Mineração de Dados . . . 109

5.3.2 Categorização de Objetos 109
5.3.3 Categorização de Cenas 114
5.3.4 Análise por Camadas do MOCE-SOM 116
5.3.5 Discussão 117
6 CONCLUSÃO 120
6.1 LIMITAÇÕES DO MODELO 122
6.2 TRABALHOS FUTUROS 123
REFERÊNCIAS 124

1 INTRODUÇÃO

Nos primórdios da robótica, os algoritmos de visão computacional utilizados por robôs pioneiros eram notoriamente simples devido às limitações de processamento dos computadores de então. Visão Computacional refere-se ao campo da inteligência artificial que capacita máquinas a interpretar e processar imagens do mundo real, de maneira similar ao olho humano. Ao longo das últimas seis décadas, as principais áreas de conhecimento na robótica passaram por inovações tecnológicas significativas, impulsionadas pelo avanço nas áreas de computação, comunicações e dispositivos eletrônicos. Esses avanços resultaram no desenvolvimento de sensores inteligentes, atuadores e aprimoramentos no planejamento e tomada de decisão, elevando consideravelmente a capacidade e adaptabilidade dos robôs móveis (GUZEL, 2013).

A navegação autônoma de robôs móveis, uma área em constante evolução, almeja replicar a habilidade dos animais na percepção de ambientes, elaboração de mapas e rotas, permitindo que os robôs se movam de maneira independente e robusta no ambiente. A Visão Computacional, especialmente, tem desempenhado um papel crucial nesse contexto, contribuindo significativamente para o avanço da navegação autônoma de robôs móveis (BONIN-FONT; ORTIZ; OLIVER, 2008). O campo da navegação robótica baseada em visão tem recebido contribuições substanciais da comunidade científica desde a década de 90

No contexto da robótica móvel, um dos maiores desafios é a criação de robôs capazes de navegar de forma autônoma em ambientes complexos e dinâmicos. Para isso, a técnica de *Simultaneous Localization and Mapping* (SLAM) se tornou uma ferramenta essencial. SLAM permite que os robôs construam um mapa do ambiente ao mesmo tempo em que localizam sua posição dentro desse mapa. Esse processo é fundamental para a navegação autônoma, pois fornece ao robô uma compreensão contínua e atualizada do ambiente ao seu redor (KOSTAVELIS; GASTERATOS, 2015).

O mapeamento semântico, uma extensão do SLAM, vai além do simples mapeamento físico do ambiente. Ele busca atribuir significado aos diferentes elementos presentes no mapa, permitindo que o robô não apenas saiba onde estão os obstáculos, mas também entenda o que são esses obstáculos. Por exemplo, um robô pode identificar e classificar diferentes tipos de objetos como mesas, cadeiras ou pessoas, e utilizar essa informação para tomar decisões mais inteligentes e contextualizadas.

Os avanços em robótica móvel e SLAM têm sido acelerados pelo desenvolvimento de sen-

sores mais sofisticados, como LiDARs, câmeras RGB-D e sensores ultrassônicos, que fornecem dados de alta resolução e em tempo real. Esses sensores são complementados por algoritmos avançados de processamento de dados e aprendizado de máquina, que permitem ao robô interpretar e reagir a mudanças no ambiente de forma mais eficaz

Além disso, o mapeamento semântico tem aplicações práticas significativas, especialmente em ambientes complexos como hospitais, fábricas e cidades inteligentes. Nessas situações, a capacidade de um robô entender e interagir com o ambiente de maneira inteligente e adaptativa é crucial para a realização de tarefas complexas de maneira eficiente e segura.

A integração dessas tecnologias de visão computacional, SLAM e mapeamento semântico representa um avanço significativo na robótica, permitindo que os robôs móveis operem de forma mais autônoma, segura e eficiente em uma variedade de ambientes. Esses desenvolvimentos não apenas expandem as capacidades dos robôs, mas também abrem novas possibilidades para suas aplicações em diversos setores da sociedade.

A extração de características de imagens é um processo fundamental em visão computacional e análise de imagens. Consiste na identificação e descrição de padrões relevantes em uma imagem, como bordas, texturas e formas. Métodos populares incluem o uso de filtros de convolução, descritores de textura e redes neurais convolucionais (CNNs). A extração de características de imagens desempenha um papel essencial na navegação semântica, pois viabiliza a criação de mapas semânticos e topológicos em diversas atividades autônomas de robôs. Mapas semânticos representam características de alto nível em um espaço de características, enquanto mapas topológicos organizam dados de forma topológica, preservando relações espaciais. Ambos são utilizados em robótica e visão computacional para facilitar a compreensão e a navegação em ambientes complexos. Mais especificamente, planejamento de rotas, desvio de obstáculos, exploração autônoma e manipulação de objetos são algumas das atividades facilitadas por essa abordagem, que contribui significativamente para o reconhecimento de espaços, categorização de ambientes e identificação de contextos específicos (PRONOBIS, 2011). A capacidade de um robô tomar decisões autônomas baseadas na compreensão semântica do ambiente é aprimorada pela extração precisa e discriminativa de características das imagens, permitindo uma representação rica e contextualizada do espaço circundante.

Nesse contexto, a construção de mapas semânticos é essencial para representar o ambiente em alto nível de abstração, proporcionando uma compreensão mais rica semanticamente do espaço. A categorização de lugares desempenha um papel crucial nesse processo, permitindo a associação de propriedades semânticas a diferentes espaços e identificando cômodos espe-

cíficos (PRONOBIS, 2011). A categorização de lugares é um processo que atribui rótulos ou categorias a regiões ou pontos em um ambiente. Essas categorias podem variar desde simples descrições, como cozinha ou quarto, até conceitos mais complexos, como locais de interesse em uma cidade. Ao categorizar lugares, os mapas semânticos tornam-se ferramentas valiosas para a tomada de decisões autônomas, permitindo que os robôs identifiquem e compreendam o ambiente ao seu redor. O Capítulo 2 discute em detalhes a categorização de lugares e cenas, oferecendo uma visão aprofundada das abordagens existentes na literatura e destacando a relevância dessa etapa para a construção de mapas semânticos significativos na navegação semântica.

O processo de extração de características de imagens apresenta desafios significativos, tanto em abordagens *handcrafted*, que são características manuais, quanto em modelos baseados em aprendizagem profunda. A abordagem de extração de características manual, ou *handcrafted*, envolve a concepção de características específicas por especialistas humanos, visando superar desafios como oclusões e variações de escala e iluminação. Essas características são projetadas para encontrar um equilíbrio entre precisão e eficiência computacional, sendo robustas em situações específicas. Abordagens *handcrafted*, como algoritmos tradicionais de processamento de imagem, muitas vezes enfrentam dificuldades na representação abstrata e hierárquica de características complexas presentes em imagens de alta resolução.

Por outro lado, a abordagem de extração de características por meio de CNNs pré-treinadas, ou *non-handcrafted*, envolve a extração de características diretamente dos dados de entrada, aprendendo com as observações. Isso permite a descoberta de múltiplos níveis de representação, proporcionando maior robustez à variabilidade intra-categoria. No entanto, as características extraídas pelas camadas inferiores de uma CNN dependem fortemente do conjunto de treinamento, exigindo cuidado na seleção do conjunto representativo de dados. Além disso, a escolha de um conjunto de treinamento representativo geralmente requer um grande esforço humano na seleção de imagens adequadas e considerável poder computacional durante a fase de treinamento (NANNI; GHIDONI; BRAHNAM, 2017). Modelos baseados em aprendizagem profunda, como redes neurais convolucionais (CNNs), têm mostrado grande eficácia na extração automática de características discriminativas (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). No entanto, a aplicação desses modelos para categorização de cenas em larga escala pode ser desafiadora devido à necessidade de treinamento em grandes conjuntos de dados, o que pode levar a problemas de escalabilidade e demandar recursos computacionais significativos (KARPATHY et al., 2014).

Agrupamento é uma técnica de análise de dados que organiza amostras em grupos com base em sua similaridade. Essa abordagem visa identificar padrões intrínsecos nos dados, agrupando elementos que compartilham características semelhantes. O agrupamento desempenha um papel fundamental na resolução de problemas em diversas áreas, destacando-se em aplicações como reconhecimento de padrões, visão computacional, processamento de linguagem natural, bioinformática e análise de redes sociais. Recentemente, a pesquisa de (EZUGWU et al., 2022) abordou diversos desafios enfrentados pelos algoritmos de agrupamento, destacando temas como velocidade de convergência, capacidade de lidar eficientemente com conjuntos de dados de alta dimensão, eficácia na identificação de agrupamentos complexos e escalabilidade para lidar com grandes volumes de dados.

Modelos de categorização podem ser utilizados para solucionar problemas mais amplos do que apenas reconhecimento de cenas. No âmbito da saúde, a base de dados Heart é uma referência importante, oferecendo dados clínicos para previsão de doenças cardíacas. A base de dados Vehicle concentra-se em características relacionadas a veículos, enquanto a base Letter é composta por imagens de letras do alfabeto, ampliando a aplicação de algoritmos para reconhecimento de padrões em caracteres. Em relação à categorização de objetos, as bases de dados CIFAR_10, CIFAR_100 e Caltech_101 são fundamentais para avaliar a capacidade de algoritmos em reconhecer e classificar objetos em imagens (KRIZHEVSKY; HINTON et al., 2009; FEI-FEI; FERGUS; PERONA, 2004). A utilização de mapeamento semântico é crucial para representar o ambiente por meio de um contexto de alto nível, aproximando-se da linguagem humana na descrição de um ambiente.

Modelos de categorização podem ser utilizados para solucionar problemas mais amplos do que apenas reconhecimento de cenas. No âmbito da saúde, a base de dados Heart é uma referência importante, oferecendo dados clínicos para previsão de doenças cardíacas. A base de dados Vehicle concentra-se em características relacionadas a veículos, enquanto a base Letter é composta por imagens de letras do alfabeto, ampliando a aplicação de algoritmos para reconhecimento de padrões em caracteres. Em relação à categorização de objetos, as bases de dados CIFAR_10, CIFAR_100 e Caltech_101 são fundamentais para avaliar a capacidade de algoritmos em reconhecer e classificar objetos em imagens (KRIZHEVSKY; HINTON et al., 2009; FEI-FEI; FERGUS; PERONA, 2004). A utilização de mapeamento semântico é crucial para representar o ambiente por meio de um contexto de alto nível, aproximando-se da linguagem humana na descrição de um ambiente.

Já os problemas de categorização de lugares e cenas são áreas cruciais em visão compu-

tacional e robótica, desempenhando um papel vital na interpretação semântica de ambientes. Enquanto a categorização de lugares concentra-se em rotular espaços específicos, como cozinha ou escritório, a categorização de cenas busca identificar o contexto geral de uma área, como ambiente doméstico ou ambiente de trabalho. Ambas as tarefas são inter-relacionadas, contribuindo para uma visão mais holística da cena e auxiliando sistemas autônomos na tomada de decisões contextualizadas (PRONOBIS, 2011).

Ao enfrentar os desafios da categorização de cenas, a complexidade aumenta consideravelmente devido à ampla diversidade de amostras e categorias presentes em ambientes do mundo real. A vasta gama de dados intrínsecos às cenas contribui para a alta-dimensionalidade do problema, intensificando ainda mais sua complexidade (ZHOU et al., 2017). A escalabilidade da abordagem torna-se uma consideração crucial, especialmente ao lidar com um grande número de cenas distintas e suas características específicas. Nesse cenário, métodos convencionais de categorização de cenas podem enfrentar limitações significativas ao tentar abordar a complexidade e a variabilidade inerentes a ambientes dinâmicos (QUATTONI; TORRALBA, 2009).

Além disso, é importante destacar que a aplicação prática desses desafios se manifesta em conjuntos de dados específicos, como o Scenes_15 (LAZEBNIK; SCHMID; PONCE, 2006), Indoors_67 (QUATTONI; TORRALBA, 2009), SUN_397 (XIAO et al., 2010) e SUN_RGBD (SONG; LICHTENBERG; XIAO, 2015). Esses conjuntos representam casos reais nos quais a categorização de cenas enfrenta diversas nuances e variações, enfatizando a necessidade de métodos escaláveis e adaptáveis para lidar eficientemente com a diversidade presente em ambientes complexos e dinâmicos.

A categorização eficiente de cenas em larga escala exige não apenas a extração de características robustas, mas também a capacidade de lidar com a diversidade intrínseca e a quantidade expressiva de dados. Os desafios enfrentados nesse processo enfatizam a necessidade de modelos de aprendizado de máquina que possam superar limitações de escalabilidade, alta-dimensionalidade e generalização para alcançar resultados eficazes em ambientes complexos. A busca por soluções inovadoras para esses desafios torna-se fundamental para o avanço contínuo da categorização de cenas em contextos do mundo real.

O modelo proposto nesta tese, denominado Múltiplos Mapas Auto-organizáveis Especialistas em Uma Categoria (*Multiple One-Category Expert Self-Organing Maps* (MOCE-SOM)), representa um avanço na abordagem de categorização de cenas em ambientes de visão computacional. Essa arquitetura inovadora incorpora diversas características que visam superar desafios específicos associados a problemas complexos de categorização. A multi-modularidade

do MOCE-SOM é uma das suas características distintivas, permitindo a divisão do modelo em módulos especializados em categorias específicas. Cada módulo atua como um especialista em uma categoria particular, proporcionando uma abordagem mais precisa e discriminativa para a categorização de cenas (JÚNIOR; ARAÚJO, 2022).

A especialização em uma categoria específica por meio de módulos dedicados no MOCE-SOM contribui para a eficácia do modelo na identificação de padrões complexos e na diferenciação entre diferentes tipos de cenas. Essa característica é particularmente crucial quando se lida com um grande número de amostras e categorias, pois permite uma representação mais específica e adaptável. Além disso, ao incorporar o conceito de categorizadores de uma categoria em cada módulo especializado, o MOCE-SOM aprimora a capacidade de generalização, destacando-se na categorização de cenas em ambientes diversos e dinâmicos (KHAN; MADDEN, 2014; SILVA et al., 2023).

A aprendizagem de relevâncias intra e inter-categorias no MOCE-SOM é um característica incorporada ao modelo que busca lidar eficazmente com a complexidade decorrente de dados com muitas amostras e categorias. A capacidade do modelo de discernir nuances específicas dentro de cada categoria, combinada com a distinção entre diferentes categorias, contribui para uma categorização mais discriminativa. Adicionalmente, a abordagem de relevâncias proporciona uma maneira de mitigar o problema da alta dimensionalidade dos dados, tornando a representação mais eficiente e facilitando a identificação de padrões significativos (JÚNIOR; ARAÚJO, 2022).

Com o intuito de melhorar a acurácia do modelo, se pode lançar mão do estudo de dsitirbuiçãod os dados. Dessa forma, fazendo a incorporação de inferência variacional na etapa final de refinamento do MOCE-SOM contribui para melhora da escalabilidade e da capacidade de generalização do modelo. A inferência variacional é fundamental para lidar com a complexidade computacional associada ao refinamento bayesiano, facilitando o cálculo em cenários nos quais a abordagem tradicional seria computacionalmente muito demandante.

O MOCE-SOM com suas características de multi-modularidade, especialização em categorias, aprendizado de relevâncias, múltiplas camadas e inferência variacional, representa uma abordagem avançada e eficiente para a categorização de cenas em ambientes desafiadores de visão computacional. Essas características integradas destacam a capacidade do MOCE-SOM de solucionar problemas complexos associados a muitas amostras, categorias, alta dimensionalidade e escalabilidade em problemas de categorização de cenas.

1.1 OBJETIVOS

Esta tese apresenta o desenvolvimento de um modelo de agrupamento destinado à categorização de cenas, com aplicabilidade em diversas áreas, além da categorização de cenas, incluindo mineração de dados e categorização de objetos. O modelo proposto visa enfrentar os desafios apresentados pela pesquisa recente e contribuir para o avanço dessas áreas em rápido desenvolvimento.

Objetivo Geral:

- Desenvolver um modelo de categorização de cenas que se caracteriza por ter baixo custo computacional, ter aprendizagem incremental, ser modularizado, paralelizável e escalável.

Objetivos Específicos:

Propor um modelo de multi-módulos de Mapa Auto-Organizáveis (SOMs) com múltiplas camadas para a tarefa de categorização.

- Desenvolver um modelo capaz de aprender a relevância dos atributos para cada categoria, com base nas informações intra-categoria.
- Propor um processo de composição de amostras adequado para propagação das amostras para um modelo de SOM com múltiplas camadas.
- Estabelecer um procedimento para a incorporação de informações intra-categoria na determinação da relevância dos atributos.
- Implementar um processo de inferência bayesiana através da inferência variacional em um modelo de agrupamento, visando melhorias no desempenho e na generalização do modelo.

1.2 HIPÓTESE DE PESQUISA

A hipótese de pesquisa desta tese é que um modelo de Mapas Auto-organizáveis pode ser escalável para a avaliação de categorização de dados complexos de alta dimensionalidade e com uma grande quantidade de categorias. Isto pode ser alcançado utilizando aprendizado das relevâncias intra-categoria, incorporando informações inter-categoria e refinando essas

relevâncias através de métodos bayesianos. Além disso, a hipótese propõe que este modelo pode ser ampliado em uma arquitetura de múltiplas camadas através da propagação de amostras baseada na composição com os protótipos do modelo.

1.3 PUBLICAÇÕES

Ao longo da elaboração desta tese, um conjunto de artigos foi desenvolvido, sendo o primeiro dedicado à criação de um método de Associação de Dados Espaço-Temporais (STDA) para o mapeamento de objetos. Este trabalho desempenhou um papel crucial no amadurecimento dos conceitos da área, na compreensão das técnicas existentes e foi submetido a testes em diversos ambientes, apresentando resultados promissores. A primeira iteração do modelo proposto nesta tese, denominado *Subspace Clustering Multi-module Self-organizing Map* (SC-MuSOM) (JÚNIOR; ARAÚJO, 2022), representa uma abordagem inovadora na categorização de dados em mineração, objetos e cenas. Este modelo é fundamentado em uma única camada de *Subspace Clustering Self-organizing Map* (SC-SOM), onde a aprendizagem de relevâncias intra-categorias desempenha um papel central no processo de categorização. Ao ser aplicado à categorização de cenas (SILVA et al., 2023), o SC-MuSOM demonstrou eficácia, sendo reconhecido e premiado no *Latin American Robotics Symposium* (LARS). Em reconhecimento a esse feito, está sendo preparada uma versão estendida para publicação no *Journal of Intelligent and Robotic Systems* (JIRS). Uma versão aprimorada, denominada *Subspace Clustering Multi-Layer Self-organizing Map* (SC-MLSOM), foi posteriormente submetida ao *International Conference on Artificial Neural Networks* (ICANN) 2024, introduzindo a ideia de multi-camadas. Essa extensão envolve a composição de amostras para a propagação entre as camadas, resultando em uma representação mais rica e complexa dos dados. No entanto, o SC-MLSOM ainda concentra-se predominantemente na aprendizagem de relevâncias intra-categorias. O modelo proposto nesta tese está em processo de escrita para submissão em periódico da alto impacto.

O modelo proposto nesta tese representa uma expansão significativa da abordagem anteriormente apresentada no SC-MLSOM. Esta evolução não apenas incorpora a influência de relevâncias inter-categorias, mas também aprimora a estrutura do modelo, resultando no MOCE-SOM, por meio da introdução de uma etapa de refinamento no final da rede, utilizando técnicas de inferência variacional. Esta extensão tem como objetivo proporcionar uma capacidade aprimorada de generalização e uma representação mais robusta dos dados, visando

melhorar o desempenho do modelo em cenários complexos de categorização. As contribuições deste trabalho não se limitam apenas ao avanço das capacidades do MOCE-SOM, mas também oferecem insights valiosos para o desenvolvimento contínuo de modelos na área de reconhecimento de padrões e categorização.

1.4 ORGANIZAÇÃO DO DOCUMENTO

Os próximos capítulos desta tese são organizados da seguinte maneira: o Capítulo 2 realiza uma revisão da literatura abordando os trabalhos relacionados a mapeamento semântico e categorização de cenas. O Capítulo 3 apresenta métodos comparativos fundamentados em k-NN, LVQ, além de fornecer uma visão geral do Mapa Auto-Organizável (SOM), que servirá como base para a solução proposta. No Capítulo 4, abordamos o modelo proposto nesta pesquisa, delineando e justificando as escolhas feitas, detalhando o comportamento do modelo e explicando o seu funcionamento. O Capítulo 5 expõe, discute e sintetiza os resultados experimentais do modelo proposto até o momento, comparando-os com uma solução referenciada na literatura. Finalmente, o Capítulo 6 encerra o trabalho com um resumo dos resultados, destacando as limitações dos modelos propostos e delineando possíveis direções para trabalhos futuros.

2 CATEGORIZAÇÃO DE CENAS E LUGARES: FORMALIZAÇÃO DO PROBLEMA E EXTRAÇÃO DE CARACTERÍSTICAS

Este capítulo apresenta uma revisão abrangente da literatura, focando nos principais componentes do problema abordado nesta tese de doutorado. Uma atenção especial é dada aos modelos que estão mais alinhados com a abordagem escolhida para a solução proposta. A estrutura do capítulo é delineada começando pela definição do problema, com ênfase na categorização de cenas. Em seguida, é realizada análise dos extratores de características *handcrafted* e das abordagens baseadas em redes neurais para a extração de informações visuais. O capítulo culmina com uma discussão a respeito de como a proposta delineada neste trabalho de doutorado pode incorporar e beneficiar-se do conhecimento adquirido ao longo da revisão literária apresentada.

2.1 CATEGORIZAÇÃO DE CENAS: DEFINIÇÃO DO PROBLEMA

O propósito da categorização de cenas é efetuar o agrupamento de imagens que representam cenas em uma das categorias pré-definidas, como praia, cozinha, livraria, escritório, entre outras. Isso é realizado com base no conteúdo do ambiente retratado na imagem, nos objetos presentes na cena e na disposição espacial desses elementos. A compreensão visual de uma cena demanda o raciocínio sobre ambientes complexos que são comuns no cotidiano das pessoas. A capacidade de reconhecer categorias visuais, tais como objetos, ações e eventos, é considerada essencial para um sistema visual eficiente (ZENG et al., 2021).

Os seres humanos possuem a capacidade de categorizar cenas naturais (GREENE; OLIVA, 2009; WALTHER et al., 2011). Contudo, tal tarefa representa um desafio para as máquinas devido à ambiguidade semântica inerente a muitas cenas e às consideráveis variações nas categorias, derivadas de fatores como a iluminação, o ângulo de visão, a escala, a distância da imagem, entre outros (ZENG et al., 2021). A categorização de cenas tem sido objeto de pesquisa ativa por várias décadas e possui diversas aplicações, incluindo recuperação de imagens baseada em conteúdo (VOGEL; SCHIELE, 2007; ZHENG; YANG; TIAN, 2017), navegação de robôs (ZHANG; YU; HE, 2017; HOU et al., 2019), vigilância por vídeo inteligente (ZHANG et al., 2012; SREENU; DURAI, 2019), e realidade aumentada (BEHZADAN; KAMAT, 2011; NEE et al., 2012).

A categorização de cenas pode ser associada a tarefas relacionadas de visão, como categorização de objetos e texturas, que envolvem a representação e agrupamento de características.

No entanto, em contraste com as imagens de objetos (centradas no objeto) ou texturas (incluindo apenas texturas), as imagens de cena são mais complexas, exigindo uma exploração mais aprofundada de seu conteúdo, como sua semântica (por exemplo, objetos, texturas, fundo), a organização da cena e a semântica das conexões entre seus componentes. A dificuldade inerente está relacionada à complexidade e à considerável variação nas cenas, além da alta dimensionalidade dos dados. Entre os desafios significativos na categorização de cenas, destacam-se as grandes variações intra-categoria, a ambiguidade semântica e a eficiência computacional (ZENG et al., 2021).

O aumento do interesse em redes neurais, especialmente no aprendizado profundo (KRIZHEVSKY; SUTSKEVER; HINTON, 2017), tem tido um impacto considerável na área de visão computacional, conduzindo a uma aplicação intensiva em tarefas como detecção e categorização de objetos (GIRSHICK, 2015; REN et al., 2015), segmentação semântica (BADRINARAYANAN; KENDALL; CIPOLLA, 2017) e categorização de cenas (ZHOU et al., 2014; GONG et al., 2014). Em 2012, a categorização de objetos com o conjunto de dados em larga escala, como ImageNet (DENG et al., 2009), alcançou um notável aprimoramento de desempenho com a introdução da Rede Neural Convolucional (CNN) denominada AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), o que impulsionou significativamente sua utilização em visão computacional. A partir desse marco, a pesquisa na categorização de cenas passou a substituir algoritmos determinísticos por técnicas de aprendizado profundo, capazes de aprender representações discriminativas diretamente dos dados. Os recentes avanços na aprendizagem profunda abriram para a categorização de cenas em conjuntos de dados em grande escala (ZHOU et al., 2014; ZHOU et al., 2017; XIAO et al., 2010; GONG et al., 2014; LIU et al., 2019a).

A considerável variação intra-categoria, em grande parte, origina-se de fatores intrínsecos à própria cena. Em termos desses fatores intrínsecos, cada cena pode apresentar uma diversidade significativa entre suas imagens, variando notavelmente nos objetos e no fundo. As condições de aquisição de uma imagem, como alterações na iluminação, ponto de vista, escala, oclusão, sombreamento, falta de foco e movimento, contribuem para as significativas variações intra-categoria. Outros desafios podem ser introduzidos por artefatos de digitalização, presença de ruído e distorção de filtragem. Por exemplo, a linha superior da Figura 1 mostra três shoppings com diferentes condições de iluminação, ângulo de visão e objetos (ZENG et al., 2021).

Imagens provenientes de categorias distintas podem partilhar objetos semelhantes, texturas, fundo, entre outros elementos. Dessa forma, a semelhança em suas aparências visuais pode resultar em erros de categorização (BOUTELL et al., 2004; LÓPEZ-CIFUENTES et al., 2020).

Figura 1 – Ilustrações de grande variação intra-categórica e ambiguidade semântica. Topo (grande variação intra-categoria). Os shoppings são bem diferentes devido à iluminação e conteúdo geral. Abaixo (ambigüidade semântica). Fonte: (ZENG et al., 2021).

A Figura 1, por exemplo, ilustra uma forte correlação visual entre três ambientes distintos: arquivo, livraria e biblioteca. A adição de novas categorias de cenas ampliaria ainda mais a ambiguidade semântica. Além disso, a anotação da categoria de cena é subjetiva, influenciada pela experiência dos anotadores, permitindo que uma cena seja atribuída a várias categorias semânticas (BOUTELL et al., 2004; ZHANG; ZHOU, 2007).

2.1.1 Mapeamento Semântico e Categorização de Lugares

Existe uma clara tendência na robótica em modelar e implementar mecanismos comportamentais inspirados nos processos cognitivos da psicologia humana. O tratamento adequado da semântica tem aberto novos caminhos de pesquisa na navegação de robôs, possibilitando, entre outras coisas, uma representação mais elaborada da informação. Nos primeiros trabalhos, a navegação era comumente tratada por navegadores puramente cartesianos, que interpretavam o ambiente como uma série de áreas acessíveis (KOSTAVELIS; GASTERATOS, 2017). Posteriormente, ela era abordada por meio da teoria de grafos. Em contraste, pesquisas mais recentes utilizam a informação semântica como uma maneira de enriquecer a descrição do ambiente na navegação robótica (CRESPO et al., 2020).

Nos últimos anos, tem havido um interesse crescente em incorporar informações de alto nível em diversas aplicações de robótica, visando aumentar a capacidade dos robôs e possi-

bilitar respostas adequadas a eventos imprevistos. Nessa linha, a robótica móvel começou a integrar informações semânticas em tarefas de navegação, resultando no conceito emergente de navegação semântica. Esse tipo de navegação busca simular a forma como humanos e robôs percebem um ambiente, utilizando significados para descrever um ambiente explorado de maneira mais próxima a linguagem humana (KOSTAVELIS; GASTERATOS, 2017).

Ao contrário da abordagem tradicional que se baseia em vértices de grafos ou coordenadas cartesianas, a navegação semântica visa reproduzir a forma como os humanos se orientam e movem em um ambiente. Os seres humanos tendem a utilizar conceitos ao descrever seu movimento em um ambiente, como exemplificado por expressões como "Eu estava no quarto e fui para a cozinha beber um copo de água". Em resposta a essa tendência, esforços têm sido direcionados para incorporar conceitos semânticos do ambiente na navegação de robôs (CRESPO et al., 2020).

A representação semântica possibilita que os robôs construam mapas compreensíveis aos humanos, diminuindo a lacuna entre interpretação geométrica e conceitos de alto nível. Os mapas semânticos incorporam elementos altamente abstratos para representar um ambiente (KOSTAVELIS; GASTERATOS, 2015). Na navegação semântica, é possível utilizar as relações entre atributos de alto nível em conjunto com informações geométricas provenientes de um mapa métrico de baixo nível. A informação de alto nível, geralmente capturada por diferentes sensores, desempenha um papel crucial no reconhecimento de lugares ou objetos.

(PRONOBIS; CAPUTO, 2007) apresentaram um método baseado em aparência para o reconhecimento de lugares, utilizando um descritor de imagem global robusto seguido por um categorizador. Este método foi avaliado em diferentes posições de câmeras, variações de horários do dia e estações do ano, demonstrando robustez. Posteriormente, em um trabalho subsequente, (MOZOS et al., 2007) propuseram uma abordagem para classificar o posicionamento de um robô móvel com base em semântica. Os autores empregaram o algoritmo AdaBoost para construir um classificador forte que realiza a categorização *on-line* das posições ao longo do trajeto de um robô móvel utilizando um modelo de Markov. Além disso, o modelo pode aprender mapas topológicos a partir de mapas geométricos, aplicando um procedimento de categorização semântica em conjunto com um método probabilístico.

Posteriormente, (PRONOBIS et al., 2010) investigaram a capacidade de um robô em representar conhecimento sobre o espaço em que está inserido, assim como seu posicionamento. Nesse sentido, um robô móvel identifica e reconhece categorias semânticas em um ambiente restrito, processando dados combinados provenientes de informações visuais e de alcance de

um sensor a laser. A solução proposta rotula o espaço semântico em tempo real, integrando dados ao longo do tempo e espaço através do uso de um modelo de *Support Vector Machine* (SVM). A avaliação experimental abrangeu duas plataformas robóticas em ambientes distintos, utilizando uma base de dados realista *offline*.

Em seu trabalho, (KRISHNAN; KRISHNA, 2010) introduziram um algoritmo de exploração baseado em visão que constrói um mapa de imagens e lida com uma combinação de mapas semânticos e topológicos. No nível superior, o mapa é representado como um grafo de construções semânticas, em que cada nó é rotulado como, por exemplo, sala ou corredor, e cada aresta define uma transição, como uma porta conectando duas regiões semânticas. Cada nó semântico está associado a um grafo topológico, um conjunto de nós que representam imagens para a construção do conceito semântico. No nível inferior, o grafo topológico codifica valores e relações métricas, onde cada nó contém as informações de posicionamento do robô ao capturar uma imagem, e as arestas representam translação e/ou rotação de uma imagem para outra. O algoritmo utiliza uma combinação de informações locais e globais para realizar inferências sobre a localização. Durante o processo de exploração, são representadas as regiões de transição, que são as passagens de um ambiente para outro, e ao visitar todas as regiões de transição, a exploração é encerrada. Regiões e técnicas baseadas em grafo são empregadas para fechar *loops*, ajustando as métricas das poses do robô. Construções semânticas são rotuladas usando uma representação de *Bag-of-Visual Word* (BoVW) com um SVM probabilístico (CSURKA et al., 2004a).

2.1.2 Categorização de Lugares e Cenas

A categorização de lugares, cenas e imagens representa três áreas distintas, mas interrelacionadas, na análise e interpretação de imagens. A categorização de imagens é uma tarefa geral que envolve atribuir rótulos ou categorias a imagens com base em seu conteúdo visual, sem foco específico no tipo de ambiente. A categorização de cenas, por outro lado, foca na identificação de contextos e ambientes visuais amplos, como praias, florestas ou ruas urbanas. Essa categorização de cenas é muitas vezes utilizada como uma etapa preliminar para a categorização de lugares, que é mais específica e envolve a identificação de localizações particulares, como uma cozinha ou uma biblioteca. Assim, enquanto a categorização de cenas fornece um contexto mais amplo e geral, a categorização de lugares aplica esses contextos para uma análise mais detalhada e específica dos espaços.

A categorização de cenas refere-se à categorização de imagens com foco na identificação de contextos amplos, como paisagens urbanas, interiores de edifícios, praias ou montanhas. Métodos nessa área visam entender o conteúdo global da imagem, muitas vezes utilizando características de alto nível para capturar informações contextuais relevantes. Trabalhos pioneiros, como Oliva e Torralba (2001), exploram a representação de cenas e a influência de características globais na categorização.

Por outro lado, a categorização de lugares se concentra na identificação e/ou categorização de ambientes específicos, como restaurantes, aeroportos ou escritórios. Essa abordagem considera características mais detalhadas e específicas do ambiente, muitas vezes relacionadas à arquitetura e à presença de elementos distintivos. Trabalhos como Quattoni e Torralba (2009) abordam a categorização de lugares considerando características locais e globais. Os avanços no desenvolvimento de categorização de cenas podem ser utilizados na categorização de lugares.

A categorização de lugares é vista como uma das etapas fundamentais para o mapeamento semântico, podendo basear-se em objetos, aparência ou segmentação (KOSTAVELIS et al., 2016) relativos a uma determinada cena. Diversos desafios estão associados a essa tarefa, como, por exemplo, a definição de métodos para extrair características das imagens utilizadas no aprendizado. Outro ponto de pesquisa envolve o tratamento de nuvens de pontos para a formação de representações 3D. Além disso, existem desafios relacionados à capacidade de generalização de padrões aprendidos, ou seja, um robô pode ter dificuldade em identificar ou criar um tipo de ambiente não previamente visto. Uma das dificuldades relatadas no contexto da semântica na etapa de categorização de lugares é a generalização dos rótulos aprendidos para novos rótulos que não foram previamente aprendidos, ou a detecção de tipos de ambientes ausentes de um dicionário inicialmente aprendido (KOSTAVELIS et al., 2016).

Hermans, Floros e Leibe (2014) introduziram um método de segmentação semântica para distinguir os tipos de locais em cenas 3D. O método realiza reconstrução 3D, seguida pela classificação da nuvem. O processo de rotulação considera que pontos vizinhos são afetados em função da distância, cor e da orientação. Por outro lado, Ranganathan (2012) propôs a rotulação do local através da segmentação da sequência de imagens. Esse método *on-line* depende da detecção de mudanças entre as imagens para inferir sobre o parâmetro em um modelo bayesiano. O método permite segmentar fluxos visuais, e novos locais podem ser categorizados por meio de testes estatísticos de hipóteses. Adicionalmente, Choudhary et al. (2014) uma abordagem que combina descoberta e modelagem de objetos com um sistema

Simultaneous Localization and Mapping (SLAM). Essa integração permite a utilização de objetos descobertos e modelados como pontos de referência para auxiliar na localização do robô de forma online. Além disso, o sistema gera um banco de dados de modelos de objetos detectados para uso em futuras tarefas de SLAM ou robótica de serviço. A novidade está na abordagem de descoberta de objetos em tempo real e na incorporação desses objetos como marcadores no processo de mapeamento e localização do robô, resultando em uma melhoria nos resultados do SLAM, especialmente na detecção de fechamentos de loop em mapas maiores.

Zhang et al. (2015) apresentaram uma estrutura multimodal para memorização de cenas que realiza segmentação e extração de características em várias escalas. Em cada escala individual, ocorre uma etapa de categorização. A fusão de modalidades e uma etapa de refinamento empregam um *Conditional Random Field* (CRF). Por sua vez, Mousavian, Košecká e Lien (2015) exploraram estruturas artificiais para aprimorar o reconhecimento de locais com base na extração de características, descartando características relacionadas a edificações e ao fundo de um cenário. Esse método aumenta a precisão de vários categorizadores. Já Paul et al. (2012) empregaram processos gaussianos para a categorização semântica não-supervisionada de cenas 3D. A nuvem de pontos é utilizada para calcular uma malha triangular e os vetores normais dessas regiões, visando segmentar mais efetivamente a imagem.

(BALASKA et al., 2020) apresentaram uma nova técnica não-supervisionada para identificar semanticamente a posição de um agente autônomo em ambientes desconhecidos. Quando o robô explora um determinado caminho pela primeira vez, a detecção baseada em grafos leva o agente a determinar semanticamente seus arredores em percursos futuros, mesmo que as condições de iluminação sejam alteradas. A técnica de agrupamento semântico proposta explora *Louvain Community Detection Algorithm* (LCDA) (BLONDEL et al., 2008), que é capaz de reconhecer grupos de medidas com similaridade consistentes. Os grupos são combinados com informações métricas, fornecidas pela odometria do robô empregando hierarquia método de agrupamento aglomerativo.

2.2 EXTRAÇÃO DE CARACTERÍSTICAS DE CENAS

A extração de características de imagens é um aspecto crítico na área de visão computacional, desempenhando um papel essencial na análise e interpretação eficazes de informações visuais complexas. Esse processo é fundamental para transformar a complexidade intrínseca das imagens em representações mais gerenciáveis e significativas, permitindo uma análise com-

putacional mais efetiva.

Os algoritmos utilizados na categorização de cenas tendem a apresentar complexidade quadrática, $O(n^2)$. A extração de características é crucial para a representação eficiente de informações visuais, reduzindo a dimensionalidade dos dados e condensando características discriminativas. Essa condensação é fundamental para lidar com a vastidão de dados visuais, melhorando o desempenho em tarefas específicas. Além disso, a extração de características desempenha um papel crucial na transferência de aprendizado, como no caso dos modelos pré-treinados em conjuntos de dados abrangentes. A compreensão semântica das imagens é aprimorada pela extração de características, permitindo a identificação de padrões, texturas e formas relevantes.

As duas abordagens de extrações de características consideradas são as de características *handcrafted* e de aprendido profundo (*deep learning-based*). As principais diferenças e vantagens entre essas duas abordagens são as seguintes (NANNI; GHIDONI; BRAHNAM, 2017):

- Vantagens do uso de *Handcrafted*:
  - Experência e Conhecimento: Características *handcrafted* são criados a partir de conhecimento adquirido ao longo dos anos por especialistas em visão computacional. Isso pode resultar em descritores de imagem robustos e eficientes para problemas específicos.
  - Rastreabilidade: O processo de criação desses descritores é mais transparente e rastreável, o que pode ser importante para aplicações que exigem explicação da decisão.
  - Eficiência Computacional: Algumas abordagens foram projetadas especialmente para ser rápidas e eficientes computacionalmente, o que torná-los adequados para dispositivos com recursos limitados.
- Desvantagens do uso de *handcrafted*:
  - Dependência de Domínio: Características *handcrafted* podem ser especializadas para um determinado domínio ou tipo de imagem. Isso poderia limitar sua generalização para outros tipos de imagens ou problemas de categorização.
  - Manutenção Manual: As alterações no domínio de aplicação ou no meio ambiente podem exigir a criação ou adaptação de novos descritores, o que é um processo manual e potencialmente demorado.

- Vantagens de caraterísticas baseadas em Aprendizado Profundo:
  - Auto-Aprendizagem: A aprendizagem profundo permite que o sistema aprenda automaticamente as características relevantes de uma imagem a partir de grandes conjuntos de dados.
  - Generalização: Caracteríticas extraídas por redes neurais profundas tendem a ser mais genéricas e podem ser aplicadas a vários problemas de categorização de imagem sem necessidade de adaptações manuais.
- Desvantagens de caraterísticas baseadas em Aprendizado Profundo:
  - Recursos Computacionais: O treinamento e a execução de redes neuradas profundas podem exigirem muitos recursos computacionais, o que poderia limitá-las para dispositivios ou ambientes com recursos limitadas.
  - Complexidade: A interpretação e a explicação dos resultados de redes neurais profundas podem ser mais difíceis em comparação aos descritores *handcrafted*, o que pode ser um inconveniente para aplicação em domínios onde a interpretação é importante.

As características *handcrafted* oferecem uma abordagem especializada e transparente, enquanto as não artesanalistas oferecem um processo automático de aprendizagem e generalização mais robusto. A escolha entre essas abordagenses depende dos requisitos específicos do problema, como a complexidade do domínio, a disponibilidade de recursos computacionals e a necessidade de explicação da solução.

2.3 MÉTODOS DE EXTRAÇÃO DE CARACTERÍSTICAS *HANDCRAFTED*

Os extratores de características *handcrafted* referem-se a métodos de extração de características em que as representações discriminativas são projetadas manualmente, em oposição a serem aprendidas automaticamente por algoritmos de aprendizado de máquina. Nesse contexto, *handcrafted* denota a abordagem artesanal na definição de características específicas com base no conhecimento prévio do problema em questão. Esses extratores são elaborados com base no conhecimento do especialista no domínio e na compreensão das características relevantes para a tarefa desejada.

Em oposição às abordagens automatizadas, como CNN, nas quais as características são aprendidas automaticamente a partir dos dados, os extratores *handcrafted* são projetados com intenção prévia. Isso envolve a seleção cuidadosa de características, muitas vezes levando em consideração conhecimentos específicos do domínio. Esses extratores podem incluir operações como detecção de bordas, extração de texturas, identificação de pontos de interesse ou qualquer outra técnica que seja considerada relevante para a tarefa em questão. A manutenção e adaptação desses extratores para diferentes conjuntos de dados ou tarefas específicas podem exigir esforço manual significativo. Serão abordados alguns algoritmos para extração de características: *Histogram of Oriented Gradients* (HoG) (DALAL; TRIGGS, 2005), *Scale-Invariant Feature Transform* (SIFT) (LOWE, 2004), *Speeded up robust features* (SURF) (BAY et al., 2008).

2.4 MÉTODOS BASEADOS EM APRENDIZAGEM PROFUNDA

Extratores *handcrafted* são frequentemente comparados a extratores profundos que se baseiam na aprendizagem automática de representações hierárquicas complexas a partir dos dados brutos. Nesta seção, será realizada uma revisão dos métodos de aprendizagem profunda aplicados à categorização de cenas. A arquitetura mais prevalente para a extração de características nesse contexto é conhecida como CNN (LECUN; BENGIO; HINTON, 2015).

2.4.1 Fundamentos da CNN

Originalmente, pode-se argumentar que as CNNs consideram que sua entrada consiste em imagens. Logo, sua arquitetura foi concebida para lidar com esse tipo específico de dados. As camadas dentro da CNN são compostas por nodos organizados em três dimensões, a dimensionalidade espacial da entrada (altura e largura) e a profundidade. Em uma CNN, um nodo de qualquer camada só se conecta a uma pequena região da camada anterior. Na prática, isso significaria que, para o exemplo dado anteriormente, o volume de entrada terá uma dimensionalidade de $64 \times 64 \times 3$ (altura, largura e profundidade), levando a uma camada de saída final composta por uma dimensionalidade de $1 \times 1 \times n$ (no qual n representa o número possível de categorias) (O'SHEA; NASH, 2015).

As CNNs são compostas por três tipos de camadas: camadas convolucionais, camadas de agrupamento e camadas totalmente conectadas. A arquitetura de uma CNN é formada por

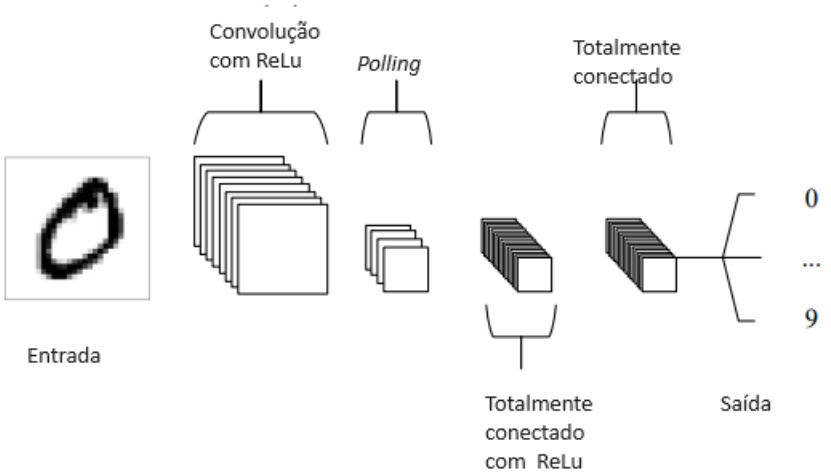

Figura 2 – Uma arquitetura CNN simples, composta por apenas cinco camadas. (O'SHEA; NASH, 2015).

essas camadas empilhadas, como apresentado na Figura 2. A funcionalidade básica da CNN pode ser dividida em quatro áreas principais.

1. Camada de Entrada: Nesta etapa, a rede recebe os valores dos pixels da imagem como entrada.

2. Camada Convolucional: Nesta camada, a saída dos nodos é determinada ao calcular o produto escalar entre os pesos associados a regiões locais da entrada e a região conectada ao volume de entrada. A função de ativação ReLu é aplicada para gerar a saída dos nodos desta camada.

3. Camada de Subamostragem: Esta camada realiza uma redução na dimensionalidade espacial da entrada, reduzindo assim o número de informações contidas na ativação.

4. Camadas Totalmente Conectadas: Estas camadas operam de maneira semelhante às redes Multilayers Perceptrons (MLP), visando reconhecer as categorias em suas saídas. A aplicação da função de ativação ReLu entre essas camadas pode ser sugerida para melhorar o desempenho da rede.

A camada convolucional desempenha um papel crucial na operação das redes neurais convolucionais (CNNs). Seus parâmetros incluem valores para o uso de *kernels*, que são aprendidos durante o treinamento da rede. Esses *kernels* geralmente possuem uma dimensionalidade espacial reduzida, mas são aplicados em toda a profundidade da entrada. Ao entrar em uma camada convolucional, o sinal é submetido a um processo de convolução com cada filtro, resultando na produção de um mapa 2D. O deslocamento de um *kernel* ao longo da representação da imagem na entrada da camada permite o cálculo do produto escalar para cada

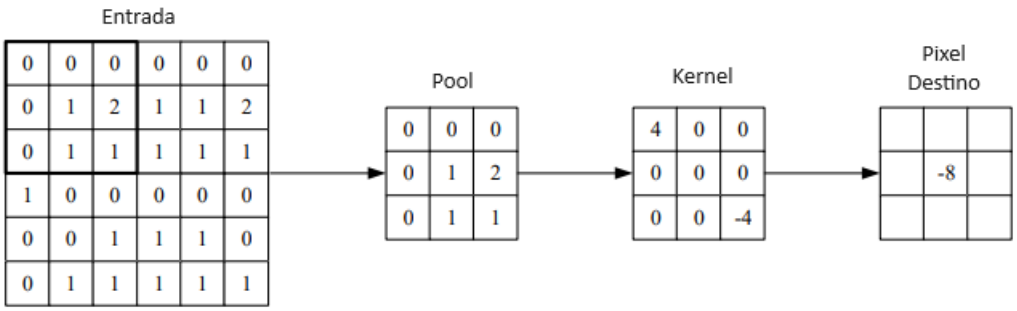

Figura 3 – Uma representação visual de uma camada convolucional. O elemento central do *kernel* é colocado sobre o vetor de entrada, do qual é então calculado e substituído com uma soma ponderada de si mesmo e quaisquer pixels próximos. (O'SHEA; NASH, 2015).

posição do *kernel* (Figura 4). Quando os *kernels* encontram regiões na imagem semelhantes a eles, ocorre uma ativação mais intensa do padrão.

O treinamento das imagens de entrada implica a aplicação de janelas em cada matriz de dados (sinal de entrada), no qual a dimensionalidade dessa região é conhecida como o tamanho do campo receptivo do neurônio. A magnitude da conectividade através da profundidade é quase sempre igual à profundidade da entrada. Por exemplo, considerando uma imagem de entrada para a rede com dimensões de $64 \times 64 \times 3$ (uma imagem colorida RGB com dimensões de 64×64) e um campo receptivo de tamanho 6×6, resulta em um total de 108 pesos em cada nó dentro da camada convolucional ($6 \times 6 \times 3$, onde 3 é a magnitude da conectividade através da profundidade do volume). Além disso, as camadas convolucionais são capazes de reduzir significativamente a complexidade do modelo, otimizando sua saída por meio de três hiperparâmetros: profundidade, passo e configuração de preenchimento de zeros.

Na pesquisa subsequente, são inicialmente apresentados os principais conceitos da CNNs para categorização. Em seguida, realiza-se uma revisão das representações de cena fundamentadas em CNN existentes.

2.4.2 Arquitetura da CNN

As CNNs representam modelos de aprendizagem profunda projetados para extrair características com alta capacidade discriminativa. As características de uma cena são extraídas de CNNs pré-treinadas, e em alguns casos, são combinadas com características obtidas por algoritmos determinísticos, sendo posteriormente categorizadas por meio de uma abordagem conhecida como *Bag-of-Visual Word* (BoVW) (ZHENG; YANG; TIAN, 2017). Em seguida, são realizados ajustes na CNN visando adaptar suas últimas camadas. É relevante destacar que

foram desenvolvidas CNNs específicas para a tarefa de categorização de cenas.

2.4.2.1 Modelo CNN Pré-treinado

A arquitetura da rede desempenha um papel crucial no desempenho dos modelos profundos. Inicialmente, o modelo convencional de CNN para representação de características e categorização foi o AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Posteriormente, a VGGNet, desenvolvida por Simonyan e Zisserman (2014), empregou vários pequenos *kernels* empilhados, produzindo resultados superiores a um único *kernel* de convolução grande. A aplicação de não-linearidades em múltiplos mapas de características constrói representações mais discriminativas, embora reduza o tamanho do campo receptivo para redes maiores.

Motivados pela ideia de que apenas um conjunto de nodos desempenha um papel efetivo na representação de características, Szegedy et al. (2015) propuseram um módulo denominado Inception para realizar uma abordagem esparsa de CNN, efetuando extração hierárquica de características. No entanto, o aumento constante da profundidade da CNN pode resultar no desaparecimento do gradiente durante os ajustes de pesos.

Em situações com dados de treinamento escassos, modelos pré-treinados podem ser empregados para extrair representações eficazes de conjuntos de dados pequenos. O treinamento da CNN em conjuntos de dados de grande escala, como o ImageNet (DENG et al., 2009) e os Places (ZHOU et al., 2014; ZHOU et al., 2017), resulta em aprendizado de representações visuais mais discriminativas. Esses modelos pré-treinados podem ser utilizados efetivamente em outras tarefas, apresentando melhor desempenho em comparação com inicializações aleatórias (YOSINSKI et al., 2014). A transferibilidade diminui à medida que a similaridade entre as tarefas de destino e origem diminui (YOSINSKI et al., 2014).

CNNs pré-treinadas como extratores de características podem ser categorizadas em duas abordagens. As CNNs centradas em objetos são inicialmente treinadas em conjuntos de dados de objetos, como o ImageNet (DENG et al., 2009), e posteriormente aplicadas no contexto de cenas. No entanto, as CNNs centradas em cenas, exemplificadas pelos conjuntos de dados Places (ZHOU et al., 2014; ZHOU et al., 2017), mostram maior eficácia na extração de características relacionadas a cenas em comparação com as centradas em objetos, devido à diversidade nas características das imagens (ZHOU et al., 2014).

2.4.2.2 *Modelo CNN Ajustado*

As CNNs pré-treinadas atuam como extratores de características, utilizando conhecimento prévio dos dados de treinamento (ZHENG; YANG; TIAN, 2017; LIU et al., 2019b). No entanto, a estratégia exclusiva de pré-treinamento pode limitar a exploração completa da capacidade dos modelos profundos em descrever cenas de maneira adaptativa. O ajuste fino das CNNs pré-treinadas no conjunto de dados específico da cena alvo aprimora seu desempenho, reduzindo a disparidade de domínio entre conjuntos de dados (LIU et al., 2019b). A inicialização adequada dos pesos é crucial devido ao alto custo computacional de treinar modelos com muitos parâmetros ajustáveis (SUTSKEVER et al., 2013). Assim, o ajuste fino da CNN pré-treinada desempenha um papel significativo no treinamento eficiente (YOO et al., 2015; DIXIT et al., 2015; DURAND; THOME; CORD, 2016; LIU et al., 2014).

Uma prática comum para o ajuste fino de CNNs é o congelamento da última camada totalmente conectada (FC) do modelo pré-treinado, substituindo-a por uma nova camada FC com nodos equivalentes ao número de categorias no conjunto de dados de destino. Os parâmetros das camadas convolucionais (CONV) anteriores permanecem inalterados. Essa CNN modificada é ajustada por meio de treinamento usando o conjunto de dados alvo, onde a retropropagação é interrompida após a última camada FC, permitindo que essas camadas extraiam características discriminantes das camadas aprendidas anteriormente.

O descongelamento gradual de algumas camadas também pode ser adotado para melhorar ainda mais a qualidade do aprendizado, permitindo que camadas anteriores se adaptem a novas representações do conjunto de dados de destino. Outra abordagem é atribuir diferentes taxas de aprendizado a diferentes camadas da CNN, onde as primeiras camadas têm uma taxa de aprendizado muito baixa, enquanto as últimas camadas têm taxas mais altas. Essa estratégia busca ajustar suavemente as representações mais abstratas das primeiras camadas, comparadas às camadas FC especializadas (ZENG et al., 2021).

Khan et al. (2016) adotaram uma abordagem de aumento de dados, expandindo o conjunto de dados de imagens da cena por meio de versões invertidas, cortadas e rotacionadas, visando melhorar a robustez no aprendizado de representações. Liu et al. (2019b) argumentam que modelos profundos podem não se beneficiar significativamente do ajuste fino, podendo resultar em efeitos negativos nas camadas FC se os dados de treinamento não forem apropriados.

2.4.2.3 *Modelo Específico de CNN*

Além dos modelos convencionais de CNNs, que consistem em CNNs pré-treinadas e ajustadas, há um grupo adicional de modelos profundos projetados especificamente para a tarefa de categorização de cenas, introduzindo novas arquiteturas de rede (LIU et al., 2018; SUN et al., 2018).

Para capturar informações discriminantes de regiões de interesse, Zhou et al. (2016) substituíram as camadas totalmente conectadas (FC) em um modelo CNN por uma camada *Global Average Pooling* (GAP) (LIN; CHEN; YAN, 2013), seguida por uma camada Softmax. Essa abordagem, chamada de *Global Average Pooling Convolutional Neural Networks* (GAP-CNN), utiliza uma combinação da camada GAP original e a operação de convolução 1×1 para formar um mapa de ativação de categorias, permitindo que GAP-CNN se concentre em regiões específicas de categorias, melhorando a categorização de cenas. Vale ressaltar que, apesar de possuir menos parâmetros do que a camada FC (ZHOU et al., 2016; DU et al., 2019), GAP-CNN pode alcançar uma precisão de categorização similar.

Considerando que a esparsidade pode aprimorar a capacidade discriminativa das representações de características (SHIH; ENDRES; HOIEM, 2014; SUN; WANG; TANG, 2015), Liu et al. (2018) apresentaram o modelo *Dictionary Learning CNN* (DL-CNN), substituindo as camadas totalmente conectadas (FC) por novas camadas de aprendizado de dicionário. Essas camadas são compostas por um número finito de unidades recorrentes, correspondendo a processos de iteração no *Approximate Message Passing* (AMP) (DONOHO; MALEKI; MONTANARI, 2009). Os dicionários de aprendizagem são atualizados por meio de retropropagação em todas as camadas da rede.

Devido às limitações das camadas CONV em descrever explicitamente as relações contextuais entre diferentes regiões de uma cena, Sun et al. (2018) propuseram o *Contextual Features in Appearance* (CFA), baseado em *Long short-term memory* (LSTM) (GRAVES, 2012). Neste método, as camadas LSTM são empregadas para descrever dependências contextuais espaciais. A saída dos módulos LSTM é concatenada para representar as relações contextuais na aparência da cena. Essas inovações visam melhorar a capacidade do modelo em compreender as relações mais complexas entre diferentes partes de uma imagem de cena.

As CNNs preservam a estrutura espacial global da imagem, demonstrando que as características globais são sensíveis a variações geométricas (GONG et al., 2014; LI et al., 2019). Para alcançar invariância geométrica, Hayat et al. (2016) desenvolveram uma camada de espaço não

estruturado, embaralhando a posição das características originais ao trocar blocos de imagens diagonais adjacentes.

2.4.3 Representação de Cena Baseada em CNN

Os métodos propostos para representações eficazes de cenas geralmente são categorizados em cinco grupos: características globais da CNN, características invariantes ao espaço, características semânticas, características multicamadas e características combinadas (características de múltiplas visualizações).

2.4.3.1 Método Baseado em Características de CNN Global

Métodos que se baseiam nas características globais da CNN preveem diretamente as probabilidades das categorias em toda a imagem da cena. Essas características globais são frequentemente extraídas de modelos CNN genéricos, pré-treinados em conjuntos de dados extensos como o ImageNet (DENG et al., 2009) e o Places (ZHOU et al., 2014; ZHOU et al., 2017), ou ajustados para conjuntos de dados específicos. O modelo DL-CNN (LIU et al., 2018) foi desenvolvido para extrair características globais esparsas de imagens de cena.

O desempenho dessas características globais da CNN é influenciado pelo conteúdo presente nas imagens de entrada. CNNs têm a capacidade de extrair representações genéricas de características globais, especialmente quando treinadas em conjuntos de dados amplos. Notavelmente, modelos CNN centrados em cena têm se mostrado eficazes na geração de representações, incorporando correlações espaciais entre objetos e propriedades globais de uma cena (ZHOU et al., 2014; ZHOU et al., 2017; HERRANZ; JIANG; LI, 2016).

É importante destacar que o desempenho de um sistema de categorização de cena está intrinsecamente relacionado às entidades presentes nas cenas, conforme evidenciado por Herranz, Jiang e Li (2016). Fundos de imagens caóticos tendem a degradar o desempenho do modelo em comparação a casos em que um objeto está isolado do fundo ou quando a imagem apresenta um fundo liso.

2.4.3.2 *Características Invariantes ao Espaço*

Para mitigar os problemas ocasionados por operações sequenciais na CNN convencional, alguns estudos Gong et al. (2014), Guo et al. (2016), Cimpoi, Maji e Vedaldi (2015) propuseram abordagens fundamentadas em características espacialmente invariantes, visando manter a robustez espacial, ou seja, características cuja saída é resiliente a mudanças na geometria de uma imagem de entrada (LI et al., 2019). As características invariantes ao espaço são geralmente extraídas por meio de um processo com cinco etapas básicas:

1. Extração de *patch* local: a imagem de entrada é dividida em *patches* locais menores, que servem como entradas para uma CNN;
2. Extração de características locais: as características são extraídas nas camadas CONV ou FC do modelo;
3. Geração da codificação: Nesta etapa, um repertório de palavras visuais é gerado com base nas características de diferentes regiões da imagem. As palavras são tipicamente aprendidas de forma não supervisionada através de agrupamentos de palavras por similaridade;
4. Geração de características invariantes ao espaço: Com base no conjunto de codificações, os atributos são codificados em uma representação invariante ao espaço;
5. Previsão da categoria: O padrão de entrada é categorizado.

As características locais podem ser extraídas por mapas CONV semânticos de uma CNN convencional (YOO et al., 2015; XIONG; YUAN; WANG, 2020; GAO et al., 2015). Essa abordagem, utilizada por Yoo et al. (2015), substitui as camadas FC por CONV para gerar uma alta quantidade de características espaciais locais, resultando em menor tempo de computação em comparação com o processamento de vários *patches* espaciais. Outra estratégia, proposta por Gao et al. (2015), estratifica as ativações em pirâmides de vários níveis, codificando informações espaciais mais discriminantes.

A agregação de características fundamentada na informação de localidade tem como principal objetivo relacionar as características extraídas localmente à representação final utilizada para a categorização, o que impacta positivamente tanto a precisão quanto a eficiência dos algoritmos de classificação de cenas, conforme demonstrado em (LIU et al., 2019b). Além disso,

diversas técnicas consagradas, como *Improved Fisher Vector* (IFV) (SÁNCHEZ et al., 2013), *Vector of Locally Aggregated Descriptors* (VLAD) (SÁNCHEZ et al., 2013) e BoVW (JÉGOU et al., 2010), são amplamente empregadas em métodos baseados em aprendizado profundo, proporcionando uma extração mais robusta e discriminativa dos padrões presentes nos dados visuais.

Métodos baseados em características invariantes ao espaço são eficazes para garantir robustez geométrica. No entanto, o paradigma de janela deslizante utilizado por esses métodos requer digitalização multi-resolução com proporções fixas, tornando-se inadequado para objetos com tamanhos ou proporções variáveis na imagem da cena. Além disso, o uso de *patches* densos pode introduzir ruído na representação final, diminuindo a precisão da categorização. Portanto, extrair características semânticas de regiões da imagem da cena pode superar essas limitações.

2.4.3.3 Características Semânticas

Processar todos os *patches* da imagem de entrada é computacionalmente caro e propenso a redundâncias. A detecção de objetos, que determina a presença de instâncias de regiões salientes em uma imagem (LIU et al., 2020), é fundamental para métodos baseados em objetos, pois encontram regiões salientes que contêm informações contextuais da imagem.

Para identificar todas as regiões semânticas possíveis, Liu et al. (2014) propuseram uma busca seletiva combinando busca exaustiva com segmentação, seguida por uma *pyramid CNN* para gerar mapas de características de cada região. Já Cheng et al. (2018) utilizaram uma rede de detecção de regiões para encontrar regiões discriminantes. Ambos os métodos baseados em características semânticas Liu et al. (2014), Cheng et al. (2018) identificam múltiplas regiões semânticas.

Características semânticas, provenientes de *patches* em diferentes escalas, oferecem pistas complementares para detectar objetos grandes, pequenos ou partes deles. Assim, para melhorar o desempenho, é comum selecionar várias regiões de cada cena durante o treinamento do modelo. Durand, Thome e Cord (2016) introduziram uma camada Max para a seleção de regiões de interesse e uma camada Min para identificar regiões com mais evidências negativas. Métodos baseados em características semânticas atuam na detecção de objetos, sendo impactados negativamente pela baixa rotulagem dos *patches*, afetando a precisão do reconhecimento de cenas (DURAND; THOME; CORD, 2016). No entanto, esses métodos podem enfrentar desa-

fios como acumulação de erro e alto custo computacional (XIONG; YUAN; WANG, 2019), duas limitações significativas.

2.4.3.4 Característiscas Multi-camadas

A informação semântica nas características das camadas inferiores é comparativamente menor do que nas camadas superiores, embora a precisão na localização de objetos seja mais confiável nas camadas inferiores (LIN et al., 2017). Em resposta a essa disparidade, alguns métodos Liu et al. (2019a), Xie et al. (2015), Song, Jiang e Herranz (2017) propuseram abordagens hierárquicas, incorporando características de alta resolução das primeiras camadas com informações semânticas mais abundantes extraídas das camadas finais. Esse processo de formação de *patches* multicamadas inclui:

1. Extração de *patches* locais: Uma determinada imagem de entrada é dividida em *patches* locais menores, que são as entradas de uma CNN.

2. Extração de características locais: As características locais são extraídas nas camadas CONV ou FC do modelo.

3. Geração da codificação: Nesta etapa, é gerado um repertório de palavras visuais com base nas características de diferentes regiões da imagem. As palavras geralmente são aprendidas de forma não supervisionada.

4. Geração de características invariantes ao espaço: Com base no conjunto de codificação, os atributos são codificados em uma representação invariante ao espaço.

5. Previsão da categoria: O padrão de entrada é categorizado.

Embora a incorporação de características de diferentes camadas possa aprimorar a representação final, esse processo também apresenta o risco de *overfitting*, o que pode prejudicar o desempenho do modelo (YANG; RAMANAN, 2015). Como alternativa, diversos métodos Liu et al. (2019a), Yang e Ramanan (2015), Xie et al. (2015), Tang, Wang e Kwong (2017), Song, Jiang e Herranz (2017) optam por extrair características de um subconjunto específico de camadas, reduzindo a complexidade. Por exemplo, Xie et al. (2015) desenvolveram representações baseadas em dicionário utilizando *Convolution Fisher Vector* (CFV) e *Mid-level Local Discriminative Representation* (MLR) para categorizar cenas. (TANG; WANG; KWONG, 2017)

organizaram as camadas do GoogLeNet em três grupos, considerando também os mapas de características finais de cada bloco. Já Liu et al. (2019a) utilizam mapas de características independentes produzidos por cada bloco residual das Redes Residuais (ResNets). Além disso, esses métodos demonstram resultados promissores. Métodos baseados em características multicamadas aplicam técnicas de fusão, que podem ser categorizadas em duas classes distintas (DONG et al., 2014), dependendo do momento da fusão:

1. Fusão precoce: As características são fundidas após sua extração, seguidas da categorização de cenas;

2. Fusão tardia: A aprendizagem supervisionada é empregada para rotular cada categoria multicamadas, e, em seguida, ocorre a fusão em uma característica final.

O desempenho da fusão tardia demonstrou ser superior, mesmo considerando o maior custo em termos de complexidade e tempo de processamento. Esse custo adicional justifica o uso mais frequente da fusão precoce (LIU et al., 2019a; XIE et al., 2015; TANG; WANG; KWONG, 2017; SONG; JIANG; HERRANZ, 2017). Além disso, as operações de adição e regras do produto são comumente aplicadas para combinar várias características (TANG; WANG; KWONG, 2017). Dado que os espaços de características em diferentes camadas são distintos, a regra do produto é preferível à adição para a fusão de características, conforme demonstrado por experimentos empíricos em (TANG; WANG; KWONG, 2017).

2.4.3.5 Características Combinadas

Descrever uma cena complexa por meio de uma representação compacta de características é uma tarefa desafiadora, levando à busca por abordagens que integrem características geradas por CNNs complementares (SUN et al., 2018; WANG et al., 2017; WANG et al., 2017). A complementaridade das características provenientes de redes treinadas em diferentes conjuntos de dados é reconhecida, como evidenciado por (HERRANZ; JIANG; LI, 2016), que combinaram dados de CNNs centradas em objetos e em cenas, considerando a escala. Esta estratégia de combinação também foi adotada por (WANG et al., 2015), que empregaram uma CNN centrada em objetos para extrair informações específicas do objeto e uma CNN centrada na cena para capturar características globais.

Em uma perspectiva de aprendizado fracamente supervisionado, Wang et al. (2017) introduziram a *PatchNet*, que utiliza características contextuais de imagens como informação de supervisão para a extração de características em nível de *patch*. A abordagem *Scene-PatchNet* e *Object-PatchNet* foi implementada conjuntamente para aprimorar o reconhecimento. Em um contexto semelhante, Wang et al. (2017) propuseram uma arquitetura de Multi Resolution Convolutional Neural Networks (MR-CNN) para capturar conteúdo visual em várias escalas, empregando a *BN-Inception* (WANG et al., 2015) para características de baixa resolução e uma versão mais profunda para características de alta resolução. A exploração de características globais e espacialmente invariantes para explicar o layout de baixa resolução da cena foi realizada por (KIM; FRAHM, 2018).

Li et al. (2020) utilizaram a ResNet18 (HE et al., 2016) para gerar mapas de atenção, que, em conjunto com a imagem original, alimentam uma CNN. A combinação das características globais extraídas pela ResNet18 e as características do mapa de atenção resultante da rede transformadora de *patches* espaciais promove o refinamento das características, destacando partes mais discriminantes da imagem. Em uma abordagem multimodal, López-Cifuentes et al. (2020) propuseram uma arquitetura composta por uma ramificação profunda e uma semântica. A primeira captura características globais da CNN, enquanto a segunda extrai objetos significativos e suas relações a partir de super pixels. O processo de categorização de lugares desempenha um papel crucial em algoritmos de SLAM.

2.4.4 Visual Geometry Group - VGG_19

A arquitetura VGG_19 (SIMONYAN; ZISSERMAN, 2014) é uma CNN que se destaca pela sua profundidade e simplicidade, é um modelo que pode ser utilizado com pré-treinado e possui características globais. A VGG_19 é uma extensão da arquitetura VGG_16, adicionando três camadas convolucionais e duas camadas totalmente conectadas, totalizando 19 camadas.

Na arquitetura VGG_19, todas as camadas convolucionais têm filtros 3x3, e as camadas de pooling são realizadas com filtros 2x2. A profundidade da rede aumenta capacidade de aprendizagem de representações complexas. As camadas principais da VGG_19 (SIMONYAN; ZISSERMAN, 2014) são:

1. Camada convolucional (Conv): Filtros 3x3, *stride* 1, com *zero-padding*.

2. *Max Pooling Layer* (MaxPool): Filtros 2x2 e *stride* 2.

3. Totalmente Conectada: Camadas totalmente conectadas no final da rede.

Seus pontos fortes incluem uma estrutura uniforme e fácil interpretação, com camadas de convolução e *pooling* consistentes, usando filtros de 3x3 e pooling de 2x2. A simplicidade dessa abordagem facilita a implementação e a compreensão da rede, tornando-a uma escolha popular em visão computacional. Além disso, a VGG_19 demonstrou bom desempenho em tarefas gerais de classificação de imagens devido à sua profundidade, permitindo a aprendizagem de representações hierárquicas detalhadas.

Entretanto, a VGG_19 também apresenta algumas limitações. Sua profundidade considerável resulta em uma alta demanda computacional, tornando o treinamento e a inferência mais demorados e exigentes em recursos. A suscetibilidade ao *overfitting*, especialmente em conjuntos de dados pequenos, é outra desvantagem, pois a complexidade da rede pode levar a uma adaptação excessiva aos dados de treinamento.

A utilização de uma rede neural convolucional CNN pré-treinada, como a VGG_19, por meio de transferência de aprendizado, apresenta diversas vantagens em tarefas de visão computacional. A VGG_19 foi pré-treinada com a base de dados ImageNet, o que a capacita a aprender representações complexas e hierárquicas de características visuais em uma ampla gama de imagens.

Ao empregar uma VGG_19 pré-treinada, beneficia-se do conhecimento adquirido durante o treinamento em tarefas de categorização em grande escala. Isso resulta em características visuais generalizadas que são úteis para uma variedade de aplicações, mesmo quando o conjunto de dados alvo é relativamente pequeno. A transferência de aprendizado permite adaptar a VGG_19 a tarefas específicas com maior eficiência, pois o modelo já aprendeu a reconhecer padrões e características importantes.

Uma das vantagens chave é a aceleração do treinamento e a redução da necessidade de grandes conjuntos de dados para treinar do zero. Além disso, ao utilizar camadas convolucionais da VGG_19 como extratores de características, é possível personalizar a rede para tarefas específicas, economizando recursos computacionais. Esta abordagem é especialmente valiosa quando há limitações de dados ou recursos de computação. Referências relevantes incluem o trabalho original da VGG_19 e estudos que aplicam transferência de aprendizado em visão computacional (YOSINSKI et al., 2014; BANSAL et al., 2021).

2.5 DISCUSSÃO

O campo de pesquisa das CNNs para categorização de cenas é uma área que tem experimentado significativo desenvolvimento e aprimoramento de modelos. Diversos modelos, técnicas, frameworks e conjuntos de dados têm sido apresentados nesse domínio. O objetivo primordial das CNNs é criar uma representação discriminante para a fase de categorização. A maioria dos trabalhos em CNN se concentra na arquitetura da extração de características, sendo que poucos trabalhos propõem abordagens distintas para a fase de categorização. Os categorizadores, em sua maioria, consistem em uma camada totalmente conectada no final de uma rede neural profunda, seguida por uma camada Softmax, SVM ou k-vizinhos mais próximos (k-NN).

As CNNs apresentam diversas vantagens significativas em tarefas de visão computacional e processamento de imagens. Ao contrário de métodos *handcrafted*, as CNNs possuem a capacidade de criar hierarquias de características, aprendendo automaticamente as características mais discriminativas dos dados e adaptando-se a padrões específicos relevantes para a tarefa em questão. Além disso, as CNNs são proficientes em realizar transferência de aprendizado, permitindo a aplicação de conhecimento adquirido em uma tarefa para outras tarefas relacionadas. Outra vantagem é a capacidade das CNNs lidarem com imagens de diferentes tamanhos sem a necessidade de ajustes significativos, uma vez que a camada de convolução é capaz de extrair padrões independentemente da resolução da imagem, realizando esse processo de forma adaptativa.

Este comparativo evidencia que a escolha entre algoritmos *handcrafted* e CNNs depende da natureza específica da tarefa, da complexidade do problema e dos recursos disponíveis. Enquanto algoritmos *handcrafted* podem ser eficazes em contextos específicos, as CNNs oferecem vantagens significativas em tarefas mais complexas e diversificadas, proporcionando uma abordagem mais automatizada e adaptativa para a extração de características em imagens.

Os modelos pré-treinados de CNNs de propósito geral podem ser empregados na etapa de treinamento dos modelos para reconhecimento de lugares. Além disso, um categorizador de baixa complexidade computacional é essencial para operações online em um robô. O objetivo desta tese de doutorado é propor um novo categorizador capaz de lidar com vetores de alta dimensionalidade e utilizar aprendizagem incremental com baixo custo computacional. A arquitetura escolhida de CNN para extração de características é a VGG_19 (SIMONYAN; ZISSERMAN, 2014), uma estrutura amplamente reconhecida na literatura, conhecida por sua

simplicidade dentro do contexto das CNNs, facilitando comparações. A VGG_19 é selecionada por sua capacidade de extrair características gerais das cenas para a categorização de cenas. O próximo capítulo abordará modelos de categorização que utilizam distância por vetores de referência: a família k-NN, família *Learning Vector Quantization* (LVQ) e família Mapa Auto-Organizável (SOM). Esses modelos devem se aproximar dos requisitos para categorizadores que foram anteriormente mencionados.

3 CATEGORIZAÇÃO POR MODELOS QUE UTILIZAM A DISTÂNCIA POR VETORES DE REFERÊNCIA

Este capítulo revisa os modelos fundamentais que realizam o cálculo da distância por vetores de referência, estabelecendo uma base para comparações com a proposta desta tese e inspirando ideias para o modelo proposto. Os modelos descritos estão fundamentados nas famílias de métodos k-NN, LVQ, e alguns modelos de SOM. A seleção dos modelos é baseada na capacidade de realizarem o cálculo das distância por vetores de referência para particionar um determinado espaço de dados.

O k-NN (COVER; HART, 1967) é uma abordagem popular que categoriza um ponto com base na maioria de seus vizinhos mais próximos. Esta técnica simples, porém eficaz, é amplamente aplicada em diversos domínios. Na mesma família k-NN, o *Large Margin Nearest Neighbor* (LMNN) (DOMENICONI; GUNOPULOS; PENG, 2005) tem como objetivo aumentar a margem entre diferentes categorias, enquanto o Locally Adaptive Nearest Neighbors (LANN) (GÖPFERT; WERSING; HAMMER, 2022) introduz matrizes de relevância locais, cada uma específica para um ponto particular, indicando a relevância das características em sua vizinhança.

O LVQ (KOHONEN, 2012) é algoritmo de quantização vetorial e entre suas extensões, foram escolhidas versões como *Generalized Learning Vector Quantization* (GLVQ) (KOHONEN, 1995), *Generalized Relevance Learning Vector Quantization* (GRLVQ) (HAMMER; VILLMANN, 2002), *Generalized Matrix Learning Vector Quantization* (GMLVQ) (SCHNEIDER; BIEHL; HAMMER, 2009a), e *Local Generalized Matrix Learning Vector Quantization* (LGMLVQ) (SCHNEIDER; BIEHL; HAMMER, 2009a). Esses métodos se mostram úteis para diferentes tipos de dados, explorando relações complexas entre amostras e aprendendo representações eficazes para o agrupamento em conjuntos de dados multidimensionais.

3.1 MÉTODOS BASEADOS EM k-NN

O algoritmo k-NN, desenvolvido por (COVER; HART, 1967), é um método de aprendizado supervisionado não-paramétrico. Neste algoritmo, a entrada consiste nos k exemplos de treinamento mais próximos em um conjunto de dados, e a saída é associada à categoria mais comum entre esses vizinhos. Em termos simples, uma amostra é categorizada com base em uma votação da sua vizinhança mais próxima, sendo atribuída à categoria prevalente entre seus k vizinhos mais próximos, sendo k um inteiro positivo geralmente pequeno. Se $k = 1$, a

amostra é atribuída à categoria do único vizinho mais próximo.

Pode-se atribuir pesos às contribuições dos vizinhos, de modo que os vizinhos mais próximos tenham uma contribuição maior. Por exemplo, uma técnica comum é atribuir a cada vizinho um peso de $1/d$, onde d representa a distância euclidiana até o vizinho. Os d vizinhos são obtidos a partir de um conjunto de distâncias para os quais a categoria é previamente conhecida, constituindo o conjunto de treinamento do algoritmo. Notavelmente, o k-NN é sensível à estrutura local dos dados.

Os exemplos de treinamento são vetores em um espaço multidimensional, cada um associado a um rótulo da categoria. A fase de treinamento do algoritmo consiste apenas em armazenar os vetores de características para as instâncias de treinamento. Durante a fase de categorização das amostras de testes, o usuário define a constante k, e um vetor sem rótulo (uma consulta ou ponto de teste) é categorizado atribuindo o rótulo mais frequente entre as k amostras de treinamento mais próximas desse ponto de consulta.

A distância euclidiana é uma métrica comumente utilizada para variáveis contínuas, enquanto a distância de Hamming é aplicada para variáveis binárias. Em contextos específicos, como dados de microarranjo de expressão gênica, o k-NN tem sido empregado com coeficientes de correlação, como Pearson e Spearman, como métricas. A precisão da classificação do k-NN pode ser aprimorada ao aprender a métrica de distância com algoritmos especializados, como a análise de componentes de vizinhança mais próxima (DOMENICONI; GUNOPULOS; PENG, 2005; GÖPFERT; WERSING; HAMMER, 2022).

O k-NN apresenta vantagens como simplicidade de implementação, adaptação a diferentes tipos de dados e eficácia em tarefas de categorização em espaços de características bem definidos. Além disso, não requer treinamento explícito durante a fase de modelagem.

Entretanto, desafios incluem a categorização enviesada pela maioria em casos de distribuição desigual de categorias, onde a categoria mais frequente tende a dominar devido à sua maior presença entre os k vizinhos mais próximos. Estratégias para contornar isso incluem a ponderação da categorização com base na distância entre o ponto de teste e seus k vizinhos mais próximos (COOMANS; MASSART, 1982). Outras desvantagens do k-NN incluem sensibilidade a *outliers*, dependência na escolha adequada de k, e a necessidade de armazenar todos os dados de treinamento, levando a um alto consumo de memória e tempo de processamento em conjuntos de dados volumosos.

Para superar limitações e aprimorar o desempenho do k-NN, Domeniconi, Gunopulos e Peng (2005), Göpfert, Wersing e Hammer (2022) aplicaram o aprendizado de métrica super-

visionado. Esses algoritmos utilizam informações dos rótulos para aprender novas métricas ou pseudo-métricas. Em casos de dados de alta dimensão, a redução da dimensionalidade antes da aplicação do k-NN pode evitar os efeitos da maldição da dimensionalidade, na qual a distância euclidiana torna-se inadequada devido à equidistância quase uniforme entre os vetores de consulta em alta dimensionalidade (BEYER et al., 1999). Algoritmos que modificam a métrica de distância serão apresentados para aprimorar o desempenho do k-NN. O pseudocódigo relacionado ao k-NN é apresentado pelo Algoritmo 1.

Algoritmo 1 k-Nearest Neighbors (k-NN)

Inicializar o número de vizinhos k.
Calcular a distância entre $\mathbf{x}_{\text{teste}}$ e cada instância de treinamento $\mathbf{x}_i$.
Ordenar as instâncias de treinamento com base nas distâncias.
Selecionar os k vizinhos mais próximos.
Realizar a votação da maioria entre os rótulos dos vizinhos selecionados.
Atribuir a amostra à categoria vencedora.

Nas próximas seções, serão tratados modelos fundamentados no k-NN: o LMNN e o LANN. Esses modelos foram selecionados devido à sua ampla utilização na literatura e/ou por representarem abordagens mais contemporâneas em relação ao k-NN. Além disso, ambos os modelos incorporam ponderações de relevância em seus processos, ampliando as possibilidades de consideração do impacto de diferentes pontos no resultado final.

3.1.1 Large Margin Nearest Neighbor (LMNN)

O método LMNN (DOMENICONI; GUNOPULOS; PENG, 2005) é fundamentado no k-NN, incorporando a distância de Mahalanobis como métrica na configuração do k-NN. A abordagem da distância de Mahalanobis busca preservar a proximidade dos k vizinhos mais próximos pertencentes à mesma categoria, enquanto simultaneamente promove uma margem ampla entre instâncias de categorias distintas.

A regra dos k vizinhos mais próximos pressupõe um conjunto de dados de treinamento com instâncias rotuladas, onde as categorias são previamente conhecidas. Esse método categoriza uma nova instância de dados atribuindo a ela a categoria resultante da maioria dos votos das k instâncias de treinamento mais próximas. A proximidade é medida através de uma métrica predefinida. O LMNN, por sua vez, aprimora essa métrica de maneira supervisionada, visando aperfeiçoar a precisão da categorização na regra dos k vizinhos mais próximos (DOMENICONI; GUNOPULOS; PENG, 2005).

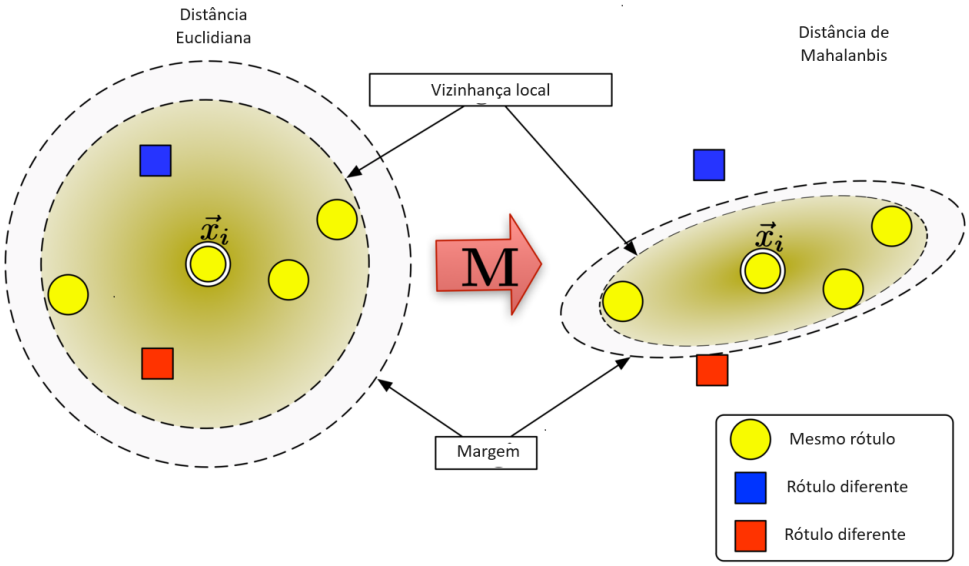

Figura 4 – Ilustração esquemática do LMNN.

O algoritmo dos vizinhos mais próximos de margem grande otimiza a matriz $\mathbf{M}$. O objetivo dessa otimização é duplo: para cada amostra, os vizinhos de destino devem estar próximos, enquanto os impostores (amostras de categorias diferentes) devem permanecer distantes. A Figura 4 ilustra o efeito dessa otimização em um exemplo representativo. A métrica aprendida pelo LMNN garante que o vetor de entrada $\mathbf{x}_i$ seja circundado por instâncias de treinamento pertencentes à mesma categoria. Se esse vetor representasse um ponto de teste, ele seria corretamente categorizado seguindo a regra do vizinho mais próximo. A Figura 4 apresenta uma visualização dessa distância aprendida no contexto do LMNN.

O objetivo do algoritmo é estudar uma transformação linear $L : R^d \rightarrow R^d$, que será usado para calcular as distâncias quadradas como na Equação 3.1.

$$D(\mathbf{x}_i, \mathbf{x}_j) = \|L(\mathbf{x}_i - \mathbf{x}_j)\|^2 \tag{3.1}$$

O algoritmo distingue entre dois tipos de amostras de dados:

1. Amostras corretas: Cada entrada $\mathbf{x}_i$ tem precisamente k vizinhos de destino diferentes com o rótulo de categorias idêntico y_i. Deseja-se que todos os k vizinhos-alvo tenham a distância mínima para $\mathbf{x}_i$, conforme calculado pela Equação 3.1. Todos os vizinhos de destino para um determinado dado de entrada $\mathbf{x}_i$ devem ser os vizinhos mais próximos sob a métrica aprendida. $\eta \in 0, 1$ é usado para determinar se a amostra $\mathbf{x}_j$ é um vizinho de destino da amostra de entrada $\mathbf{x}_i$.

2. Amostras incorretas: Um impostor de uma amostra de entrada $\mathbf{x}_i$ é outra amostra $\mathbf{x}_j$ com um rótulo de categoria diferente (isto é, $y_i \neq y_j$) que é um dos vizinhos adjacentes de x_i. Durante o período de aprendizado, o algoritmo tenta minimizar o número de impostores para todas as entradas de dados no conjunto de treinamento.

A função de custo associada ao LMNN com base nas métricas de distância expressas na Equação 3.1 é composta por dois termos concorrentes. O primeiro termo impõe uma penalidade para grandes distâncias entre cada entrada e seus vizinhos-alvo, enquanto o segundo termo impõe uma penalidade para pequenas lacunas entre cada entrada e todas as amostras incorretas. A expressão da função de custo é fornecida pela Equação 3.2.

$$\epsilon(L) = \sum_{ij} \eta_{ij} \left\| L(\mathbf{x}_i - \mathbf{x}_j) \right\|^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) \cdot [1 + \left\| L(\mathbf{x}_i - \mathbf{x}_j) \right\|^2 - \left\| L(\mathbf{x}_i - \mathbf{x}_l) \right\|^2] \quad (3.2)$$

Na Equação 3.2, o segundo termo representa a saturação, sendo a constante $c > 0$, geralmente definida por meio de validação cruzada. Para cada entrada $\mathbf{x}_i$, a saturação ocorre quando há amostras incorretas rotuladas de forma diferente dentro de uma unidade absoluta de distância, ou seja, a distância da entrada $\mathbf{x}_i$ para qualquer um de seus vizinhos-alvo. A função de custo, portanto, contribui para a busca de métricas de distância otimizadas, de modo que amostras com rótulos diferentes mantenham uma grande margem de distância e não invadam a vizinhança umas das outras. O pseudocódigo do LMNN é apresentado no Algoritmo 2. O fator de penalização do LMNN é uma medida que incentiva a aprendizagem de uma métrica de distância discriminativa. Ele ajusta a estrutura de vizinhança dos dados para maximizar a separação entre diferentes categorias, enquanto minimiza as variações dentro de cada categoria. Esse fator desempenha um papel crucial na otimização do algoritmo, pois controla o equilíbrio entre a maximização da margem de separação e a preservação da estrutura local dos dados.

3.1.2 Locally Adaptive Nearest Neighbors (LANN)

Alguns modelos neurais atribuem pesos diferentes às diversas dimensões dos vetores de entrada. Abordagens que buscam indicar intuitivamente a relevância de características para a categorização em determinado modelo podem ser alcançadas por meio da aprendizagem de métricas durante o treinamento. O algoritmo Locally Adaptive Nearest Neighbors (LANN)

Algoritmo 2 Large Margin Nearest Neighbor (LMNN)

Inicializar o número de vizinhos, k, o número de categorias C, fator de penalização ν e a matriz de transformação L como a matriz identidade.
while *Convergência não alcançada* **do**
    for *Cada par de pontos* $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ *pertencentes à mesma classe* **do**
        Calcular o gradiente da função de perda para o par.
        Atualizar a matriz de transformação L usando o gradiente e o fator de penalização ν.
Aplicar a matriz de transformação L aos dados de treinamento.
Calcular as distâncias entre x_{teste} e as instâncias de treinamento transformadas.
Ordenar as instâncias de treinamento com base nas distâncias.
Selecionar os k vizinhos mais próximos.
Realizar a votação da maioria entre os rótulos dos vizinhos selecionados.
Atribuir a amostra à categoria vencedora.

(GÖPFERT; WERSING; HAMMER, 2022) representa uma abordagem que possibilita a aprendizagem de métricas adaptativas localmente para o algoritmo k-NN. Essa técnica é uma extensão da ideia de que é benéfico adaptar a representação dos dados por meio da aprendizagem de uma métrica específica para o problema. A principal diferença entre a abordagem global e a local é que a técnica do LANN permite a aprendizagem de um esquema de ponderação individual para cada amostra do k-NN.

Essa técnica proporciona uma explicação local específica para uma decisão do modelo, permitindo regras de atualização online na forma de um gradiente estocástico. Além disso, a técnica do LANN oferece uma indicação de relevância para cada recurso em relação a cada ponto de dados individual, possibilitando uma compreensão local da importância das características.

As métricas aprendidas pelo LANN apresentam duas características importantes:

1. Diagonalidade: Cada peso aprendido corresponde diretamente a uma característica no espaço de entrada. Se o próprio espaço de entrada for interpretável, as métricas aprendidas também o serão.

2. Localidade: Para cada ponto no conjunto de treinamento, é identificada uma métrica local. Se as métricas aprendidas forem interpretáveis, elas fornecerão informações sobre como amostras individuais contribuem para uma predição.

3.2 MODELOS LVQ E SUAS VARIAÇÕES COM RELEVÂNCIAS

Os algoritmos LVQ estão relacionados a outros algoritmos de aprendizado competitivo, como SOM e k-NN. Esses algoritmos baseiam-se na regra de aprendizado "o vencedor leva tudo"e variantes nas quais apenas alguns elementos e suas vizinhanças têm pesos durante o aprendizado. Os algoritmos LVQ, tanto os originais quanto suas extensões modernas, utilizam aprendizado supervisionado para obter protótipos da categoria a partir de dados rotulados (NOVA; ESTÉVEZ, 2014).

Os modelos baseados em LVQ são intuitivos e fáceis de entender, pois se baseiam na ideia de representantes da categoria (protótipos) e particionamento do espaço de entrada em regiões. Essa abordagem é vantajosa em comparação com perceptrons multi-camadas ou SVM, que são considerados "caixas pretas", pois o uso de pro´totipos provêm uma interpretação da dinâmica do algoritmo. As fronteiras das regiões representam valores extremos dos conjuntos de dados, enquanto os protótipos LVQ são vetores típicos. Além disso, os algoritmos LVQ são simples e rápidos de aprender, baseando-se em modelos simples de aprendizagem hebbiana. O custo computacional dos algoritmos LVQ depende do número de protótipos, geralmente pré-definidos.

Os LVQs buscam aproximar a fronteira bayesiana teórica e podem lidar diretamente com problemas multi-categoria. Inicialmente, as regras de aprendizado do LVQ apresentavam sensibilidade à inicialização, problemas de convergência lenta e instabilidades. No entanto, duas abordagens principais foram propostas para definir funções de custo explícitas, baseadas nas regras de aprendizado por meio do gradiente descendente, solucionando assim o problema de convergência do algoritmo LVQ original.

Seja $\mathbf{X} = (\mathbf{x}_i, y_i) \subset R^D \times 1, .., C \mid i = 1, ..., N$ um conjunto de dados de treinamento, no qual $\mathbf{x} = [x_1 ... x_D]^T \in R^D$ são amostras de entrada, com cardinalidade $|\mathbf{X}| = N; y_i \in 1, ..., C \; i = 1, ..., N$ são os rótulos da amostra e C é o número de categorias. A rede neural consiste em uma série de protótipos, caracterizados por vetores $\mathbf{w}_i \in R^D$, para $i = 1, ..., D$, e seus rótulos $c(\mathbf{w}_i) \in 1, ..., C$ com $\mathbf{Y} = c(\mathbf{w}_j) \in 1, ..., C \mid j = 1, ..., M$. O esquema de categorização é baseado na unidade vencedora (BMU), onde o vencedor ganha tudo. O campo receptivo do protótipo $\mathbf{w}_i$ é definido pela Equação 3.3, na qual $d(\mathbf{w}, \mathbf{x})$ é uma medida de distância. O aprendizado visa determinar os vetores de peso (protótipos) de modo que as amostras de dados de treinamento sejam mapeadas para seus rótulos de categoria correspondentes.

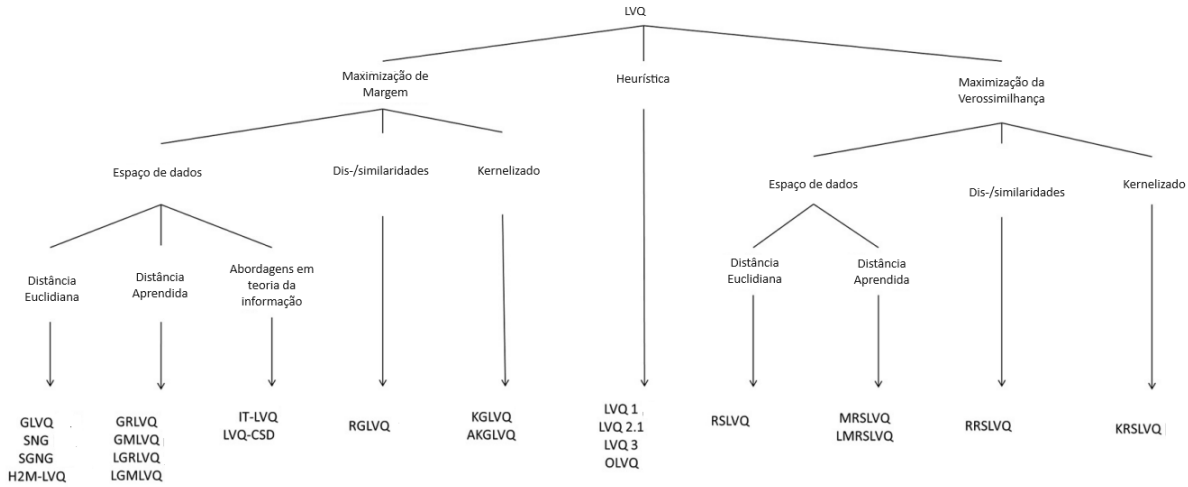

Figura 5 – Taxonomia dos mais relevantes classificadores LVQ desde o trabalho inicial de Teuvo Kohonen nos anos 80. Fonte: (NOVA; ESTÉVEZ, 2014).

$$R^i = \{\mathbf{x} \in \mathbf{X} \mid \forall\ \mathbf{w}_j (j \neq i) \rightarrow d(\mathbf{w}_i, \mathbf{x}) \ \leq \ d(\mathbf{w}_j, \mathbf{x})\} \tag{3.3}$$

A Figura 5 apresenta uma taxonomia dos classificadores LVQ, destacando os mais relevantes desenvolvidos desde o trabalho pioneiro de Kohonen. Os métodos LVQ são categorizados em três famílias principais: métodos LVQ de Kohonen (ramo central: heurística), métodos baseados em maximização de margem (ramo esquerdo) e métodos baseados na razão de maximização de verossimilhança (ramo direito). Todas essas versões do LVQ são fundamentadas na aprendizagem hebbiana, ou seja, são heurísticas (NOVA; ESTÉVEZ, 2014). A Tabela 1 complementa a Figura 5, apresentando diversas variações do LVQ juntamente com suas características distintas.

3.2.1 Learning Vector Quantization 1 (LVQ1)

O LVQ (KOHONEN, 2012; KOHONEN, 1990b) é uma abordagem amplamente utilizada para categorização. Esta técnica emprega uma arquitetura de rede semelhante à rede SOM, com a particularidade de que cada nodo de saída é associado a um membro de categoria, e nenhuma suposição é feita em relação à estrutura topológica. A rede LVQ está associada à rede de aprendizagem competitiva de duas camadas. Os algoritmos SOM e LVQ baseiam-se em protótipos, e seus processos de convergência consistem em distribuir os protótipos pelo espaço para representar com precisão as regiões das categorias.

Tabela 1 – Tabela com as características dos principais métodos da família LVQ (NOVA; ESTÉVEZ, 2014).

Nome	Características	Parâmetros	Distância	Restrições
LVQ	Heurística	$\{\epsilon, \omega, \mathbf{W}\}$	Distância euclidiana $d(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|$	$min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s$, no qual $s = \frac{1-w}{1+w}$
GLVQ	Maximização de margem	$\{\epsilon, \mathbf{W}\}$	Distância euclidiana $d(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|$	-
SRLVQ	Maximização de margem de verossimilhaça	$\{\epsilon, \sigma, \mathbf{W}\}$	Distância euclidiana $d(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|$	-
SNG	Maximização de margem	$\{\epsilon, \mathbf{W}\}$	Distância euclidiana $d(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|$	-
SGNG	Maximização de margem	$\{\epsilon, \mathbf{W}\}$	Distância euclidiana $d(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|$	-
H2MLVQ	Maximização de margem	$\{\epsilon, \mathbf{W}\}$	Distância mínima para harmônico	-
GRLVQ	Descida de gradiente de maximização de margem	$\{\epsilon, \mathbf{W}\}$	Distância adaptativa $d^{\lambda}(\mathbf{x}, \mathbf{w}) = \sum_i \lambda_i (x_i - w_i)^2$	$\sum_i \lambda_i = 1$
GMLVQ	Maximização de margem	$\{\epsilon, \Lambda, \mathbf{W}\}$	Distância adaptativa $d^{\Lambda}(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^T \Lambda (\mathbf{x} - \mathbf{w})$	
LGRLVQ	Descida de gradiente de maximização de margem	$\{\epsilon, \mathbf{W}\}$	Distância adaptativa $d^{\lambda}(\mathbf{x}, \mathbf{w}) = \sum_i \lambda_i (x_i - w_i)^2$	$\sum_i \lambda_i = 1$
LGMLVQ	Maximização de margem	$\{\epsilon, \mathbf{W}\}$	$d^{\Lambda}(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^T \Lambda (\mathbf{x} - \mathbf{w})$	Matrix $\Lambda = \Omega\Omega^T$ com $\sum_i \Lambda_{ii} = 1$
KRSLVQ	Maximização de margem de verossimilhaça kernelizada	$\{\epsilon, \sigma_k, \sigma, \gamma\}$	Distância Euclidiana no Espaço de características F $d(\Phi(\mathbf{x}), \mathbf{w}^{\mathbf{F}}) = \left\|\Phi(\mathbf{x}), \mathbf{w}^{\mathbf{F}}\right\|$	-

Algumas versões do LVQ foram propostas, como LVQ1, LVQ2, LVQ3 e o *Optimized-learning-rate Learning Vector Quantization* (OLVQ). O LVQ1 seleciona alguns protótipos iniciais e ajusta esses protótipos. Quando uma instância da base é categorizada com erro, o protótipo mais próximo é afastado, e quando é categorizada corretamente, o vencedor é aproximado. O nó vencedor é determinado pela Equação 3.4, na qual $\mathbf{w}_i$ representa cada um dos nós da rede e $\mathbf{w}$ é o peso do nó vencedor mais próximo da amostra.

$$\|\mathbf{x} - \mathbf{w}\| = \min_i \|\mathbf{x} - \mathbf{w}_i\| \tag{3.4}$$

O LVQ é uma abordagem baseada na categorização de vetores de características. Ele é utilizado para a quantização e categorização vetorial, além de ajustes finos do SOM. O LVQ busca minimizar a Função 3.5, onde $\mu_{kp} = 1$ se o neurônio k for o vencedor e $\mu_{kp} = 0$ caso contrário, quando o par padrão *p* é apresentado. O LVQ opera em um conjunto de *N* pares de padrões $(\mathbf{x}_p, \mathbf{y}_p)$, onde $\mathbf{x}_p \in R^J$ é o vetor de entrada e $\mathbf{y}_p \in R^J$ é o vetor de destino binário que codifica a categoria, ou seja, apenas uma saída de $\mathbf{y}_p$ tem valor unitário, enquanto todas as outras entradas são iguais a zero. (KOHONEN, 1990b) propôs uma família de algoritmos LVQ, incluindo LVQ1, LVQ2.1, LVQ3 e OLVQ. Considerando que o padrão $\mathbf{x}$ é apresentado no tempo n, o processo de ajuste de pesos do LVQ1 é dado pela Equação 3.6. Um esquema didático da regra de aprendizado de atualização do LVQ2.1 é ilustrado na Figura 6.

$$E = \frac{1}{N} \sum_{p=1}^{N} \sum_{k=1}^{K} \mu_{kp} \|\mathbf{x}_p - \mathbf{w}_k\| \tag{3.5}$$

O algoritmo LVQ1 pode demandar um número considerável de épocas para atingir a estabilização. Apesar dessa característica, para algumas das aplicações, o LVQ1 demonstra resultados satisfatórios, embora apresente uma certa demora e dificuldade de parametrização. Um dos parâmetros críticos é $\eta(n)$, uma função de ajuste que regula a aproximação ou afastamento dos protótipos. Esse comportamento é controlado pelo valor de $\eta(n)$, onde $0 < \eta(n) < 1$. É importante observar que $\eta(n)$ é uma função do tempo, geralmente modelada como uma exponencial decrescente, e o algoritmo encerra quando $\eta(n)$ se torna insignificante. Outro desafio do LVQ1 é a escolha adequada da quantidade inicial de protótipos, uma vez que essa quantidade permanece inalterada durante toda a execução do algoritmo. Em cenários de dados desbalanceados, é possível aplicar ajustes diferenciados para cada categoria ou escolher uma quantidade aproximada de protótipos para cada categoria. Com essas adaptações, o LVQ1 pode proporcionar resultados mais eficazes em bases desbalanceadas.

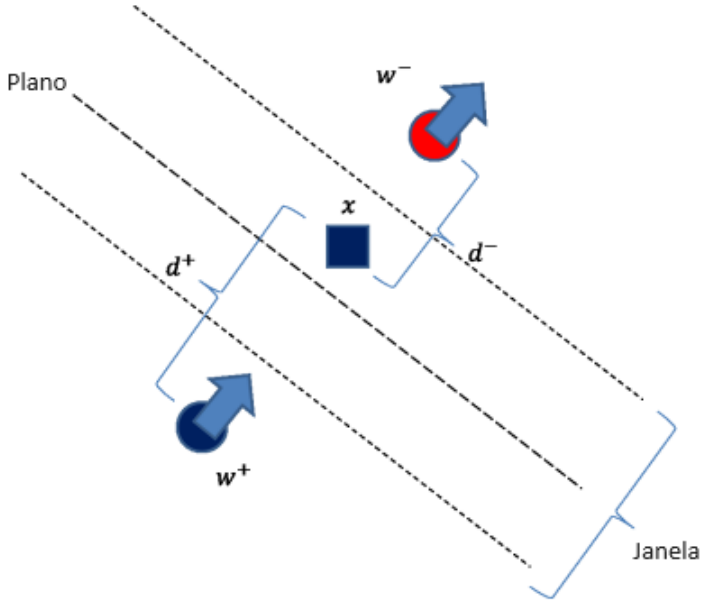

Figura 6 – Esquema da regra de aprendizado para o LVQ2.1. Os círculos representam os protótipos; o quadrado representa uma amostra de entrada; e as cores indicam rótulos de categorias diferentes. Neste exemplo, a amostra é categorizada incorretamente e o protótipo mais próximo da categoria correta se move em direção à amostra, enquanto o protótipo mais próximo da categotia incorreta se afasta da amostra (NOVA; ESTÉVEZ, 2014).

A expressão da função de energia, na Equação 3.5 , é formulada para medir a adequação dos protótipos às amostras de treinamento, buscando minimizar as distâncias entre os vetores de entrada e os protótipos associados às suas categorias. A função de energia fornece uma métrica objetiva para avaliar a eficácia do modelo em categorizar os padrões.

A atualização dos pesos no LVQ é realizada por meio de equação 3.6 que visa ajustar os protótipos de forma a melhor representar as características das categorias. A equação de atualização de pesos leva em consideração tanto a proximidade quanto a categorias associada ao protótipo vencedor, ajustando seus pesos de acordo. Essa abordagem visa otimizar a disposição dos protótipos no espaço de entrada, melhorando a capacidade do modelo LVQ em discriminar entre diferentes categorias de padrões.

$$\begin{aligned} \mathbf{w}(n) &= \mathbf{w}(n) + \eta(n)[\mathbf{x}(n) - \mathbf{w}(n)], \ y_{p,w} = 1 \\ \mathbf{w}(n) &= \mathbf{w}(n) - \eta(n)[\mathbf{x}(n) - \mathbf{w}(n)], \ y_{p,w} = 0 \end{aligned} \tag{3.6}$$

A variável w representa o índice do neurônio vencedor, enquanto $\mathbf{x}_t = \mathbf{x}_p$, $y_{p,w} = 1$ e 0 denotam, respectivamente, os casos de categorias corretas e incorretas para $\mathbf{x}_p$. A função de ajuste $\eta(n)$ é definida conforme as formulações anteriores. Quando utilizado para ajustar o LVQ1, é comum começar com um $\eta(0)$ pequeno, geralmente inferior a $0,1$. Ao final de cada época, a taxa de aprendizado decai de acordo com a Equação 3.7, onde n representa a época

de treinamento e n_{max} é o número máximo de épocas. O Algoritmo LVQ1 é apresentado na Equação 3.

$$\eta(n) = \eta(0)\left(1 - \frac{n}{n_{max}}\right) \tag{3.7}$$

Algoritmo 3 Algoritmo LVQ1

Inicializar $\mathbf{w}_k(0)$ e $\eta(o)$.
for $n \leftarrow 0$ ***to*** n_{max} **do**
    Apresente o padrão $\mathbf{x}$.
    Selecione o nodo vencedor de $\mathbf{x}$ através da Equação 3.4.
    Atualize os protótipos para todos os neurônios através da Equação 3.6.
    Decresça $\eta(n)$.
    Setar $n = n + 1$.
Atribuir as amostras às categorias vencedoras.

3.2.2 Learning Vector Quantization 2.1 (LVQ2.1)

Kohonen (1995) propôs versões aprimoradas do LVQ, incluindo o LVQ2.1. Nesta versão, a atualização ocorre nos dois protótipos mais próximos, desde que determinadas condições sejam atendidas. A principal ideia por trás do LVQ2.1 é realizar ajustes apenas nos protótipos próximos às regiões de indecisão. Para evitar a divergência entre esses protótipos, foi introduzida a Regra da Janela, a qual define se um elemento está na janela, ou seja, na região de indecisão.

O processo de categorização no LVQ2.1 é idêntico ao do LVQ1. Durante o aprendizado, no entanto, dois vetores de pesos, $\mathbf{w}_i$ e $\mathbf{w}_j$, que representam os vizinhos mais próximos para $\mathbf{x}$, são atualizados simultaneamente. Um desses vizinhos deve pertencer à categoria correta, enquanto o outro pertence à categoria incorreta. Além disso, $\mathbf{x}$ deve estar dentro de uma zona de valores chamada janela, definida em torno do plano médio de $\mathbf{w}_i$ e $\mathbf{w}_j$. Se d_i e d_j representam as distâncias euclidianas de $\mathbf{x}$ para $\mathbf{w}_i$ e $\mathbf{w}_j$, respectivamente, então $\mathbf{x}$ é considerado dentro da janela com largura relativa w, conforme definido pela Equação 3.8.

$$\begin{gathered} min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, no\ qual \\ s = \frac{1-l}{1+l} \end{gathered} \tag{3.8}$$

Recomenda-se uma largura relativa de janela, $\mathbf{w}$, entre 0,2 e 0,3. A versão aprimorada do algoritmo LVQ2.1, conhecida como LVQ2, representa uma melhoria em relação ao LVQ2.1

original, permitindo que $\mathbf{w}_j$ seja o vetor de pesos mais próximo de $\mathbf{x}$, enquanto no LVQ2.1 original, $\mathbf{w}_i$ deveria ser o mais próximo. A atualização dos pesos para o LVQ2.1 é expressa pela Equação 3.9, na qual $\mathbf{w}_i$ e $\mathbf{w}_j$ são os dois vetores de pesos mais próximos de $\mathbf{w}$, e $\mathbf{x}$ e $\mathbf{w}_j$ pertencem à mesma categoria. No entanto, $\mathbf{x}$ e $\mathbf{w}_i$ pertencem a categorias diferentes. Além disso, $\mathbf{x}$ deve estar dentro da janela l.

$$\begin{aligned} \mathbf{w}_i(n) &= \mathbf{w}_i(n) - \eta(n)[\mathbf{x}(n) - \mathbf{w}_i(n)], para\ y_{p,w} = 1 \\ \mathbf{w}_j(n) &= \mathbf{w}_j(n) + \eta(n)[\mathbf{x}(n) - \mathbf{w}_j(n)], para\ y_{p,w} = 0 \end{aligned} \tag{3.9}$$

3.2.3 Learning Vector Quantization 3 (LVQ3)

A segunda melhoria proposta por Kohonen (1995) foi o LVQ3. Este método busca evitar o sobreajuste do LVQ2.1 atuando mesmo quando o elemento já está sendo categorizado corretamente pelos dois protótipos mais próximos, ambos da instância utilizada para o ajuste. Além disso, a terceira versão do LVQ, o LVQ3, introduz um fator de estabilização ϵ.

O algoritmo LVQ2 foi baseado na ideia de deslocar os protótipos afetando as fronteiras de decisão em direção aos limites. Portanto, foram necessárias correções para garantir que $\mathbf{w}_i$ continue aproximando as distribuições de categoria, dentro dos limites da fronteira, pelo menos aproximadamente. Na Equação 3.10, $\mathbf{w}_i$ e $\mathbf{w}_j$ são os dois vetores de pesos mais próximos de $\mathbf{x}$, onde $\mathbf{x}$ e $\mathbf{w}_j$ pertencem à mesma categoria, enquanto $\mathbf{x}$ e $\mathbf{w}_i$ pertencem a categorias diferentes. Além disso, $\mathbf{x}$ deve estar dentro da janela w, para $k \in i, j$, se $\mathbf{x}$, $\mathbf{w}_i$ e $\mathbf{w}_j$ pertencem à mesma categoria. Recomenda-se valores de ϵ entre 0,1 e 0,5, em relação a $w = 0,2$ ou $0,3$. O valor ótimo de ϵ parece depender do tamanho da janela, sendo menor para janelas mais estreitas. Este algoritmo tende a se autoestabilizar, ou seja, o posicionamento ideal do $\mathbf{w}_i$ não muda durante o aprendizado.

$$\begin{aligned} \mathbf{w}_i(n) &= \mathbf{w}_i(n) - \epsilon\eta(n)[\mathbf{x}(n) - \mathbf{w}_i(n)] \\ \mathbf{w}_j(n) &= \mathbf{w}_j(n) + \epsilon\eta(n)[\mathbf{x}(n) - \mathbf{w}_j(n)] \end{aligned} \tag{3.10}$$

O fator de estabilização, ϵ, é utilizado para suavizar o ajuste quando os protótipos já estão categorizados corretamente por $\mathbf{x}$. Como as outras versões do LVQ, o LVQ3 é robusto a inicialições ruins dos protótipos, mas seu maior desafio reside em determinar valores adequados para parâmetros como η e ϵ. No entanto, há uma dificuldade adicional em saber quando os

protótipos foram suficientemente ajustados.

3.2.4 Optimized Learning Rate Learning Vector Quantization (OLVQ)

O OLVQ, proposto por Kohonen (1990a), representa uma versão otimizada do LVQ, desenvolvida para aumentar a velocidade de convergência do LVQ1. Neste modelo, cada protótipo possui taxas de aprendizagem individuais. Essas taxas de aprendizagem aumentam quando o protótipo é categorizado corretamente e diminuem caso contrário. A variação do valor de $\eta(n)$ é governada pela Equação 3.11. Essa adaptação na taxa de aprendizagem contribui para uma convergência mais rápida do OLVQ, tornando-o uma opção mais eficiente em termos de desempenho, enquanto mantém as características fundamentais do LVQ1.

A principal inovação introduzida pelo OLVQ reside na inclusão de um otimizador, que permite ajustar os protótipos de maneira mais eficiente e adaptativa durante o treinamento. Essa abordagem pode resultar em um desempenho aprimorado em tarefas de categorização quando comparada ao LVQ tradicional. A capacidade de adaptação das taxas de aprendizagem individualizadas, combinada com o otimizador, confere ao OLVQ uma vantagem significativa no processo de treinamento, oferecendo uma abordagem mais eficaz para a convergência do modelo.

$$
\begin{gathered}
\eta(n) = \frac{\eta(n-1)}{1+s(n)\cdot\eta(n-1)} \\
s(n) = +1, \text{se } \mathbf{x} \quad \text{for} \quad \text{categorizado} \quad \text{corretamente} \\
s(n) = -1, \text{se } \mathbf{x} \quad \text{for} \quad \text{categorizado} \quad \text{incorretamente} \\
0 < \eta(n) < 1
\end{gathered}
\tag{3.11}
$$

3.2.5 Generalized Learning Vector Quantization (GLVQ)

Os primeiros algoritmos baseados em LVQ enfrentaram desafios significativos, incluindo sensibilidade à inicialização, problemas na taxa de convergência e instabilidades no processo de aprendizagem. O Generalized Learning Vector Quantization (GLVQ), proposto por Sato e Yamada (1995), foi concebido para abordar essas limitações. No GLVQ, uma função de custo é introduzida com o objetivo de ser minimizada por meio de um gradiente estocástico descendente.

O GLVQ representa uma modificação no algoritmo LVQ tradicional, visando uma convergência mais rápida e melhorias de desempenho. Essas melhorias ampliaram a aplicabilidade dos algoritmos LVQ em diversas áreas. No GLVQ, a função de custo é formulada de maneira a modelar uma regra de aprendizado que é otimizada por meio do método de descida mais íngreme. Essa função de custo está associada à minimização de erros e à maximização da margem do categorizador.

No GLVQ, a minimização da função custo é realizada através de um gradiente estocástico descendente, onde a função $sgd(x) = (1 + \exp(-x))^{-1}$ representa uma sigmoide logística e $\mu(\mathbf{x}_i)$ é definida pela Equação 3.13. Essa abordagem visa aprimorar a capacidade do modelo em aprender de maneira mais eficiente, contribuindo para uma convergência mais rápida e estável.

$$E_{GLVQ} = \frac{1}{2}\sum_{i=1}^{N}\phi(\mu(\mathbf{x}_i)) \tag{3.12}$$

$$\mu(\mathbf{x}_i) = \frac{d^+(\mathbf{x}_i) - d^-(\mathbf{x}_i)}{d^+(\mathbf{x}_i) + d^-(\mathbf{x}_i)} \tag{3.13}$$

As distâncias $d^+(\mathbf{x}_i)$ e $d^-(\mathbf{x}_i)$ representam as distâncias entre a amostra em análise e os protótipos mais próximos corretos e incorretos, respectivamente. No contexto do algoritmo, essas distâncias ponderadas são utilizadas como fatores de ponderação para ajustar os movimentos nos protótipos, visando otimizar esse ajuste. Na Equação 3.13, quando $\mu(\mathbf{x}_i)$ é negativo, a amostra é categorizada corretamente, enquanto, se positivo, a amostra é categorizada incorretamente. A variável $\mu(\mathbf{x}_i)$ pode assumir valores entre -1 e 1, sendo $d^+(\mathbf{x}_i)$ e $d^-(\mathbf{x}_i)$ as distâncias euclidianas quadráticas da amostra para os protótipos da mesma categoria $\mathbf{w}^+$ e de uma categoria oposta $\mathbf{w}^-$, respectivamente.

A função de custo, expressa pela Equação 3.12, é formulada com o objetivo de ser minimizada durante o treinamento do algoritmo, que ocorre ao longo de n épocas ou até que a convergência seja alcançada. Durante cada época de treinamento, uma amostra $\mathbf{x}_i$ é aleatoriamente selecionada, e os protótipos mais próximos de sua categoria $\mathbf{w}^+$ e de uma categoria diferente $\mathbf{w}^-$ são identificados usando as distâncias euclidianas quadráticas $d^+(\mathbf{x}_i) = (\mathbf{x}_i - \mathbf{w}^+)^2$ e $d^-(\mathbf{x}_i) = (\mathbf{x}_i - \mathbf{w}^-)^2$.

Os protótipos $\mathbf{w}^+$ e $\mathbf{w}^-$ são atualizados pelas equações 3.14, onde ε^+ e ε^- são as taxas de aprendizagem positiva e negativa, respectivamente, e ξ^+ e ξ^- são as distâncias derivativas (Equação 3.15). As distâncias derivativas representam uma medida da diferença entre o vetor

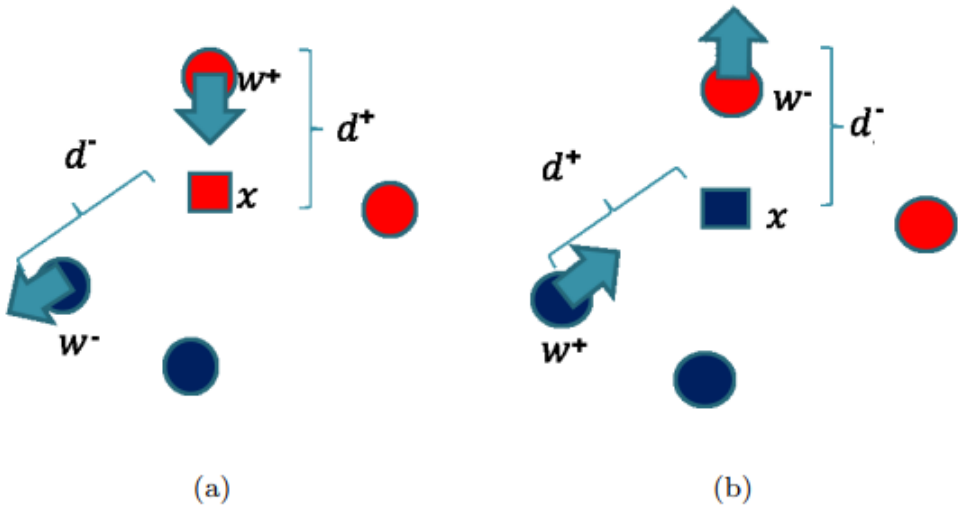

Figura 7 – Esquema da regra de aprendizado GLVQ que analisa o comportamento dos dois protótipos pertencentes a diferentes categorias. Os círculos representam os protótipos; o quadrado representa a amostra de entrada; e as cores indicam os diferentes rótulos de categorias (regra aprendizado quando uma amostra de entrada com um rótulo de categoria vermelha é apresentada). (b) Regra de aprendizado quando uma amostra de entrada com um rótulo de categoria azul é apresentada. (NOVA; ESTÉVEZ, 2014).

de entrada e um protótipo da categoria, levando em consideração uma matriz de pesos que pode ser ajustada durante o treinamento. Essa matriz de pesos é utilizada para transformar o espaço de características, tornando as comparações entre vetores mais discriminativas. Uma representação esquemática da regra de aprendizagem GLVQ é ilustrada na Figura 7.

$$\begin{aligned} \mathbf{w}^{+}(n+1) &= \mathbf{w}^{+}(n) + 2 \cdot \varepsilon^{+} \cdot \phi'(\mu(\mathbf{x}_i)) \cdot \xi^{+} \cdot (\mathbf{x}_i - \mathbf{w}^{+}) \\ \mathbf{w}^{-}(n+1) &= \mathbf{w}^{-}(n) - 2 \cdot \varepsilon^{-} \cdot \phi'(\mu(\mathbf{x}_i)) \cdot \xi^{-} \cdot (\mathbf{x}_i - \mathbf{w}^{-}) \end{aligned} \tag{3.14}$$

$$\begin{aligned} \xi^{+} &= \frac{2 \times d^{-}(\mathbf{x}_i,)}{(d^{-}(\mathbf{x}_i) + d^{+}(\mathbf{x}_i))^2} \\ \xi^{-} &= \frac{2 \times d^{+}(\mathbf{x}_i)}{(d^{-}(\mathbf{x}_i) + d^{+}(\mathbf{x}_i))^2} \end{aligned} \tag{3.15}$$

Ao término de cada época, as taxas de aprendizagem positiva e negativa passam por um decaimento, conforme a Equação 3.16, com o intuito de estabilizar o processo de aprendizagem dos protótipos. O procedimento detalhado do algoritmo GLVQ é descrito no Algoritmo 4, onde τ representa a taxa de decaimento do ϵ.

$$\varepsilon(t) = \frac{\varepsilon(0)}{1 + \tau(t - t_0)} \tag{3.16}$$

Algoritmo 4 Algoritmo GLVQ

Inicialize os parâmetros: ε^+, ε^-, τ, n, m.
Inicialize os protótipos: $\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2...\mathbf{w}_m]^T$ com posições aleatórias e categorias alternadas.
for *j* ← *número de épocas n* **do**
  for *i* ← *tamanho do conjunto de dados* **do**
    Selecione uma amosta aleatória $\mathbf{x}_i$ do conjunto de dados.
    Encontre os protótipos mais próximos da mesma categoria $\mathbf{w}^+$ e da categoria distinta $\mathbf{w}^-$.
    Atualize os protótipos com a Equação 3.14.
    $\mathbf{w}^+ = \mathbf{w}^+ + \Delta\mathbf{w}^+$.
    $\mathbf{w}^- = \mathbf{w}^- + \Delta\mathbf{w}^-$.
  Realize o decaimento das taxas de aprendizagem de acordo com a Equação 3.16.
  $\varepsilon^+ = \varepsilon^+ \times \tau$.
  $\varepsilon^- = \varepsilon^- \times \tau$.
Atribuir as amostras às categorias vencedoras.

3.2.6 Generalized Relevance Learning Vector Quantization (GRLVQ)

O GRLVQ (HAMMER; VILLMANN, 2002) representa uma modificação amplamente utilizada em comparação com o GLVQ. A principal distinção entre o GRLVQ e o GLVQ é a incorporação de uma métrica adaptativa durante o treinamento do categorizador, utilizando informações provenientes do vetor de relevâncias. Semelhante ao *Relevance Learning Vector Quantization* (RLVQ), o algoritmo GRLVQ emprega a distância euclidiana ponderada (Equação 3.18), onde λ_j representa as relevâncias para as diversas dimensões, a fim de calcular os campos receptivos dos protótipos (Equação 3.17).

$$R_\lambda^i = \{\mathbf{x} \in \mathbf{X}\ \forall \mathbf{w}_j (j \neq i)\} \rightarrow \|\mathbf{x} - \mathbf{w}_i\|_\lambda < \|\mathbf{x} - \mathbf{w}_j\|_\lambda \tag{3.17}$$

A adaptação da métrica de distância é alcançada mediante a introdução do vetor de relevância no cálculo da distância. Assim, obtém-se uma distância ponderada pela relevância de cada dimensão. Essa modificação confere ao método a capacidade de lidar eficazmente com conjuntos de dados de alta dimensionalidade. Isso ocorre porque as dimensões irrelevantes e/ou ruidosas têm sua influência significativamente reduzida durante o treinamento do algoritmo, graças ao uso do vetor de relevâncias. A equação da distância no GRLVQ é expressa como uma variação ponderada da distância euclidiana padrão. Seja $d^\lambda(\mathbf{x}, \mathbf{w})$ a medida de dissimilaridade entre o vetor de entrada $\mathbf{x}$ e um protótipo $\mathbf{w}$, as relevâncias λ_j normalizadas são incorporadas no cálculo da distância. A contrapartida ponderada da distância euclidiana é formalizada

Equação 3.18.

$$d^{\lambda}(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|_{\lambda}^{2} = \sum_{i=1}^{D} \lambda_j (x_j - w_j)^2 \tag{3.18}$$

Assim como no algoritmo GLVQ, os vetores $\mathbf{w}^+$ e $\mathbf{w}^-$ no GRLVQ representam o protótipo mais próximo correto e o mais próximo incorreto. Logo, o índice de distância relativa $\mu_{\lambda}(\mathbf{x}_i, \mathbf{W}, \lambda)$ para este algoritmo pode ser descrita pela Equação 3.20, no qual $d_{\lambda q}^{+}$ e $d_{\lambda q}^{-}$ são distâncias ponderadas quadráticas de $\mathbf{x}$ para $\mathbf{w}^+$ e $\mathbf{w}^-$. A função custo do GRLVQ a ser minimizada (Equação 3.19), considera o acúmulo da distância relativa $\mathbf{x}$ de todos os padrões de entrada de treinamento. Enquanto o GLVQ também possui sua versão linear, a função de transformação f sobre este índice é exclusivamente não-linear para o algoritmo GRLVQ. A função para f é a função logística $sgd(x) = (1 + exp(-x))^{-1}$.

A função objetivo do GRLVQ é semelhante à apresentada no GLVQ, no entanto, utiliza uma função de distância com pesos λ em suas Equações 3.19 e 3.20.

$$E_{GRLVQ} = \frac{1}{2} \sum_{i=1}^{N} \phi(\mu_{\lambda}(\mathbf{x}_i)) \tag{3.19}$$

$$\mu_{\lambda}(\mathbf{x}_i, \mathbf{W}, \lambda) = \frac{d^{\lambda}(\mathbf{x}_i, \mathbf{w}^+) - d^{\lambda}(\mathbf{x}_i, \mathbf{w}^-)}{d^{\lambda}(\mathbf{x}_i, \mathbf{w}^+) + d^{\lambda}(\mathbf{x}_i, \mathbf{w}^-)} \tag{3.20}$$

No método GRLVQ, a medida de dissimilaridade $d(\mathbf{x}_i, \mathbf{w})$ deve ser diferenciável em relação ao segundo argumento. Um exemplo importante é a contrapartida ponderada da distância euclidiana padrão (Equação 3.18) com relevâncias $\lambda_i > 0$ normalizadas tal que satisfaça a restrição imposta pela Equação 3.21. Após o cálculo da Equação 3.23, o vetor é ponderado pelas restrições $\lambda \geq 0$ e da Equação 3.21.

$$\sum_{i=1}^{D} \lambda_m = 1 \tag{3.21}$$

A cada novo padrão $\mathbf{x}_i$ apresentado, são atualizados o protótipo mais próximo de mesma categoria $\mathbf{w}^+$ e o protótipo mais próximo de outra categoria $\mathbf{w}^-$ segundo a Equação 3.22 e o vetor de relevâncias segundo a Equação 3.23 no qual λ_m é o vetor de relevâncias, ε^-, ε^+ e ε^{λ} são respectivamente, as taxas de aprendizagem positiva, negativa e do vetor de relevâncias.

$$\begin{aligned} \mathbf{w}^+(n+1) &= \mathbf{w}^+(n) + 2 \cdot \varepsilon^+ \cdot \phi'(\mu(\mathbf{x}_i)) \cdot \xi_{\lambda}^{+} \cdot (\mathbf{x}_i - \mathbf{w}^+) \\ \mathbf{w}^-(n+1) &= \mathbf{w}^-(n) - 2 \cdot \varepsilon^- \cdot \phi'(\mu(\mathbf{x}_i)) \cdot \xi_{\lambda}^{-} \cdot (\mathbf{x}_i - \mathbf{w}^-) \end{aligned} \tag{3.22}$$

$$\lambda_m(n+1) = \lambda_m(n) - \varepsilon\phi' \left(\xi^+(\mathbf{x}_m - \mathbf{w}_m^+)^2 - \xi^-(\mathbf{x}_m - \mathbf{w}_m^-)^2\right) \tag{3.23}$$

As distâncias derivativas no algoritmo GRLVQ auxiliam no ajuste dos pesos dos nodos, destacando características mais relevantes para a tarefa de categorização. Esses pesos são calculados por meio da Equação 3.24.

$$\begin{aligned} \xi_\lambda^+ &= \frac{2d^\lambda(\mathbf{x}_i,\mathbf{w}^-)}{(d^\lambda(\mathbf{x}_i,\mathbf{w}^+)+d^\lambda(\mathbf{x}_i,\mathbf{w}^-))^2} \\ \xi_\lambda^- &= \frac{2d^\lambda(\mathbf{x}_i,\mathbf{w}^+)}{(d^\lambda(\mathbf{x}_i,\mathbf{w}^+)+d^\lambda(\mathbf{x}_i,\mathbf{w}^-))^2} \end{aligned} \tag{3.24}$$

Algoritmo 5 Algoritmo GRLVQ

Inicialize os parâmetros: ε^+, ε^-, ε^λ, τ, n, m.
Inicialize os protótipos: $\mathbf{W} = \mathbf{w}_1, , \mathbf{w}_2, ..., \mathbf{w}_m$ aleatoriamente.
for *j* ← *1 número de épocas N* **do**
  for *i* ← *1 tamanho do conjunto de dados* **do**
    Selecione uma amosta aleatória $\mathbf{x}_i$ do conjunto de dados.
    Encontre os protótipos mais próximos da mesma categoria $\mathbf{w}^+$ e da categoria diferente $\mathbf{w}^-$ através da Equação 3.18.
    Atualize os protótipos de acordo com a Equação 3.22.
    $\mathbf{w}^+ = \mathbf{w}^+ + \Delta\mathbf{w}^+$.
    $mathbfw^- = mathbfw^- + \Delta mathbfw^-$.
    Atualize o vetor de relevâncias de acordo com a Equação 3.23.
    $\lambda = \lambda + \Delta\lambda$.
  Realize o decaimento das taxas de aprendizagem de acordo com a Equação 3.16.
Atribuir as amostras às categorias vencedoras.

3.2.7 Generalized Matrix Learning Vector Quantization (GMLVQ) e Local Generalized Matrix Learning Vector Quantization (LGMLVQ)

O GMLVQ (SCHNEIDER; BIEHL; HAMMER, 2009a; SCHNEIDER; BIEHL; HAMMER, 2009b) representa uma modificação do GLVQ que incorpora uma matriz de relevâncias para ponderar os pesos das dimensões, assemelhando-se à distância de Mahalanobis. Diferentemente do GRLVQ, que utiliza apenas a diagonal da matriz de co-ocorrência, o GMLVQ incorpora todos os índices dessa matriz. A distância empregada pelo GMLVQ é descrita pela Equação 3.25, no qual Λ é a matriz de pesos. O campo receptivo para esse caso é expresso pela Equação 3.26.

$$d^\Lambda(\mathbf{x},\mathbf{w}) = \|\mathbf{x}-\mathbf{w}\|_\Lambda^2 = \sum_i (\mathbf{x}_i - \mathbf{w}_i)^T \Lambda (\mathbf{x}_i - \mathbf{w}_i) \tag{3.25}$$

$$R_{\Lambda}^{i} = \{\mathbf{x} \in \mathbf{X}\ \forall \mathbf{w}_j (j \neq i)\} \rightarrow \|\mathbf{x} - \mathbf{w}_i\|_{\Lambda} < \|\mathbf{x} - \mathbf{w}_j\|_{\Lambda} \tag{3.26}$$

De maneira análoga, a expressão para a minimização de energia no GMLVQ pode ser representada pela Equação 3.27.

$$E_{GMLVQ} = \frac{1}{2} \sum_{i=1}^{N} \phi(\mu_{\Lambda}(\mathbf{x}_i)) \tag{3.27}$$

A Equação 3.28 calcula a distância relativa, levando em consideração a distância ponderada pela matriz de relevâncias. Essa distância relativa é uma ponderação entre a distância até o protótipo mais próximo correto e o protótipo mais próximo incorreto.

$$\mu_{\Lambda}(\mathbf{x}_i, \mathbf{W}, \Lambda) = \frac{d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{+}) - d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{-})}{d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{+}) + d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{-})} \tag{3.28}$$

Os pesos dos nodos são atualizados levando em consideração a Equação 3.29, no qual ε^{+} e ε^{+} são as taxas de aprendizagem positiva e negativa, ξ_{Λ}^{+} e ξ_{Λ}^{-} são as distâncias derivaticas positiva e negativa, Λ é a matrix de pesos.

$$\begin{aligned} \mathbf{w}^{+}(n+1) &= \mathbf{w}^{+}(n) + 2 \cdot \varepsilon^{+} \cdot \phi'(\mu_{\Lambda}(\mathbf{x}_i)) \cdot \xi_{\Lambda}^{+} \cdot \Lambda \cdot (\mathbf{x}_i - \mathbf{w}^{+}) \\ \mathbf{w}^{-}(n+1) &= \mathbf{w}^{-}(n) - 2 \cdot \varepsilon^{-} \cdot \phi'(\mu_{\Lambda}(\mathbf{x}_i)) \cdot \xi_{\Lambda}^{-} \cdot \Lambda \cdot (\mathbf{x}_i - \mathbf{w}^{-}) \end{aligned} \tag{3.29}$$

A matriz de pesos Ω é calculada através da Equação 3.30.

$$\begin{aligned} \Omega(n+1) &= \Omega(n) - \varepsilon_{\Lambda} \cdot 2 \cdot \phi_{\Lambda} \\ &\cdot (\xi_{\Lambda}^{+} \cdot (\mathbf{x}_m - \mathbf{w}_m^{+})[\Omega(\mathbf{x} - \mathbf{w}^{+})] \\ &- \xi_{\Lambda}^{-} \cdot (\mathbf{x}_m - \mathbf{w}_m^{-})[\Omega(\mathbf{x} - \mathbf{w}^{-})])) \end{aligned} \tag{3.30}$$

As distâncias derivativas são calculadas através da Equação 3.31.

$$\begin{aligned} \xi_{\Lambda}^{+} &= \frac{2 d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{-})}{(d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{+}) + d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{-}))^2} \\ \xi_{\Lambda}^{-} &= \frac{2 d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{+})}{(d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{+}) + d^{\Lambda}(\mathbf{x}_i, \mathbf{w}^{-}))^2} \end{aligned} \tag{3.31}$$

0 LGMLVQ (SCHNEIDER; BIEHL; HAMMER, 2009a) é uma variação do GMLVQ onde a matriz de pesos (Λ) é adaptada localmente de forma independente em torno de cada protótipo. Assim, Λ possui ajuste fino baseado na estrutura local dos dados.

Algoritmo 6 Algoritmo GMLVQ

Inicialize os parâmetros: ε^+, ε^-, ε^λ, τ, n, m.
Inicialize os protótipos: $\mathbf{W} = \mathbf{w}_1, , \mathbf{w}_2, ..., \mathbf{w}_m$ aleatoriamente.
for *j ← 1 número de épocas N* **do**
  for *i ← 1 tamanho do conjunto de dados* **do**
    Selecione uma amosta aleatória $\mathbf{x}_i$ i do conjunto de dados.
    Encontre os protótipos mais próximos da mesma categoria $\mathbf{w}^+$ e da categoria distinta $\mathbf{w}^-$.
    Atualize os protótipos de acordo com a Equação 3.29.
    $\mathbf{w}^+ = \mathbf{w}^+ + \Delta\mathbf{w}^+$.
    $\mathbf{w}^- = \mathbf{w}^- + \Delta\mathbf{w}^-$.
    Atualize a matriz de relevâncias de acordo com a Equação 3.30 $\Omega = \Omega + \Delta\Omega$.
  Realize o decaimento das taxas de aprendizagem de acordo com a Equação 3.16.
Atribuir as amostras às categorias vencedoras.

3.3 MODELO SOM

O SOM é uma rede neural que segue o processo de aprendizado não-supervisionado. Ele busca criar agrupamentos nos dados apresentados a ele, sendo uma abordagem de agrupamento que procura descobrir padrões subjacentes automaticamente. Os algoritmos de agrupamento visam identificar instâncias semelhantes, mantendo aquelas diferentes em regiões separadas no mapa. Esses algoritmos utilizam critérios de similaridade para agrupar conjuntos de dados (BENABDESLEM; LEBBAH, 2007). Em contraste, o LVQ considera a categoria da amostra durante o treinamento, enquanto o SOM não utiliza rótulos no algoritmo de treinamento.

O SOM foi inicialmente concebido com uma única camada de nós para realizar o agrupamento e visualização de dados, sendo empregado como um quantizador de dados. Seu treinamento é não-supervisionado e ocorre em um espaço de dimensão inferior à dos dados de treinamento (ASTUDILLO; OOMMEN, 2014).

O objetivo do mapeamento dentro do SOM é representar todos os casos dos dados de entrada quantizando-os para um conjunto menor de protótipos. Assim, os pontos próximos no espaço de entrada são representados por nós também próximos, enquanto os pontos distantes no espaço de entrada são representados por nós distantes no espaço-alvo (ASTUDILLO; OOMMEN, 2014).

Alguns conceitos cruciais para compreender o SOM são apresentados abaixo (KOHONEN; HARI, 1999; KOHONEN; SCHROEDER; HUANG, 2001):

- Estímulos de entrada, sinais de entrada ou simplesmente entrada da rede são vetores de

dados *D*-dimensionais, representados por $\mathbf{x} = [x_1\ x_2\ ...\ x_D]^\top$. Este vetor é uma lista de números que expressa os valores do estímulo em cada dimensão;

- Nodo ou unidade n_i possui um conjunto de valores numéricos ou pesos sinápticos, w_i. O vetor $\mathbf{w}_i = [w_{i1}\ w_{i2}\ ...\ w_D]^\top$ possui a mesma dimensão que $\mathbf{x}$ e pode ser considerado como uma posição no espaço de entrada;
- Nodo vencedor s é aquele cujo vetor sináptico $\mathbf{w}_s$ possui o maior grau de semelhança para um dado estímulo de entrada;
- Arestas ou conexões unem os nodos para formar suas vizinhanças.

O SOM utiliza uma malha A com N nodos. As posições, interconexões e pesos dos nodos são determinados em função do conjunto de entrada $\chi \subseteq R^D$. O algoritmo de treinamento busca organizar os nodos de forma a preservar a distribuição e topologia dos dados de entrada χ. Durante esse processo de formação, o vetor de pesos $\mathbf{w}_i \subseteq R^D$ de cada nodo $i \subseteq R^D$ é ajustado. O vetor $\mathbf{w}_i$ representa a posição do nodo da posição i no espaço de características.

Em cada passo do treinamento, um estímulo $\mathbf{x}$, um exemplo do conjunto de treinamento, é apresentado à rede, e os nodos competem entre si para identificar o nodo vencedor, conhecido como *Best Matching Unit* (BMU) (ASTUDILLO; OOMMEN, 2014). O treinamento do algoritmo SOM padrão inicia definindo uma estrutura topológica da rede cuja estrutura neural permanece estática durante o treinamento. O objetivo desse treinamento é ajustar os pesos $\mathbf{w}_i$ para preservar, ao máximo possível, as propriedades topológicas do espaço de entrada M_A. Em cada iteração, os pesos $\mathbf{w}_i$ são atualizados pela regra de atualização de pesos dada pela Equação 3.32, em que $\mathbf{w}_i(n+1)$ é o próximo peso para o nodo i, $\mathbf{w}_i$ é o peso atual para o nodo i, $\eta(n)$ é a função de taxa de aprendizagem, e $h_{v(x)}$ é a função de vizinhança. Este processo de atualização será descrito adiante.

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta(n)h_{v(x)}(n)(\mathbf{x} - \mathbf{w}_i(n)) \tag{3.32}$$

A determinação do BMU é uma etapa crucial no algoritmo SOM, na qual é identificado o nodo mais adequado para uma determinada entrada $\mathbf{x} \subseteq \chi$. O vetor $\mathbf{x}$ é apresentado ao SOM, e os nodos competem entre si para determinar qual responde mais fortemente ao estímulo do padrão de entrada. Assim, apenas um único vencedor é selecionado, o BMU, denotado por $s(\mathbf{x})$, conforme a Equação 3.33.

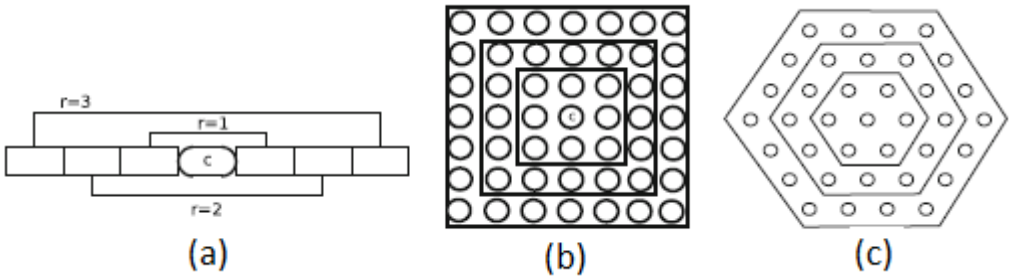

Figura 8 – (a) Grade Linear. (b) Grade Retangular. (c) Grade Hexagonal (ASTUDILLO; OOMMEN, 2014).

$$s(\mathbf{x}) = arg\min_i \|\mathbf{x} - \mathbf{w}_i\| \tag{3.33}$$

Uma vez determinado o nodo vencedor, alguns nodos além deste também sofrem ajuste. Esse subconjunto de nodos é conhecido como a vizinhança do BMU. Tal conceito envolve determinar quais nodos devem ser atualizados em cada etapa de aprendizagem de modo a alcançar a convergência. Em geral, o SOM inicia com uma vizinhança ampla que diminui de tamanho para um valor que tende a zero à medida que o número de iterações aumenta, conforme descrito na Equação 3.34. Na Figura 8, são apresentados exemplos de diferentes arranjos de nodos, como Grade Linear, Retangular ou Hexagonal.

$$h_{v(x)}(n) = \exp^{\left[-d_v/2\sigma^2(n)\right]} \tag{3.34}$$

A função de vizinhança determina que tanto o BMU quanto os nodos de sua vizinhança serão atualizados. No SOM padrão, a função de vizinhança é gaussiana, conforme apresentado pela Equação 3.34, em que $v(x)$ é o nodo vencedor, $d_{j,v(x)}$ é a distância entre o nodo vencedor e o nodo atual, e σ é o fator que decai em função das n épocas. Uma vez que essa função de vizinhança é calculada, o próximo passo do processo consiste na atualização dos nodos na vizinhança. Os pesos w_i do BMU e os nodos em sua vizinhança são atualizados utilizando a Equação 3.32. O conceito implícito dessa atualização em grupo é que esses nodos nas proximidades do BMU vão aprender parte das informações fornecidas pelo estímulo da entrada **x**.

A magnitude pela qual os nodos são movidos em direção ao estímulo é determinada pela taxa de aprendizagem, $\eta(n) \in [0, 1]$, calculada pela Equação 3.35, onde η_0 é a taxa de aprendizagem inicial, n é a época e τ é a taxa de decaimento exponencial. Em geral, a taxa de aprendizagem sofre um decaimento durante as épocas, tendendo a um valor próximo a zero conforme a Equação 3.35. O SOM padrão é descrito pelo Algoritmo 7.

$$\eta(n) = \eta_0 \exp^{-\frac{n}{\tau}} \tag{3.35}$$

Algoritmo 7 Pseudo-Algoritmo SOM

```
/* Definição de variáveis                                          */
1  X_i: vetores de entrada.
   m: quantidade de atributos do padrão X_i.
   q: quantidade de nodos da malha.
   η_0: taxa de aprendizado inicial.
   w: vetores de pesos.
   n: quantidade de épocas do algoritmo.

2  /* Fase de inicialização                                         */
3  Inicialize os pesos da malha de nodos com valores aleatórios próximos a zero [-0.01 +0.01].

4  for cada época n do
5      for cada X_i do
6          /* Fase de competição                                   */
7          Apresente o padrão de entrada X_i à rede.
           for cada nodo i do
8              Calcule a distância entre X_i e peso w_j.
9          Escolha o BMU através da Equação 3.33   /* Fase de Adaptação */
10         Atualize a função de vizinhança através da Equação 3.34.
           for cada nodo da vizinhaça do BMU do
11             Atualize os pesos conforme a equação 3.32.
12     Atualize a taxa de aprendizado 3.35
```

3.4 LVQ VERSUS SOM

O LVQ e SOM são modelos baseados em protótipos que e compartilham características. A principal diferença entre eles reside no método de aprendizado, no qual o LVQ é um modelo de aprendizado supervisionado, enquanto o SOM é não-supervisionado. Em redes neurais, a distinção entre métodos supervisionados e não-supervisionados está relacionada ao uso de rótulos durante o treinamento. Os métodos supervisionados utilizam rótulos para treinar o modelo, ajustando os pesos de conexão para minimizar a diferença entre as saídas do modelo e as saídas desejadas. Por outro lado, as redes não-supervisionadas procuram encontrar agrupamentos de entradas semelhantes, sem conhecimento prévio do dos rótulos para as amostras.

O SOM transforma dados de alta dimensionalidade em uma grade bidimensional, preservando a topologia dos dados. Isso é alcançado mapeando itens de dados semelhantes para o mesmo nodo na grade ou para células vizinhas. O algoritmo LVQ, embora relacionado ao SOM, não leva em consideração a topologia dos dados. Em vez disso, utiliza agrupamentos predefinidos por meio de rótulos para itens de dados de entrada, minimizando a probabilidade média esperada de categorização incorreta.

Enquanto o SOM gera automaticamente agrupamentos com base nas semelhanças dos itens, o LVQ trabalha com agrupamentos pré-definidos, representados pelos rótulos. Em comparação direta, quando os conjuntos de dados de treinamento são grandes, LVQ e SOM apresentam resultados semelhantes. No entanto, à medida que o tamanho do conjunto de dados diminui, os resultados do SOM tendem a deteriorar mais rapidamente do que os do LVQ (GOREN-BAR; KUFLIK, 2002).

3.4.1 SOM para Alta Dimensionalidade (*Subspace Clustering*)

O SOM padrão enfrenta desafios ao lidar com dados de alta dimensionalidade, devido à maldição da dimensionalidade, que torna muitas dimensões não discriminantes para o processo de categorização. Estratégias pioneiras para superar esse desafio incluem a redução da dimensionalidade dos dados ou o uso de modelos como *Gaussian Mixture Model* (GMM). Técnicas de redução de dimensionalidade envolvem extração e/ou seleção de características para preservar informações globais, enquanto modelos parcimoniosos requerem a estimação de parâmetros.

Os algoritmos de Agrupamento de Subespaço (SC) exploram diferentes subespaços para identificar agrupamentos distintos, com *Hard Subspace Clustering* (HSC) e *Soft Subspace Clustering* (SSC) como seus principais tipos. Os algoritmos HSC identificam subespaços exatos para diferentes agrupamentos, atribuindo 1 ou 0 a cada relevância. Por outro lado, os algoritmos SSC atribuem pesos de valor real a cada atributo de cada agrupamento para indicar sua importância específica.

Adaptações ao SOM original foram propostas para aprimorar a categorização, especialmente em relação a dados de alta dimensão. Essas adaptações são cruciais para superar os desafios impostos pela complexidade e quantidade de dados presentes em muitas aplicações modernas. Entre as melhorias propostas, destaca-se a implementação de arranjos de SOM, onde a saída de um SOM é conectada à entrada de outro. Essa abordagem em cascata permite que diferentes níveis de abstração sejam capturados, aumentando a capacidade do modelo

de reconhecer padrões complexos em dados de alta dimensionalidade.

Abordagens hierárquicas também foram exploradas para melhorar o desempenho do modelo. Nessas abordagens, SOMs são organizados em camadas, onde cada camada refina as categorias identificadas pela camada anterior. Esse refinamento local é essencial para capturar nuances que podem ser perdidas em uma análise única e global dos dados. Adicionalmente, o uso de técnicas de refinamento local, onde pequenos ajustes são feitos em áreas específicas do mapa auto-organizado, tem se mostrado eficaz para aumentar a precisão da categorização.

Além dessas técnicas, diversas variantes do SOM foram desenvolvidas para lidar com desafios específicos. O *Dimension Selective Self-organizing Map* (DSSOM) (BASSANI; ARAÚJO, 2012), por exemplo, foi projetado para identificar dimensões relevantes em conjuntos de dados de alta dimensionalidade, melhorando a capacidade do SOM de focar nas características mais informativas. O *Local Adaptive Receptive Field Dimension Selective Self-organizing Map* (LARFDSSOM) (BASSANI; ARAUJO, 2014) e sua versão aprimorada *Local Adaptive Receptive Field Dimension Selective Self-organizing Map 2* (LARFDSSOM2) (ARAÚJO; ANTONINO; PONCE-GUEVARA, 2020), oferecem adaptações locais dinâmicas que permitem ao SOM ajustar sua estrutura de nós de acordo com a complexidade dos dados, facilitando a mudança dinâmica na estrutura do modelo.

Mais recentemente, o Soft Subspace Neural Gas for Data Stream Clustering (S2G-Stream) (ATTAOUI et al., 2021) foi introduzido para lidar com fluxos de dados contínuos e em tempo real. Esse modelo é capaz de crescer e se adaptar conforme novos dados chegam, garantindo que o SOM mantenha sua relevância e precisão mesmo em ambientes de dados em constante mudança. Essa capacidade de adaptação é crucial em aplicações modernas, onde os dados não são estáticos e a capacidade de aprendizado contínuo é uma vantagem significativa.

Uma abordagem específica para dados intervalares foi apresentada por (HAJJAR; HAMDAN, 2013), que propôs uma versão de SOM baseada na distância de Mahalanobis para agrupamento de dados intervalares. Dois métodos foram propostos, um utilizando uma distância comum de Mahalanobis para todos os agrupamentos, e o outro adaptando a distância por agrupamento para melhor se adequar ao conjunto de dados fornecido. Essas adaptações destacam a flexibilidade do SOM e suas potenciais melhorias em diferentes contextos e desafios.

Sousa e Bassani (2022) apresentaram de mapeamento semântico topológico que utiliza características visuais profundas consolidadas para reconhecer propriedades semânticas em ambientes mapeados. Este trabalho propõe um método que consolida as características visuais profundas extraídas por uma CNN (GoogLeNet) para permitir o reconhecimento flexível

de propriedades semânticas e o uso em outras tarefas visuais. O método proposto se trata de um modelo baseado em SOM com estrutura crescente de nodos. As características visuais extraídas de imagens são consolidadas nos nodo do mapa por meio de uma média móvel, permitindo o reconhecimento de objetos e categorias de lugar. Além disso, o método foi avaliado quanto à localização de imagens no mapa ao longo do tempo, demonstrando resultados estáveis ou crescentes à medida que mais dados são consolidados no processo de mapeamento. Os experimentos realizados com um conjunto de dados de ambiente confinados mostraram que o método é capaz de consolidar as características visuais das regiões, reconhecer propriedades semânticas e localizar imagens topologicamente. Essas inovações mostram que o SOM continua a evoluir para atender às necessidades de categorização em contextos de alta dimensionalidade e grandes volumes de dados, mantendo-se relevante e eficaz através de diversas adaptações e melhorias contínuas.

3.4.2 SOM para Categorização de Cenas e Lugares

A pesquisa científica relacionada ao uso de SOM na etapa de categorização de cenas e lugares é limitada, sendo mais comum encontrar sua aplicação em tarefas como agrupamento e extração de características, como na construção de dicionários de palavras. Uma aplicação específica consiste em agrupar cores em uma imagem, resultando em uma redução de cores com base na distribuição presente na própria imagem (ARAÚJO; COSTA, 2009). Essa relativa escassez de estudos nessa área pode ser atribuída às dificuldades que modelos SOM e suas variações enfrentam ao lidar com vetores de alta dimensionalidade.

O desafio se torna mais evidente quando se trata da categorização de cenas e lugares, onde os modelos precisam lidar com vetores de alta dimensionalidade. No entanto, algumas versões específicas de SOM foram propostas para enfrentar esse desafio. O LARFDSSOM (BASSANI; ARAUJO, 2014) e sua versão atualizada, LARFDSSOM2 (ARAÚJO; ANTONINO; PONCE-GUEVARA, 2020), são exemplos de abordagens que visam resolver o problema da alta dimensionalidade nos SOMs. Experimentos realizados com esses algoritmos incluíram bases de dados de categorização de cenas, notadamente aquelas com alta dimensionalidade de dados extraídos por CNNs e algoritmos extratores de características de imagens determinísticos.

Além disso, o *Online Local Adaptive Receptive Field Dimension Selective Self-Organizing Map* (OLARFDSSOM) (SOUSA; BASSANI, 2018) representa uma abordagem de mapeamento semântico incremental projetada para agentes robóticos. Esse método incorpora um módulo

de mapeamento que cria de forma incremental um mapa topológico do ambiente, enriquecido com objetos reconhecidos ao redor de cada nó topológico, e um módulo de categorização de lugares, equipado com um SOM de aprendizado não-supervisionado incremental. Essas abordagens oferecem soluções potencialmente interessantes para os desafios enfrentados na categorização de cenas e lugares com dados de alta dimensionalidade.

3.4.3 Refinamento

O refinamento de algoritmos em tarefas de agrupamento busca identificar amostras erroneamente categorizadas, realocando-as para grupos mais apropriados, resultando na melhoria da precisão do agrupamento. Uma estratégia comum é a abordagem gulosa, onde um item significativo é movido a cada iteração para aprimorar o agrupamento. No entanto, essa estratégia pode convergir a mínimos locais, especialmente quando otimizações em larga escala exigem realocações extensas para melhorias substanciais.

Alternativas, como algoritmos capazes de escapar de mínimos locais, geralmente resultam em aumento significativo no custo computacional. O refinamento de agrupamento compreende avaliação da qualidade do agrupamento por meio de uma função objetivo e algoritmos eficientes para identificar conjuntos de amostras para realocação, visando melhorar a função objetivo.

Métodos de agrupamento centrados em centróides são comuns, agrupando dados pela similaridade com centróides. A determinação prévia do número de grupos é crucial, e os centróides são iterativamente atualizados até a convergência ou um número máximo de iterações. Métodos gaussianos são versáteis para modelar variáveis contínuas com distribuições desconhecidas, aproximando funções de densidade de probabilidade. Este método visa minimizar a distância entre elementos de um grupo e seus centróides, reduzindo a variância intra-grupo (RODRÍGUEZ-SOTELO et al., 2012).

Um exemplo de modelo de agrupamento centrado é o Agrupamento Baseado em Centróides (CBC). Esse modelo visa a minimização de uma função objetivo que avalia a qualidade da solução de agrupamento obtida por meio da iteração das atualizações dos centróides. A partição resultante em cada iteração é determinada pela atribuição de elementos ao subconjunto cujo centróide está mais próximo. O método de agrupamento baseado em centróides GIM + CBC, com uma abordagem bayesiana (RODRÍGUEZ-SOTELO et al., 2012), incorpora uma função objetivo que consiste na combinação linear de distribuições gaussianas centradas nos centróides de cada grupo, conforme ilustrado na Equação 3.36. Nessa equação, $p(\mathbf{x}_i, \mathbf{w}_j)$ representa

a probabilidade de $\mathbf{x}_i$ gerado por uma distribuição gaussiana centrada em $\mathbf{w}_j$, e $p(\mathbf{w}_j)$ é a probabilidade a priori do grupo com centróide $\mathbf{w}_j$. A formulação logarítmica é empregada para simplificar as operações, e o sinal negativo é utilizado para minimizar a função objetivo. O Modelo Iterativo Genérico (GIM) realiza atualizações iterativas dos centróides, com todos os centróides sendo atualizados antes de avaliar o critério de convergência. Esse método pode ser aplicado ao CBC.

$$f\left(C^{(r)}\right) = -\sum_{j=1}^{K} p(\mathbf{x}_i, \mathbf{W}_j) \cdot p(\mathbf{W}_j) \tag{3.36}$$

Essa abordagem de agrupamento é amplamente disseminada na literatura. Inicialmente, seleciona-se um conjunto de k pontos como centróides, e as amostras são alocadas aos grupos correspondentes com base na proximidade em relação aos centróides. Essa perspectiva destaca a importância da escolha dos centróides de forma a assegurar que cada amostra esteja próxima de pelo menos um deles. Essa abordagem tem suas raízes em duas perspectivas fundamentais: otimização combinatória e métodos de máxima verossimilhança. Sob esses contextos, os centróides podem ser conceituados como centros de funções de densidade de probabilidade (KLEINBERG, 2002).

O método de agrupamento baseado em centróides segue o procedimento geral representado pelo Algoritmo 8. Esse algoritmo envolve a especificação do número de grupos, a seleção dos centróides iniciais (através de métodos como aleatoriedade ou técnicas de inicialização), a atribuição de cada amostra ao grupo cujo centróide está mais próximo, a atualização iterativa dos centróides com base em critérios específicos do método, e a repetição dos passos 3 e 4 até que um critério de convergência seja satisfeito.

Algoritmo 8 Algoritmo de refinamento baseado em centróide

Defina o número de grupos.
Escolha os centróides iniciais (pode ser aleatório ou por meio de um método de inicialização).
Atribua cada amostra ao grupo cujo centróide é o mais próximo.
Atualize os centróides de acordo com os critérios do método utilizado.
Repita os passos 3 e 4 até que o critério de convergência seja alcançado.

3.4.3.1 Inferência Variacional

A Inferência Variacional representa um problema central na área de otimização, que se dedica a buscar abordagens analíticas para determinar a distribuição a posteriori por meio

de cálculos de variações. Esse processo de otimização de funções se caracteriza por um espaço de busca limitado pela distribuição a posteriori. A aplicação da inferência variacional viabiliza a obtenção de estimativas para os parâmetros que governam determinados modelos, e o desenvolvimento dessa abordagem frequentemente depende da função de probabilidade subjacente.

No contexto da aprendizagem de máquina, a inferência variacional bayesiana é amplamente utilizada para inferir a distribuição condicional das variáveis latentes, dadas as observações e os parâmetros. Esse conceito é também conhecido como a distribuição posterior das variáveis latentes. A expressão matemática que descreve essa probabilidade posterior está representada na Equação 3.37.

$$p(z|x,\alpha) = \frac{p(z,x|\alpha)}{\int_z p(z,x|\alpha)} \tag{3.37}$$

Geralmente, métodos de inferência aproximada, como a abordagem bayesiana, são utilizados para estimar a distribuição posterior de um modelo, uma vez que o cálculo direto dessa distribuição é computacionalmente proibitivo. Em outras palavras, a função de densidade de probabilidade da distribuição posterior assume uma forma complexa, frequentemente envolvendo integrações que não podem ser resolvidas analiticamente. Esse desafio é ainda mais acentuado em cenários de alta dimensionalidade dos dados, o que torna o cálculo dos parâmetros impraticável ou computacionalmente ineficiente para grandes conjuntos de dados.

Considerando aspectos mais detalhados, como o conceito de otimização variacional, esse modelo pode ser aplicado ao problema de inferência. Supondo um modelo completamente bayesiano no qual todos os parâmetros possuam distribuições a priori e que o modelo contenha variáveis latentes, além dos parâmetros. Para este algoritmo, definimos o conjunto de todas as variáveis observadas como $\mathbf{X} = x_1, \ldots, x_n$ e $\mathbf{Z} = z_1, \ldots, z_n$. O modelo probabilístico especifica a distribuição conjunta $p(\mathbf{X}, \mathbf{Z})$ e tem como objetivo encontrar uma aproximação para a distribuição posterior $p(\mathbf{Z}|\mathbf{X})$, bem como para a evidência do modelo $p(\mathbf{X})$.

$$\begin{aligned} \ln\ p(\mathbf{X}) &= L(q) + KL(q||p) \\ L(q) &= \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X},\mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \\ KL(q||p) &= \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z},\mathbf{X})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \end{aligned} \tag{3.38}$$

O processo de inferência variacional envolve a maximização do limite inferior $L(q)$, otimizado em relação à distribuição $q(\mathbf{Z})$, o que equivale a minimizar a divergência de Kullback-

Leibler entre p e q. Se todas as opções possíveis forem consideradas para $q(\mathbf{Z})$, o limite superior do limite inferior é alcançado quando a divergência de KL se anula, o que ocorre quando $q(\mathbf{Z})$ coincide com a distribuição posterior $p(\mathbf{Z}|\mathbf{X})$.

Os algoritmos que utilizam informações de covariância dos dados como parâmetro geralmente possuem um custo computacional substancial. Para superar essa limitação, os métodos de inferência variacional visam calcular uma inferência subsequente aproximada, onde a distribuição condicional das variáveis latentes é estimada com base em informações disponíveis.

A complexidade exponencial em algoritmos de estatística Bayesiana, que expressam inferências sobre quantidades desconhecidas através de cálculos com densidades posteriores, destaca-se. A inferência variacional, uma abordagem de aprendizado de máquina, aproxima densidades de probabilidade por meio de técnicas de otimização, sendo mais eficiente em velocidade comparada a métodos clássicos como a amostragem de Monte Carlo por Cadeias de Markov. A ideia fundamental é definir uma família de distribuições e encontrar aquela que melhor se ajusta à distribuição alvo, medindo a proximidade pela divergência de Kullback-Leibler. O modelo proposto incorpora a inferência variacional em uma etapa específica.

Em algoritmos de agrupamento, aprimoramentos por métodos de refinamento podem modificar grupos interativamente, envolvendo a realocação de itens, fusão de grupos semelhantes e exclusão de grupos inadequados. Este trabalho propõe uma abordagem Bayesiana para o refinamento, visando melhorar resultados de agrupamento por meio de iterações sucessivas. A abordagem Bayesiana proposta utiliza um modelo GMM para o agrupamento probabilístico, onde cada elemento possui uma probabilidade de pertencer a cada grupo. A proposta considera que os dados seguem um modelo GMM, tratando cada grupo como um componente da mistura. As probabilidades a posteriori são calculadas usando o teorema de Bayes, representando o grau de pertinência dos dados aos centróides dos grupos.

Os métodos baseados em distribuições gaussianas são amplamente utilizados devido à sua flexibilidade na modelagem de variáveis aleatórias contínuas cujas distribuições não são conhecidas. Apesar do sucesso desses métodos em diversas aplicações, eles podem apresentar um alto custo computacional quando aplicados a conjuntos de dados de alta-dimensionalidade. O objetivo fundamental desse tipo de método de agrupamento é encontrar uma partição dos dados que minimize a distância entre os elementos pertencentes a cada grupo e seu respectivo centróide, ou seja, que minimize a variação intra-classe. Essa abordagem encontra correspondência com o algoritmo clássico k-means.

3.5 DISCUSSÃO

Neste capítulo, discutem-se os modelos e métodos relevantes para a proposta desta tese, além de discutir modelos que servem como pontos de comparação para o modelo proposto. Foram analisados modelos como LVQ, SOM, e k-NN, juntamente com suas variações, especialmente considerando o contexto de alta dimensionalidade, que é inerente aos vetores de saída das CNNs.

Há arquiteturas de CNN que incorporam o SVM como a camada final de categorização. Nesta tese, propõe-se uma alternativa a essa abordagem por meio de modelos que baseaiam a distância por vetores de referência, como k-NN, LVQ, e SOM. Em comparação com o SVM, esses modelos podem oferecer vantagens em diversas situações, dependendo das características específicas dos dados e dos requisitos da aplicação. As famílias de modelos abordadas neste capítulo têm demonstrado eficácia no processo de categorização e reconhecimento de padrões, especialmente em cenários onde a interpretabilidade e a adaptabilidade a mudanças são cruciais (KOHONEN, 1995; KOHONEN; SCHROEDER; HUANG, 2001).

O SOM, em particular, organiza os dados em uma representação geralmente bidimensional, preservando a topologia dos dados. Essa organização é valiosa para a visualização e compreensão de padrões complexos (KOHONEN; SCHROEDER; HUANG, 2001). Todos esses modelos são eficientes e apresentam razoável complexidade computacional, tornando-os adequados para aplicações que demandam aprendizado incremental.

Os modelos discutidos neste capítulo visam abordar os desafios associados à alta dimensionalidade, buscando melhorar a acurácia do modelo e manter uma baixa complexidade computacional. Essa abordagem favorece a implementação desses modelos em plataformas robóticas, reduzindo a complexidade computacional sem comprometer a eficácia. O próximo capítulo abordará o modelo proposto nesta tese, discutindo como as ideias dos algoritmos analisados neste capítulo inspiraram nossa abordagem.

4 MÚLTIPLOS MAPAS AUTO-ORGANIZÁVEIS ESPECIALISTAS EM UMA CATEGORIA (MOCE-SOM)

Neste capítulo, é apresentado um modelo inovador para abordar desafios em tarefas de categorização, em particular na categorização de cenas, modelo que pode ser utilizado na construção de mapas semânticos (PRONOBIS et al., 2006). O método proposto também pode ser utilizado para outras contextos de categorização como para mineração de dados e objetos.

O modelo proposto assume uma arquitetura composta por múltiplos módulos. O conhecimento sobre uma dada categoria é incorporado por um módulo em particular. Cada um dos módulos é composto por um conjunto de Mapas Auto-Organizáveis (SOMs) especializados dispostos em série. Cada módulo é treinado para realizar o agrupamento de dados pertencentes a uma categoria específica, o que leva à criação de uma representação compacta das características das amostras pertencentes a essa categoria (KOHONEN, 1995). Os SOMs em série dentro de um módulo visam incorporar características mais sutis entre os subgrupos de uma categoria. É óbvio que o número de módulos iguala o número de categorias. Em síntese, nesta arquitetura, cada módulo é treinado independentemente dos demais e cria protótipos para uma categoria. As camadas dentro de um módulo visam aproximar sutilmente as amostras de uma mesma subcategoria, isso é, reduzir o campo receptivo de um protótipo. Uma vez que todos os módulos tenham sido treinados e todos os protótipos para cada categoria tenham sido determinados, um dado de teste vai procurar sua unidade de melhor casamento (nodo vencedor) entre o conjunto união de todas as subcategorias (protótipos) de todas as categorias (módulos).

A abordagem adotada se baseia na exclusividade de treinamento de cada módulo do SOM com amostras de uma única categoria. Isso não apenas preserva a exclusividade dos dados de treinamento, mas também facilita o processo de aprendizado de padrões intrínsecos a cada categoria (CSURKA et al., 2004b). Cada módulo do SOM dentro do modelo proposto tem a capacidade de aprender um vetor protótipo além de um vetor de relevância por SOM em cada módulo. Este vetor estima a importância das características aprendidas para cada categoria. A relevância tem dois fatores importantes que são compostos no modelo. O primeiro visa enfatizar as dimensões mais importantes para caracterizar uma amostra como pertencente a uma categoria, esta é a relevância intra-categoria. O segundo fator objetiva identificar os atributos mais importantes para distinguir diferentes categorias, a relevância inter-categoria. Todo este procedimento ainda passa por um processo de refinamento que busca fazer pequenos

ajustes nos centroides (protótipos), sem alterar relevâncias, com base na distribuição espacial das amostras.

Este capítulo está estruturado de maneira a apresentar o modelo proposto, iniciando com as motivações que levaram à sua concepção. Será tratada a estrutura geral do algoritmo *Multiple One-Category Expert Self-Organing Maps* (MOCE-SOM), detalhando a etapa de treinamento do *Subspace Clustering Self-organizing Map* (SC-SOM), destacando o papel das relevâncias nesse processo, com suas equações e uma explicação sobre o propósito de cada etapa. Segue a descrição do procedimento de incorporação de informações inter-categorias, passando pela proposta de combinação das amostras com os protótipos do SC-SOM que ocorre durante a propagação de sinais entre camadas. Por último, apresenta-se o refinamento por meio da inferência variacional. Cada etapa será acompanhada pela definição das equações, pseudo-códigos e diagramas necessários para descrição do modelo.

4.1 MOTIVAÇÃO E CARACTERÍSTICAS DO MODELO

O modelo proposto, denominado Múltiplos Mapas Auto-organizáveis Especialistas em Uma Categoria (*Multiple One-Category Expert Self-Organing Maps* (MOCE-SOM)) lida com desafios substanciais associados a conjuntos de dados volumosos, uma ampla gama de categorias e alta dimensionalidade. Desafios que caracterizam um conjunto de problemas complexos de categorização. A abordagem principal do MOCE-SOM envolve a introdução de módulos especialistas denominados Mapas Auto-organizáveis para Agrupamento em Subespaço (*Subspace Clustering Self-organizing Map* (SC-SOM)), que são treinados independentemente para lidar com categorias específicas, uma única categoria. Assim, um problema complexo é dividido em subproblemas, permitindo que o modelo lide eficazmente com problemas complexos de categorização.

A relevância intra e inter-categorias desempenha um papel crucial na eficácia do MOCE-SOM. Ao introduzir relevâncias que consideram tanto os valores de atributos compartilhados dentro de uma categoria quanto os valores de atributos diferentes entre categorias, o modelo adapta dinamicamente a ponderação de dimensões por categoria durante o treinamento para categorização. Espera-se que o resultado seja aprimoramento na capacidade de distinguir nuances que produzam categorização mais precisa e discriminativa. O processo de aprendizagem de relevâncias intra foi inspirado em modelos como LARFDSSOM (BASSANI; ARAUJO, 2014) e sua versão atualizada, LARFDSSOM2, já introduação das relevâncias inter não é abordada

por esses modelos.

O conceito de categorizador de uma categoria, ou categorizador unívoco, refere-se a um modelo de aprendizado de máquina projetado para lidar com tarefas em que a ênfase está na categorização de uma única categoria, enquanto ignora ou agrupa em uma única categoria todas as instâncias que não pertencem a essa categoria de interesse. Diferentemente dos categorizadores tradicionais que lidam com múltiplas categorias, o categorizador unívoco visa identificar padrões e características distintivas exclusivas de uma categoria de interesse (KHAN; MADDEN, 2014). No contexto do MOCE-SOM, a incorporação do conceito de categorizador de uma categoria representa uma abordagem estratégica para a categorização de cenas. Ao se concentrar em cada categoria individualmente por meio dos módulos SC-SOM especialistas, o modelo segue ao conceito de categorizador treinado com uma única categoria, destacando as características distintivas intrínsecas de cada categoria. Busca-se criar capacidade apurada para identificar padrões específicos e complexidades únicas associadas a cada categoria de cena

No aprendizado de máquina, existem três abordagens principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. No aprendizado supervisionado, os modelos são treinados com um conjunto de dados rotulados, onde cada entrada está associada a uma saída conhecida, permitindo ao modelo aprender a mapear entradas para saídas corretas. No aprendizado não supervisionado, por outro lado, os modelos são treinados com dados sem rótulos, no qual o objetivo é descobrir padrões ou estruturas ocultas nos dados, como agrupamento e redução de dimensionalidade. O modelo Múltiplos Mapas Auto-organizáveis Especialistas em Uma Categoria (MOCESOM), no entanto, não se encaixa estritamente em nenhuma dessas categorias tradicionais. O MOCESOM é projetado para aprender e categorizar cenas em ambientes de visão computacional, utilizando uma abordagem multi-modular onde cada módulo é especializado em uma categoria específica. Este modelo combina aspectos de aprendizado supervisionado, ao se especializar em categorias definidas, e aprendizado não supervisionado, ao descobrir relevâncias intra e inter-categorias de forma autônoma focando em somente uma categoria por vez.

O MOCE-SOM incorpora uma etapa de refinamento através de inferência variacional, ampliando ainda mais sua capacidade de lidar com complexidades intrínsecas nos dados. A inferência variacional permite ajustar as distribuições dos parâmetros do modelo, aprimorando a qualidade das representações aprendidas. Essa etapa final de refinamento contribui para a robustez e precisão do modelo, consolidando sua aplicabilidade em cenários desafiadores

de categorização de cenas. Além de ser extensível na solução de outras problemas como a categorização de objetos e mineração de dados.

Um modelo precursor do MOCE-SOM, denominado *Subspace Clustering Multi-module Self-organizing Map* (SC-MuSOM) (JÚNIOR; ARAÚJO, 2022). Este modelo consiste em uma única camada de SC-SOM, no qual apenas relevâncias intra-categorias são aprendidas para categorização em mineração de dados, objetos e cenas. Esse modelo foi proposto para reconhecimento de lugares (SILVA et al., 2023). Posteriormente, uma versão aprimorada foi submetida ao *International Conference on Artificial Neural Networks* (ICANN) 2024, denominada *Subspace Clustering Multi-Layer Self-organizing Map* (SC-MLSOM), introduzindo o conceito de multi-camadas que envolve a composição de amostras para a propagação entre as camadas. O modelo apresentado nesta tese expande a proposta do SC-MLSOM, incorporando a influência de relevâncias inter-categorias. Além disso, ao final da rede, é incluída uma etapa de refinamento por meio de inferência variacional.

4.2 MÚLTIPLOS MAPAS AUTO-ORGANIZÁVEIS ESPECIALISTAS EM UMA CATEGORIA - MOCESOM

O MOCE-SOM visa lidar com questões como a velocidade de convergência, capacidade de lidar com dados de alta dimensionalidade, eficácia e escalabilidade. O modelo foi validado para categorização em problemas envolvendo reconhecimento de padrões e categorização de objetos ou cenas.

A arquitetura do MOCE-SOM possui vários módulos (chamados de módulo de categoria daqui em diante). Cada módulo de categoria configura um categorizador especialista, portanto, cada módulo é treinado para agrupar padrões pertencentes a uma única categoria, criando protótipos das amostras dessa categoria. O processo de treinamento por categoria pode facilitar a determinação dos protótipos representativos de cada categoria (CSURKA et al., 2004b). Além disso, cada módulo de categoria também aprende um único vetor de relevância para codificar a importância das características dos dados para cada categoria. O MOCE-SOM é baseado no SC-MuSOM (JÚNIOR; ARAÚJO, 2022) com as principais alterações:

- O MOCE-SOM possui múltiplas camadas;
- O MOCE-SOM treina os seus módulos de forma independente para cada uma das categorias;

- Durante o treinamento do modelo são aprendidas as relevâncias intra-categorias para ajuste da distância entre as amostras e os protótipos;
- O MOCE-SOM possui a contribuição de informações inter-categoria para melhoria da acurácia do modelo;
- O MOCE-SOM inclui uma composição de amostra para o sinal de entrada entre as camadas.
- O MOCE-SOM possui uma etapa de refinamento com inferência variacional no final de todas as camadas;

O MOCE-SOM emprega a estratégia de dividir e conquistar para categorizar dados, reduzindo assim a complexidade do processo. Essa propriedade, aliada ao processo de aprendizado de relevâncias para os atributos, torna o MOCE-SOM potencialmente útil em diferentes contextos com um grande número de categorias de dados, dimensões e instâncias. Essas características tornam o MOCE-SOM facilmente paralelizável, pois o treinamento pode ser feita de forma independente. Além disso, as múltiplas camadas do modelo proporcionam um aprimoramento na precisão da categorização. Por fim, é importante destacar que o MOCE-SOM requer conhecimento prévio sobre o número de categorias e os rótulos associados às amostras.

4.2.1 Papel do SC-SOM no MOCE-SOM

MOCE-SOM é composto por um conjunto de SC-SOM, conectados em série por categoria. Cada SC-SOM consiste em uma camada SOM (mapa este com um vetor de peso associado a cada nodo e um vetor de relevância único para o mapa), seguido por um misturador de entrada. Uma entrada é apresentada ao primeiro SC-SOM, que a processa e envia a saída para o próximo SC-SOM. Esse processo se repete até a última camada de saída. Um conjunto de SC-SOMs conectados em série forma um módulo de categoria.

Na etapa de treinamento, cada módulo de categoria recebe instâncias de uma categoria específica como entrada. O primeiro SC-SOM é treinado com esses exemplos. Portanto, uma vez concluído esse primeiro processo de aprendizado, o segundo SC-SOM é treinado. Esse procedimento continua até que o último SC-SOM do módulo seja treinado. O primeiro SC-SOM recebe um exemplo de uma determinada categoria. Todos os outros SC-SOMs subsequentes recebem uma combinação do exemplo apresentado ao primeiro SC-SOM e do protótipo

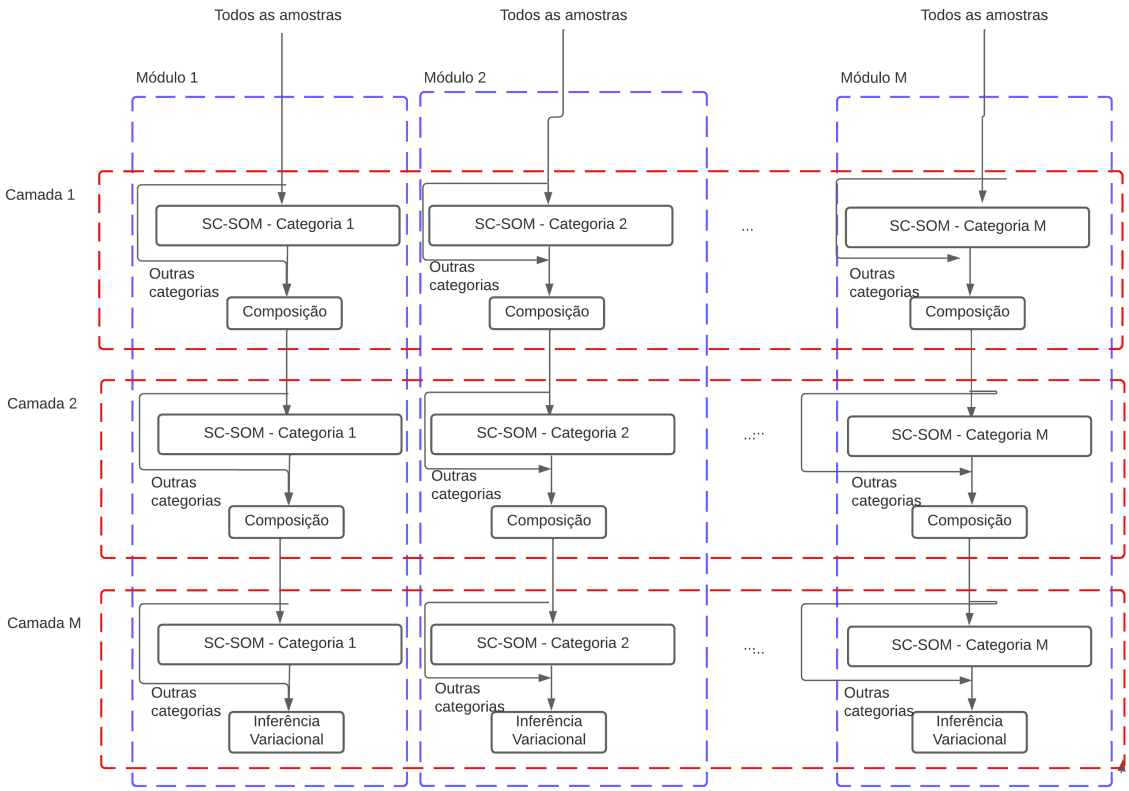

Figura 9 – Arquitetura do modelo proposto.

vencedor da camada anterior. Essa entrada combinada aproxima ligeiramente as amostras de entrada de um protótipo vencedor, acentuando a similaridade entre exemplos de uma mesma subcategoria, representada por um protótipo.

O modelo proposto inclui vários módulos de categorias dispostos em filas paralelas, cada um considerado como um SOM especialista. Cada módulo de categoria possui N camadas (Figura 9). Não há cruzamento de sinal entre os módulos. Após a etapa de aprendizado, qualquer entrada fornecida ao MOCE-SOM é processada por todos os módulos. Cada módulo de categoria produz um vencedor para uma entrada específica, sendo que o vencedor com o maior valor de ativação é a saída do MOCE-SOM, determinando a categoria da amostra de entrada. Esse procedimento é delineado no Algoritmo 9.

Algoritmo 9 Modelo Proposto - MOCE-SOM

L é quantidade de camadas do modelo.
for $i \rightarrow L$ **do**
  Algoritmo 10 - Treinamento da Camada i.
  if $i < N$ **then**
    Execute a composição através da Eq. 4.13.
  Algoritmo 11 - Para inferência Variacional.
  Algoritmo 12 - Para predição das categorias.

4.2.1.1 Treinamento de cada SC-SOM

Em um SC-SOM, os vetores de pesos associados a cada nodo são inicializados pela média aritmética das instâncias de treinamento, somada a um valor aleatório diferente para cada instância no intervalo de [-0.01 a $+0.01$]. Por sua vez, a inicialização de um componente do vetor de relevância associado a um determinado SC-SOM é igual a um menos a variância de cada atributo para a categoria. Como todos os valores de atributos são normalizados no intervalo [0, 1], as variações pertencem à mesma faixa. As relevâncias intra são inicalizadas com o valor unitário. No final da primeira época as relevâncias são calculadas pela Equação 4.1, no qual $\mathbf{w}_j$ são os vetores de cada uma dos centros do SC-SOM j. O cálculo iterativo das relevâncias compreende a soma da relevância chamada instantânea (Equação 4.1) com as relevâncias acumuladas até então. Este cálculo será definido logo mais ainda nesta Seção.

$$\mathbf{r}_{j/insta} = (\mathbf{1} - \mathbf{variancia}(\mathbf{w}_j)) \tag{4.1}$$

O valor de relevância do k-ésimo atributo atribui a importância dessa característica para a distância entre dois vetores $\mathbf{x}$ e $\mathbf{y}$ (Equação 4.2). Dimensões caracterizadas por menor variância tendem a ser mais significativas (ter maiores relevâncias) na determinação da distância entre dois vetores. A estratégia adotada para a inicialização e cálculo das relevâncias decorre dos princípios de auto-organização e adaptabilidade, destacando a capacidade do SOM em modelar as interações intrínsecas dos dados.

$$d_r(\mathbf{x}, \mathbf{y}) = \sqrt{r_1 \cdot (x_1 - y_1)^2 + ... + r_N \cdot (x_N - y_N)^2} \tag{4.2}$$

Um módulo de categoria gera um conjunto de subcategorias representadas pelos nodos dentre de cada categoria. Considerando cada nodo j pertencente a um SC-SOM de um módulo

de categoria, a identificação do nodo vencedor $(\mathbf{w}_j)$ é realizada através do cálculo da distância ponderada pela relevância entre um padrão de entrada $\mathbf{x}$ e os vetores de peso associados a cada nodo (Equação 4.2). Durante o treinamento, o objetivo é determinar o nodo vencedor entre os nodos existentes em cada SC-SOM dentro de cada módulo de categoria através da Equação 4.3.

$$\|\mathbf{x} - \mathbf{w}_j\| = \min_j d_r(\mathbf{x}, \mathbf{w}_j) \tag{4.3}$$

A atualização de peso de cada SC-SOM é realizada para um módulo de categoria de cada vez. Portanto, considerando uma categoria específica, um vetor de entrada é apresentado, em seguida, o nodo vencedor é determinado e seu centróide $(\mathbf{w}_j)$ é ajustado de acordo com a Equação 4.4.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n) \cdot h_w(n) \left[\mathbf{x} - \mathbf{w}(n)\right] \tag{4.4}$$

onde $\mathbf{w}_j$ é o protótipo do vencedor; para os nodos vizinhos, o índice w é substituído pelo índice $j \in N(\mathbf{w}_j)$.

A taxa de aprendizagem, $\eta(n)$, é determinada pela Equação 4.5, onde n denota o número da épocas. A vizinhança $(N(\mathbf{w}))$ é definida pela Equação 4.6 que determina o alcance dos vetores de peso a serem atualizados. O valor da função de vizinhança h_w diminui à medida que a distância entre $\mathbf{w}_j$ e $\mathbf{w}$ que é o protótipo vencedor aumenta. Geralmente, essa função é modelada como uma distribuição gaussiana. Nesse contexto, uma estratégia comum envolve definir uma vizinhança topológica que se contrai ao longo do tempo.

$$\eta = \eta(0) \cdot e^{-\frac{n}{\tau_\eta}} \tag{4.5}$$

$$h_w(n) = h_0 \cdot \exp^{-\frac{\left\|\mathbf{w}_j - \mathbf{w}\right\|^2}{\sigma^2(n)}} \tag{4.6}$$

A diminuição da vizinhança topológica é determinada pela Equação 4.7. Ela visa garantir que a influência do nodo vencedor em seus vizinhos diminua progressivamente à medida que o processo de treinamento avança.

$$\sigma(n) = \sigma_0 \cdot e^{-\frac{n}{\tau}} \tag{4.7}$$

Em cada época no treinamento, todos os padrões são apresentados de forma aleatória para cada módulo de categoria. Após cada época, o vetor de relevância instantânea da categoria

em treinamento é atualizado pela Equação 4.1. Uma vez calculada a relevância instantânea, o vetor resultante é normalizado usando a normalização Min-Max, conforme expresso na Equação 4.8, onde p denota cada um dos atributos no vetor e P é o número total de atributos. O vetor de relevância normalizado é então usado na Equação 4.9 para atualizar as relevâncias. Esse processo de atualização leva em consideração as relevâncias instantâneas anteriores, de épocas passadas. Notadamente, esse processo ocorre com a influência de um fator de ponderação λ da importância da recência, que diminui exponencialmente ao longo das iterações temporais, seguindo o comportamento descrito na Equação 4.10. Essa atualização ocorre apenas uma vez por época.

$$\mathbf{r}_{jp/norm} = \frac{r_{jp/insta} - min(r_{jp})}{max(r_{jp}) - min(r_{jp})},\ onde\ p = 1, ..., P \tag{4.8}$$

$$\mathbf{r}_j(n+1) = \lambda \cdot \mathbf{r}_{j/instaNorm} + (1-\lambda) \cdot \mathbf{r}_j(n) \tag{4.9}$$

$$\lambda(n) = \lambda(0) \cdot e^{-\frac{n}{\tau_\lambda}} \tag{4.10}$$

4.2.1.2 Inserção da Influência entre Grupos nas Relevâncias

Durante o treinamento de cada SC-SOM, ajusta-se o vetor de relevâncias que considera atributos importantes, aqueles que compartilham valores entre si dentro de cada categoria. Contudo, as relações entre mesmos atributos de grupos diferentes também são importantes uma vez que busca-se aquelas dimensões que têm valores marcadamente diferentes entre os diferentes grupos. Nesta etapa do processo, busca-se a incorporação de elementos que promovam a distinção entre as categorias.

Com o intuito de alcançar esse objetivo, realiza-se o cálculo das variâncias entre os vetores de relevância associados a cada categoria. A relevância inter-categoria é definida como a diferença entre 1 e essa variância (considerando que a variância está sempre no intervalo [0,1], devido à normalização dos dados de entrada da rede para o intervalo [0, 1]), conforme apresentado na Equação 4.11. A normalização da relevância inter é realizada pela Equação 4.8. A relevância final é então determinada por meio da combinação ponderada, expressa na Equação 4.12, na qual β representa uma variável sujeita a parametrização. Esta equação

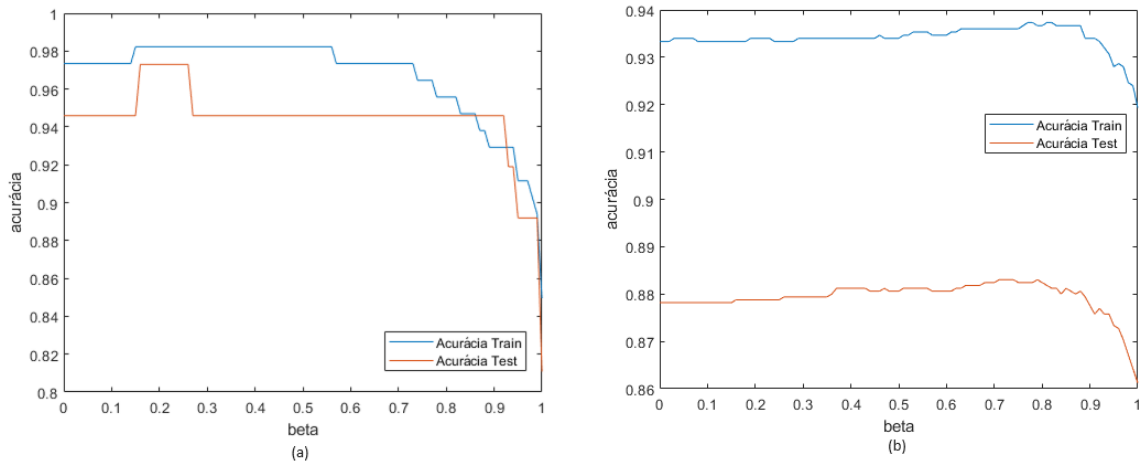

Figura 10 – Análise da influência do β na acurácia do conjunto de treinamento e teste. (a) Base de dados: Iris (b) Base de dados: Scenes_15. Fonte: Autor

incorpora de maneira sinérgica as influências simultâneas das relevâncias intra e inter-categoria, resultando em uma relevância final mais refinada e discriminativa.

$$r_{j/inter} = (1 - variancia(\mathbf{r}_j)) \tag{4.11}$$

$$\mathbf{r}_j(n+1) = \beta \cdot \mathbf{r}_j + (1-\beta) \cdot \mathbf{r}_{j/inter}(n) \tag{4.12}$$

Resolver a parametrização do valor de β é importante no contexto do modelo, pois este parâmetro busca maximizar a função de acurácia com a influência da relevância inter. A investigação do impacto das relevâncias inter-categorias na acurácia do modelo se traduz em um problema contínuo e unimodal (por aproximação). Uma abordagem que pode ser adaptado para abordar essa busca é por meio do Método da Bissecção. Este método se mostra eficaz na resolução do desafio de encontrar o valor ótimo de β que maximiza a acurácia do modelo. A Figura 10 ilustra o comportamento das funções em diversos conjuntos de dados, evidenciando a natureza bem-comportada da função que é aproximadamente unimodal. O Método da Bissecção é eficaz para funções que não possuem máximos locais. O Método da Bissecção tem complexidade $O(log(n))$. Quanto menor for o β que maximize a acurácia melhor, pois isso irá modificar a avaliação das amostras em relação a categorização sem a influência das relevâncias inter-categorias.

O pseudo-código do SC-SOM com inserção da influência entre Grupos nas relevâncias é a apresentado no 10.

Algoritmo 10 SC-SOM com inserção da influência entre Grupos nas relevâncias

Inicialize todos os centros $\mathbf{w}_j(0)$ e os parâmetros de entrada η_0, η_f, λ_0, λ_f, h_0, ρ, N que o número de épocas.
for $n = 0$ *até* N **do**
    for *todos os padrões de treinamento* $\boldsymbol{x}_t$ *da categoria* **do**
        Padrão de treinamento $\boldsymbol{x}_t$ no tempo t.
        Selecione o nodo vencedor $\boldsymbol{x}_t$ pela Eq. 4.3.
        Atualize os pesos, $\boldsymbol{w}$ vencedor, dos nodos e vizinhos usando a Eq. 4.4.
    Calcule o $\mathbf{r}j/insta$ através da Eq. 4.1.
    Normalize o $\mathbf{r}j/insta$ através da Eq. 4.8.
    Atualize o $\boldsymbol{r}_j$ para todos os nodos através da Eq. 4.9.
    Atualize o $\eta(n)$ através da Eq. 4.5.
    Atualize o $\lambda(n)$ através da Eq. 4.10.
    Atualize o $\sigma(n)$ através da Eq. 4.7.
Calcule a relevância inter utilizando as relevâncias intras através da Equação 4.11.
Normalize o $\mathbf{r}_{inter}$ através da Eq. 4.8.
Encontre o valor de β que maximiza a acurácia do conjunto de treinamento através do Algoritmo Busca Binária.
Calcule as relevâncias finais através de Eq. 4.12 para β.

4.2.2 Composição

A MOCE-SOM é possui o processamento execurado por camada. Após cada etapa do processo a informação é congelada e o processo passado adiante. A saída do SC-SOM no MOCE-SOM é determinada pela Equação 4.13, onde ρ representa o fator de composição e j é o índice do nó vencedor. Todos os módulos SC-SOM são considerados para se determinar um único nodo vencedor, assim, há somente um nodo svencedor para todas os SC-SOMs. O cálculo distância entre a amostra e o protótipo segue a Equação 4.2 que leva em consideração o vetor de relevância de cada SC-SOM de forma independente. O valor de ρ é ajustado para um valor próximo a zero, assegurando uma contribuição pequena do nodo vencedor para a composição. Portanto, a amostra que é produzida no módulo da categoria vencedora é o resultado da composição da amostra com uma pequena contribuição do valor do nodo vencedor. Para todos os outros módulos de categoria (que não contêm o nodo vencedor), a amostra propagada não sofre contribuição dos protótipos desse módulo, permanecendo inalterada.

Cada módulo de categoria possui um fluxo de dados independente para todas as amostras. Contudo, somente as amostras desse módulo de categoria vitoriosa participam do treinamento, enquanto todas as amostras seguem para a avaliação. Neste modelo, dado que apenas um

pequeno número de amostras de entrada é combinado para gerar novos valores, a maioria mantém os valores originais. Esses valores podem ser referenciados ao valor original da amostra de entrada do modelo, eliminando a necessidade de replicação de dados para as camadas módulo nas camadas subsequentes. Dessa forma, mesmo que o algoritmo apresente vários módulos no caso de muitas categorias, a quantidade de memória alocada cresce linearmente em relação à quantidade de amostras no pior cenário, não dependendo da multiplicação da quantidade de amostras pela quantidade de categorias.

$$\mathbf{x}(n+1) = (1-\rho) \cdot \mathbf{x}(n) + \rho \cdot \mathbf{w}_j(n) \tag{4.13}$$

4.2.3 Refinamento através de Inferência Variacional

Na etapa de Refinamento, as relevâncias não sofrem mais atualizações, mantendo seus valores fixos ao longo de todo o processo. Dado que as relevâncias desempenham um papel crucial na influência sobre os vetores de amostras e pesos dos centróides, a imutabilidade de seus valores é adotada para evitar redundâncias computacionais no algoritmo e simplificar a notação. Daqui em diante, todas as amostras e pesos estarão transformados pelas relevâncias. Essa transformação é realizada através da obtenção de uma matriz $\mathbf{R}_j$, na qual $\mathbf{r}_j$ representa a diagonal da matriz, e a subsequente multiplicação pelos vetores de amostra e centros correspondentes a cada categoria.

Nesta tese, as características dos dados são tratadas como funções de densidade condicional, representadas por $p(\mathbf{x}_i|\mathbf{w}_j)$, indicando a probabilidade de um vetor de entrada $\mathbf{x}_i$ pertencer a um grupo $\mathbf{w}_j$. A abordagem adotada para o agrupamento baseia-se no estudo do grau de associação de um elemento a um grupo e na influência de cada elemento na atualização dos centróides. A função de associação $m(\mathbf{w}_j|\mathbf{x}_i)$ estima a associação de $\mathbf{x}_i$ ao grupo cujo centróide é $\mathbf{w}_j$.

Métodos gaussianos são frequentemente empregados devido à sua flexibilidade na modelagem de variáveis aleatórias de valores reais com distribuições desconhecidas, sendo comuns para a aproximação em diferentes contextos (ZHU; GHAHRAMANI; LAFFERTY, 2003). Apesar de serem eficazes em diversas aplicações, esses métodos tornam-se computacionalmente custosos em dados de alta dimensão (RODRÍGUEZ-SOTELO et al., 2012). O objetivo principal do agrupamento é encontrar uma partição dos dados que minimize a distância intra-grupo, buscando reduzir a variância dentro de cada classe.

No contexto de agrupamento baseado em centróides, assume-se que as características dos dados possuem funções de densidade condicional, representadas por $p(\mathbf{x}_i|\mathbf{w}_j)$, onde um vetor de entrada $\mathbf{x}_i$ pode pertencer a uma categoria $\mathbf{w}_j$. A abordagem geral envolve a análise da associação de elementos aos grupos, estimada pela função de associação $m(\mathbf{w}_j|\mathbf{x}_i)$, que quantifica a associação de $\mathbf{x}_i$ ao grupo com centróide $\mathbf{w}_j$. Este valor de associação varia entre 0 e 1, indicando o grau de participação de um elemento em um grupo específico.

A ideia geral do agrupamento baseado em centróides (CBC) é minimizar uma função objetivo, que define quão boa é a solução do agrupamento; essa solução é obtida iterativamente através da atualização de centróides, no qual a partição resultante para cada iteração corresponde à atribuição dos elementos ao subconjunto cujo centróide é o mais próximo. O método Bayesiano de agrupamento baseado em centróides GIM + CBC (RODRÍGUEZ-SOTELO et al., 2012) tem como função objetivo a combinação linear de distribuições gaussianas centradas nos centróides de cada grupo como na Equação 3.36, no qual $p(\mathbf{x}_i|\mathbf{w}_j)$ é a probabilidade de $\mathbf{x}_i$, gerado por uma distribuição gaussiana centrada em $\mathbf{w}_j$ e $p(\mathbf{w}_j)$ é a probabilidade a priori do grupo cujo centróide é $\mathbf{w}_j$. A função logaritmo é usada para facilitar o cálculo matemático para que a tarefa seja minimizar a função objetivo.

O agrupamento iterativo baseado em centróides possui um determinado grau de associação entre um elemento em seu grupo que caracteriza a influência de cada elemento na atualização dos centróides. O grau de associação de $\mathbf{x}_i$ no grupo cujo centróide é $\mathbf{w}_j$ e o número de grupos é k. O grau de associação é um valor não negativo e associação absoluta é 1; portanto, a função $m(.)$ deve satisfazer: $m(\mathbf{w}_j|\mathbf{x}i) \geq 0$ e $\sum_{j=1}^{k} m(\mathbf{w}_j|\mathbf{x}_i) = 1$. As funções correspondentes à associação e peso são, respectivamente, a Equação 4.14 e 4.15. A atualização dos centróides pode ser escrita pela Equação 4.16, no qual r representa a iteração do algoritmo, porém $(r-1)$ corresponde à inicialização.

$$m(\mathbf{w}_j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|\mathbf{w}_j) \cdot \mathbf{w}_j}{p(\mathbf{x}_i)} \tag{4.14}$$

$$\omega(\mathbf{x}_i) = 1 \tag{4.15}$$

$$C_j^r = \frac{\sum_{i=1}^{n} m(C_j^{(r-1)}) \cdot w(\mathbf{x}_i) \cdot \mathbf{x}_i}{\sum_{i=1}^{n} m(C_j^{(r-1)}) \cdot w(\mathbf{x}_i)} \tag{4.16}$$

Dado que $m(\mathbf{x}_i|\mathbf{w}_j)$ corresponde a uma distribuição Gausiana, pode ser calculado pela Equação 4.17, μ é o centróide $(\mu = c_j)$, d é o número de dimensões ou atributos, Σ representa a

matriz de covariância do cluster j e $det(\cdot)$ denota o determinante de sua matriz de argumentos (BISHOP; NASRABADI, 2006). Os algoritmos CBC são sensíveis à inicialização, ou seja, uma partição inicial não adequada pode levar o algoritmo a convergir para um mínimo local distante do mínimo global. Para assegurar uma boa inicialização dos centróides há algumas alternativas como o algoritmo máximos e mínimos. No nosso caso, a inicialização será realizada através dos nodos resultantes do mapa gerado pelo método de agrupamento com relevâncias. Portanto, pode-se dizer que a inicialização está sendo aprimorada com conhecimento prévio. O cálculo da evidência $p(\mathbf{x}_i)$ é dado pela Equação 4.18.

$$p(\mathbf{x}_i|\mathbf{w}_j) = \frac{1}{det(\sum_j)^{1/2}} \cdot (2\pi)^{-d/2} \cdot e^{-1/2 \cdot (\mathbf{x}_i - \mathbf{w}_j) \sum_j^{-1} (\mathbf{x}_i - \mathbf{w}_j)^T} \tag{4.17}$$

$$p(\mathbf{x}_i) = \sum_{j=1}^{k} p(\mathbf{x}_i/\mathbf{w}_j) \cdot p(\mathbf{w}_j) \tag{4.18}$$

O refinamento é baseado no uso de conhecimento prévio para melhorar, por várias iterações, o agrupamento de um algoritmo de base. Neste caso, o agrupamento realizado é usado como ponto de partida sendo apurado com um algoritmo CBC. Nesse sentido, o mapa final é usado, ou seja, os nodos finais do mapa são considerados ponto de partida para o refinamento. Isto mudará um pouco a cada iteração até o novo algoritmo convergir. Com o agrupamento já realizado, a probabilidade priori $p(\mathbf{w}_j)$ chamada de evidência pode ser calculada. Como um evento anterior já ocorreu, esta probabilidade pode ser calculada com o valor das funções de ativação de cada nodo para cada amostra apresentada no mapa, da forma $p(\mathbf{w}_j) = \frac{(a_w)_j}{\sum (a_w)_j}$. No contexto do modelo proposto, a Inferência Variacional é empregada para viabilizar o cálculo de parâmetros em bases de dados de alta dimensionalidade, com uma discreta perda na acurácia associada (KINGMA; WELLING, 2013).

O Algoritmo 11 ilustra o processo de refinamento. A etapa de treinamento por Inferência Bayesiana é realizado em cada módulo de forma independente. Nesse procedimento, os centros são inicializados com base nos pesos e relevâncias obtidos na primeira etapa. Posteriormente, as probabilidades dos centros são calculadas a partir da matriz de confusão. Em seguida, o grau de pertinência m é calculado conforme a Equação 4.14. Os centróides são atualizados utilizando a Equação 4.16, e a função de custo objetivo é avaliada com base na Equação 3.36. Esse ciclo é repetido por um número predefinido de iterações ou até que não ocorra uma mudança significativa nos valores da função objetivo entre iterações consecutivas. Ao final, cada amostra é atribuída ao nodo que maximiza a função m (RODRÍGUEZ-SOTELO et al., 2012).

Algoritmo 11 Inferência Varicional

Inicialize $\mathbf{w}_j$ grupos, N_{inf} interações, o valor de convergência δ.
Inicialize o vetor de centróides com os centróides da última camada de SC-SOM.
for *todos os nodos dos mapa* $\mathbf{w}_j$ **do**
    Calcular evidência $p(\mathbf{w}_j)$ baseada em matriz de confusão.
    $p(\mathbf{w}_j) = \left(\sum_j True\right)/n_j$.
Calcular a função de peso $\mathbf{w}(\mathbf{x}_i) = 1$.
for *r = 1 até* N_{inf} **do**
    Calcular a função de grau de pertinência através da Equação 4.17.
    Atualizar os centróides atrvés da Equação 4.16.
    Calcular a função objetivo através da Equação 3.36.
    if $\left|f\left(C^{(r)}\right) - f\left(C^{(r-1)}\right)\right|$ **then**
        break.
for *para cada amostra* x_i **do**
    if $l = arg\ max\ m(\mathbf{w}_{j(r)}, \mathbf{x}_i)$ **then**
        Atribuir $\mathbf{x}_i$ *ao centróide* $\mathbf{W}_l^r$

No processo de previsão da categoria a que cada amostra pertence, emprega-se um método semelhante ao treinamento, porém a avaliação do nó vencedor é feita de maneira global. O nó vencedor, globalmente determinado, será responsável por atribuir a categoria à qual a amostra pertence. Dessa forma, é calculada a probablidade gaussiana para cada amostra apresentada pela Equação 4.17. Posteriormente se faz o cálculo da evidência para amostra através da Equação 4.18. Finalmente é calculado o grau de pertinência é dado pela Equação 4.14. O Algoritmo 12 apresenta essa etapa de predição das categorias.

Algoritmo 12 Cálculo do grau de pertinência

Inicialização: probabilidade priori $p(\mathbf{w}_j)$, covariância dos dados $\sum = cov(X)$.
for *toda amostra* $\mathbf{x}_i$ **do**
    Calcular a distribuição Gaussiana através da Equação 4.17.
    Cálcular a evidência através da Equação 4.18.
    Cálcular o grau de pertinência através da Equação 4.14.

5 EXPERIMENTOS

Neste capítulo, são apresentados os experimentos conduzidos para avaliar o desempenho do modelo proposto neste trabalho, comparando-o com modelos selecionados da literatura que utilizam o cálculo da distência por vetores de referência. Diversos experimentos foram conduzidos, abrangendo diferentes tipos de problemas que originam os dados, incluindo mineração de dados, categorização de objetos e categorização de cenas, o principal objetivo. As bases de dados variaram amplamente no número de instâncias, no número de atributos e no número de categorias. É importante analisar a capacidade do MOCE-SOM para escalonar e para responder a diferentes naturezas de dados. O esforço computacional para obter um nível desejado de acurácia também é um ponto bem relevante. Além dos testes de desempenho, foram explorados vários conjuntos de parâmetros de forma meticulosa, a fim de identificar aqueles que resultam no melhor desempenho do algoritmo.

5.1 BASE DE DADOS

O MOCE-SOM com 3 camadas foi validado utilizando conjuntos de dados em aplicações de mineração de dados, categorização de objetos e categorização de cenas. Os números de padrões, atributos, categorias e tipos de padrões dos conjuntos de dados selecionados são apresentados na Tabela 2. A tabela de base de dados foi organizada inicialmente por tipo de conjunto de dados: mineração de dados, categorização de objetos e cenas. O segundo critério de ordenamento foi a quantidade de atributos e por fim a quantidade de categorias. Foram conduzidas dez rodadas de testes com diferentes inicializações aleatórias, e os resultados foram apresentados como médias de desempenho com seus respectivos desvios padrão. O desempenho do MOCE-SOM foi comparado com métodos das famílias k-NN e LVQ.

Os experimentos foram conduzidos com 17 conjuntos de dados (duas bases de dados possuem duas representações distintas, totalizando 19 casos), sendo destas 10 de mineração de dados *Machine Learning Repository* (UCI) (KELLY; LONGJOHN; NOTTINGHAM, 2023). Esses conjuntos de dados abrangem uma diversidade de domínios, sendo essenciais para a validação e comparação de algoritmos de mineração de dados. A clássica base de dados Iris, por exemplo, destaca-se por suas medidas detalhadas das características de diferentes espécies de íris, servindo como um *benchmark* para algoritmos de categorização. A base Wine, que con-

Tabela 2 – Descrição dos conjuntos de dados: número de amostras, atributos, categorias e tipo de problema: Mineração de Dados - MD, Categorização de Objetos - CO e Categorização de Cenas - CC.

Base de dados	Amostras	Atributos	Categorias	Extrator	Tipo
Iris	150	4	3	-	MD
Wine	178	13	3	-	MD
Heart	270	13	2	-	MD
Vehicle	846	18	4	-	MD
Letter	20000	16	26	-	MD
Ionosphere	351	34	2	-	MD
Digits	5.620	64	10	-	MD
USPS	9.298	256	10	-	MD
Motion Tracking	10.299	561	6	-	MD
Isolet	7.797	617	26	-	MD
CIFAR_10	60.000	3.072	10	Vectorizado	CO
		2.880		HoG	CO
CIFAR_100	60.000	3.072	100	Vectorizado	CO
		2.880		HoG	CO
Caltech_101	9.144	4.096	101	VGG_19	CO
Scenes_15	4.885	4.096	15	VGG_19	CC
SUN_RGBD	10.335	4.096	19	VGG_19	CC
Indoors_67	15.620	4.096	67	VGG_19	CC
SUN_397	108.754	4.096	397	VGG_19	CC

tém informações químicas de diferentes tipos de vinhos, contribui para estudos relacionados à análise de qualidade e categorização de bebidas.

No âmbito da saúde, a base de dados Heart é uma importante referência, oferecendo dados clínicos que permitem a previsão de doenças cardíacas. A base de dados Vehicle, por sua vez, concentra-se em características relacionadas a veículos, sendo valiosa para algoritmos de categorização nesse contexto. A base Letter, composta por imagens de letras do alfabeto, amplia a aplicação dos algoritmos para reconhecimento de padrões em caracteres.

Para problemas envolvendo a detecção de padrões em sinais, a base Ionosphere destaca-se ao fornecer dados de radar para distinguir sinais ionosféricos benignos e malignos. As bases Digits e USPS são essenciais para aprimorar algoritmos em reconhecimento de dígitos manuscritos, enquanto a base Motion Tracking, centrada em sequências de imagens, contribui para a análise de algoritmos em cenários dinâmicos. Essas bases permitem um conjunto de experimentos para avaliação da robustez e eficácia dos algoritmos de mineração de dados em

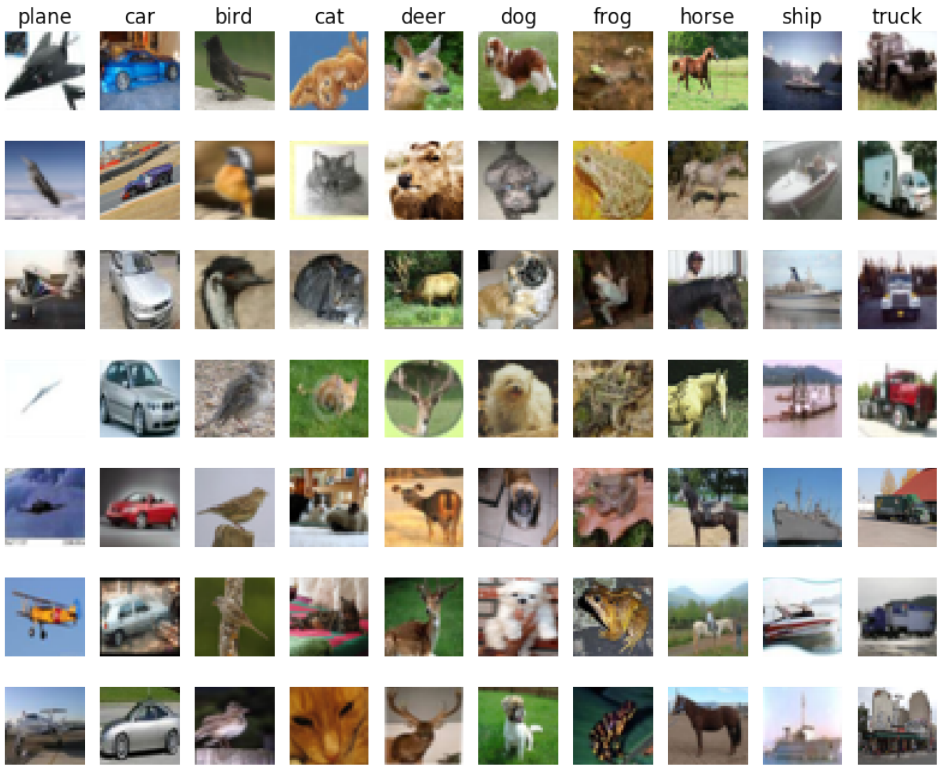

Figura 11 – Imagens de exemplo do conjunto de dados CIFAR_10 (KRIZHEVSKY; HINTON et al., 2009).

uma variedade de domínios.

As bases de dados para categorização de objetos, CIFAR_10 (KRIZHEVSKY; HINTON et al., 2009), CIFAR_100 (KRIZHEVSKY; HINTON et al., 2009), e Caltech_101 (FEI-FEI; FERGUS; PERONA, 2004) formam conjuntos de dados desafiadores e diversificados para avaliar a capacidade de algoritmos em reconhecer e categorizar diferentes objetos visuais. As imagens do CIFAR_10 podem ser vistas na Figura 11. O CIFAR_10 e CIFAR_100 consistem em imagens de 32x32 pixels, divididas em 10 e 100 classes, respectivamente, abrangendo uma variedade de objetos cotidianos. Essas bases são amplamente utilizadas para treinar e testar modelos de aprendizado profundo em tarefas de categorização de imagens. Os experimentos para as bases de dados CIFARs foram executados com duas formas de representação das amostras empregadas na literatura: Vetorizado e HoG (DALAL; TRIGGS, 2005). Para vetorização, as imagens 2D foram transformadas em um vetor unidimensional concatenando cada linha da imagem para cada um dos 3 canais RGB: 3x32x32 = 3.072. Para a representação em HoG, foram extraídos com dois tamanhos de janela, [4 4] e [2 2], gerando vetores de 576 e 2.304 posições, respectivamente, resultando em um vetor completo concatenado de 2.880 dimensões.

Por sua vez, a base Caltech_101 é reconhecida por sua diversidade e complexidade, contendo imagens de alta resolução de objetos distribuídos em 101 categorias. Imagens exemplo

Figura 12 – Imagens de exemplo do conjunto de dados Caltech_101 (FEI-FEI; FERGUS; PERONA, 2004).

da Caltech_101 podem ser vistas na Figura 12. Essa base apresenta desafios significativos, demandando que os algoritmos de categorização de objetos sejam capazes de lidar com variações nas condições de iluminação, pose e fundo. A diversidade presente nessas bases de dados permite uma avaliação abrangente do desempenho de algoritmos em cenários do mundo real, impulsionando avanços na área de visão computacional e categorização de objetos. A representação das imagens do Caltech_101 e todas as bases de dados de cenas utiliza o modelo VGG_19. A extração de características usando a rede VGG_19 implica na utilização de uma arquitetura de CNN. Ao aplicar essa rede a uma imagem de entrada, o vetor de características gerado possui 4.096 dimensões, representando uma codificação densa e rica em informações. Cada dimensão desse vetor captura diferentes níveis de abstração e hierarquia de características presentes na imagem, permitindo uma representação compacta e semântica das informações visuais.

O dataset Scenes_15 (LAZEBNIK; SCHMID; PONCE, 2006) é um conjunto de dados amplamente utilizado na pesquisa de reconhecimento de cenas e categorização de imagens. Ele contém imagens de 15 categorias distintas de cenas naturais e artificiais, como praias, florestas, ruas urbanas e interiores de casas. Na Figura 13 pode ser visto imagens do dataset 15_Scenes. Cada categoria inclui um número significativo de imagens capturadas em diferentes condições de iluminação, perspectivas e composições, o que torna o conjunto de dados um desafio interessante para algoritmos de visão computacional e aprendizado de máquina. A diversidade e a riqueza das cenas representadas fornecem uma base robusta para a avaliação de métodos de extração de características e categorização de imagens.

SUN_RGBD (SONG; LICHTENBERG; XIAO, 2015) consiste em 10.335 imagens RGB-D com

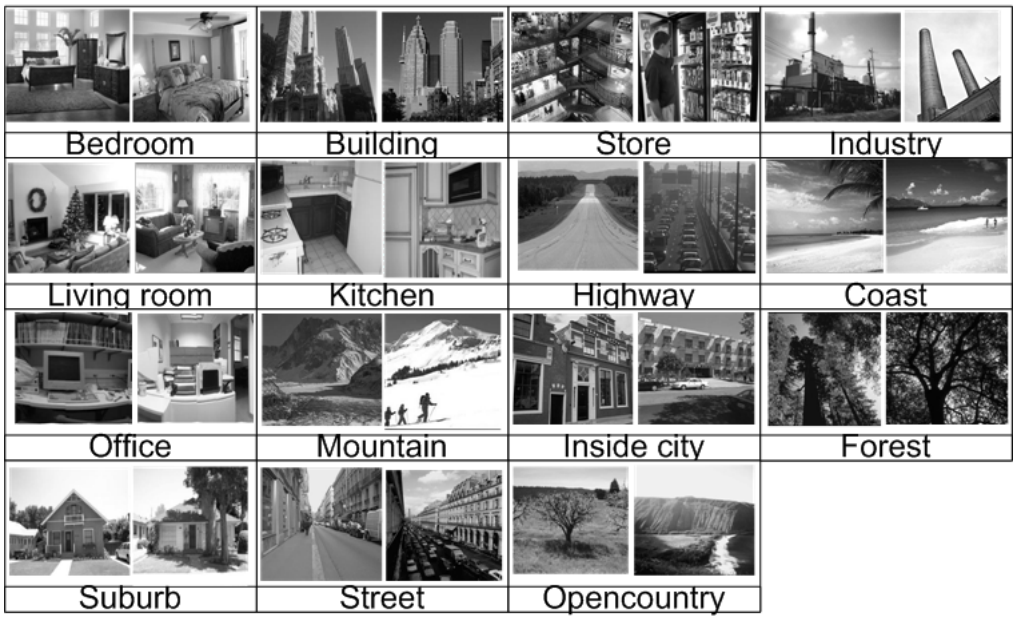

Figura 13 – Imagens de exemplo do conjunto de dados Scenes_15 (LAZEBNIK; SCHMID; PONCE, 2006).

anotações em 2D e 3D, para ambos os objetos e cenas. Desenvolvida para abordar os desafios da compreensão de cenas em ambientes 3D, o SUN_RGBD contém uma ampla gama de categorias, permitindo a exploração de métodos de aprendizado profundo em tarefas de categorização e compreensão de cenas complexas. Somente a informação RGB foi utlizado para representação das amostras.

O conjunto de dados Indoors_67 (QUATTONI; TORRALBA, 2009) se destaca por seu enfoque em ambientes internos. Com 67 categorias distintas, cada uma com aproximadamente 100 imagens, cada uma delas com resolução mínima de 200×200 pixels, esse conjunto oferece uma variedade de desafios relacionados à categorização de cenas em espaços fechados. Exemplos dessas imagens e categorias podem ser vistas na Figura 14. Isso o torna valioso para pesquisas que visam compreender o reconhecimento de ambientes internos por meio de algoritmos de visão computacional. Devido às semelhanças entre os objetos neste conjunto de dados, a categorização das imagens é desafiadora. Os conjuntos de treinamento e teste têm 80% e 20% do total de imagens.

SUN_397 (XIAO et al., 2010) é dividida em 397 categorias de cenas (175 internas, 220 externas e 2 misturadas), cada uma com mais que 100 imagens. O conjunto de dados contém 108.754 imagens com um tamanho de 500 × 300 pixels. Exemplos dessas imagens e categorias são apresentadas pela Figura 15. As várias configurações de divisão de treinamento/teste,

Figura 14 – Imagens de exemplo do conjunto de dados Indoors_67 (QUATTONI; TORRALBA, 2009).

no qual o particionamento mais utilizado é com 50 imagens de treinamento por categoria. Com uma ampla diversidade de cenas capturadas em diferentes condições de iluminação e ambientes, o SUN_397 é uma escolha robusta para avaliações abrangentes de algoritmos de categorização de cenas em ambientes variados.

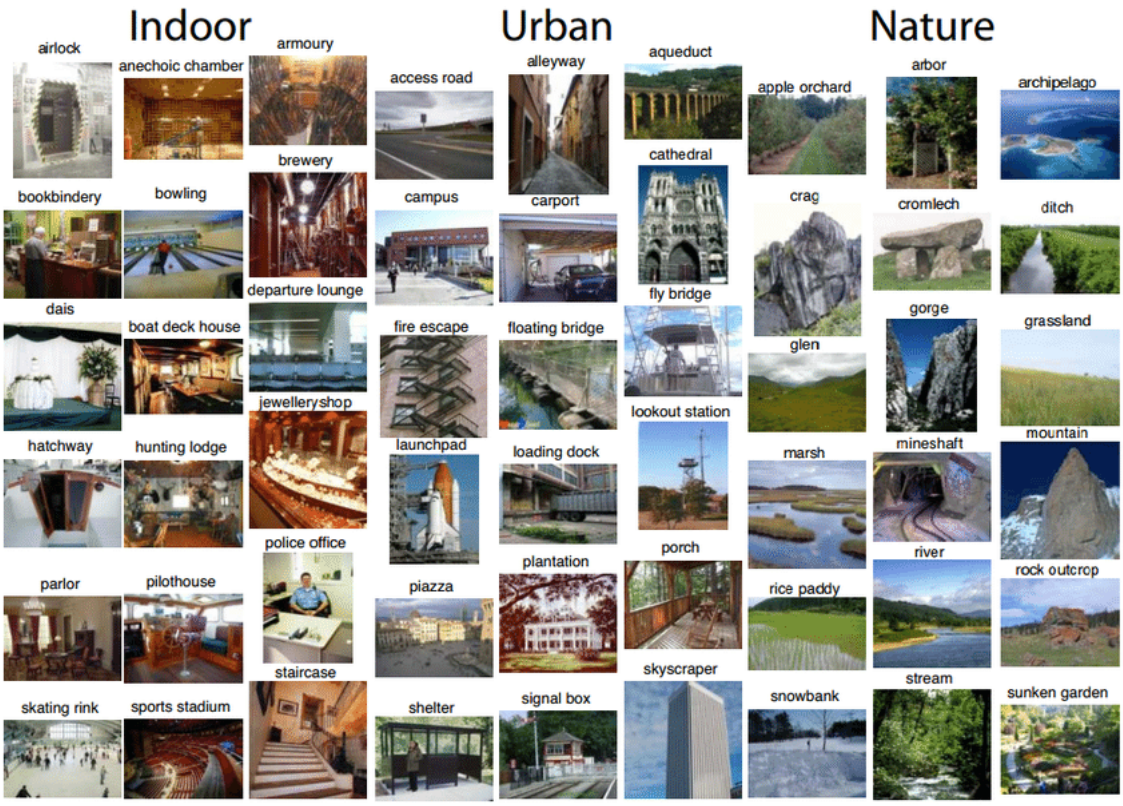

Figura 15 – Imagens de exemplo do conjunto de dados Indoors_67 (XIAO et al., 2010).

5.2 METODOLOGIA DOS EXPERIMENTOS

Os experimentos foram conduzidos em um computador com a seguinte configuração: processador Intel Xeon Silver 4310 de 2,1 GHz com 128 GB de RAM. O tempo de processamento de execução de todos os algoritmos foi medido sem paralelização do algoritmo para garantir uma comparação justa, sendo executados em um único processo desde o início até o final do treinamento. Os algoritmos foram executados utilizando todos os atributos do conjunto de dados, exceto quando foram testados o desempenho dos modelos GMLVQ e LGMLVQ, para os conjuntos de testes com 4.096 atributos (CIFAR_10, CIFAR_100 Caltech_101, Scenes_15, Indoors_67 e SUN_397). Nestes casos, os dois algoritmos mencionados se mostraram impraticáveis de serem executados nesses conjuntos de dados devido à sua necessidade de memória RAM que escala na ordem de magnitude de $O(n^2)$ em relação ao número de atributos. Esses algoritmos oferecem uma opção de redução de dimensionalidade durante o treinamento. Portanto, essa redução de dimensionalidade foi aplicada apenas nesses casos para viabilizar a execução dos experimentos.

Para todos os conjuntos de dados e para todos os métodos, foi realizada a normalização dos dados de entrada para cada atributo, garantindo que todos os valores de entrada para todos

os métodos estejam dentro do intervalo [0, 1]. Os experimentos foram executados seguindo as partições de treinamento e teste conforme previamente descrito na literatura. O MOCE-SOM utilizado nos experimentos implementou 3 camadas: as duas primeiras com SC-SOM seguindo de composição e a última camada com um SC-SOM seguido de Inferência Variacional.

Com o propósito de avaliar a significância estatística dos resultados experimentais, foram conduzidas múltiplas rodadas de experimentos, utilizando diferentes inicializações aleatórias de parâmetros. Os desempenhos médios juntamente com seus desvios padrão foram calculados para cada método. Para investigar se existem diferenças estatisticamente significativas entre os resultados obtidos pelo MOCE-SOM e os algoritmos de comparação, foi empregado o teste de hipótese U de Mann-Whitney para cada par de categorizadores (MANN; WHITNEY, 1947). Este teste estatístico não paramétrico possibilita a ordenação do conjunto de experimentos e a atribuição de valores às observações, permitindo determinar qual é estatisticamente maior. A base desse teste reside na comparação entre cada observação em uma amostra $\mathbf{x}_i$ e cada observação na segunda amostra $\mathbf{y}_j$. Caso as amostras compartilhem a mesma mediana, cada observação possui uma probabilidade de 0.5 de ser maior ou menor que a observação correspondente na outra amostra. Sendo assim, as hipóteses levantadas são:

$$\begin{aligned} H_0 &: P(\boldsymbol{x}_i > \boldsymbol{y}_j) = 1/2 \\ H_1 &: P(\boldsymbol{x}_i > \boldsymbol{y}_j) \neq 1/2 \end{aligned} \tag{5.1}$$

Os algoritmos tiveram os seus parâmetros ajustados usando Latin Hyper-cube Sampling (LHS) independente para cada base de dados (CHATTERJEE; HADI, 2009). Em seguida, os algoritmos são executados para configurações distintas de parâmetros, todas dentro dos seus intervalos de validade que podem ser encontradros no trabalhos originais ou adicionais para os métodos: k-NN (COVER; HART, 1967; BATISTA; SILVA et al., 2009; GÖPFERT; WERSING; HAMMER, 2022), LMNN (DOMENICONI; GUNOPULOS; PENG, 2005; GÖPFERT; WERSING; HAMMER, 2022), LANN (GÖPFERT; WERSING; HAMMER, 2022), LVQ (KOHONEN, 2012; NOVA; ESTÉVEZ, 2014), GLVQ (SATO; YAMADA, 1995; NOVA; ESTÉVEZ, 2014), GRLVQ (HAMMER; VILLMANN, 2002; NOVA; ESTÉVEZ, 2014), GMLVQ (SCHNEIDER; BIEHL; HAMMER, 2009a; NOVA; ESTÉVEZ, 2014) e GMLVQ (SCHNEIDER; BIEHL; HAMMER, 2009a; NOVA; ESTÉVEZ, 2014). A quantidade de protótipos foi o mesmo para cada categoria nos diferentes métodos utilizados, o intervalo de variação foi de 5 à 10 protótipos independente para cada modelo. A tabela 3 apresentar o valores dos parâmetros utilizados no LHS.

Tabela 3 – Tabela com os intervalos de valores de parametrização para o MOCE-SOM.

Base de dados	N° de nodos	N	η_0	η_f	λ_0	λ_f	h_0	ρ	N_{inf}	δ
Iris	[5 10]	[10 500]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Wine	[5 10]	[10 500]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Heart	[5 10]	[10 500]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Vehicle	[5 10]	[10 500]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Letter	[5 10]	[10 500]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Ionosphere	[5 10]	[10 500]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Digits	[5 10]	[10 100]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
USPS	[5 10]	[10 100]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Motion Tracking	[5 10]	[10 100]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
Isolet	[5 10]	[10 100]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 50]	$10^{-4}[10,001]$
CIFAR_10 Vectorized	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
CIFAR_10 HoG	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
CIFAR_100 Vectorized	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
CIFAR_100 HoG	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
Caltech_101	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
Scenes_15	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
SUN_RGBD	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
Indoors_67	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$
SUN_397	[5 10]	[10 50]	[0,1 0,01]	[0,01 0,0001]	[0,1 0,9]	0,01	[3 9]	[0,01 0,1]	[10 20]	$10^{-4}[10,001]$

5.3 RESULTADOS

O MOCE-SOM é comparado com diversos algoritmos da literatura: k-NN (COVER; HART, 1967), LMNN (DOMENICONI; GUNOPULOS; PENG, 2005), LANN (GÖPFERT; WERSING; HAMMER, 2022), LVQ (KOHONEN, 2012), GLVQ (SATO; YAMADA, 1995), GRLVQ (HAMMER; VILLMANN, 2002), GMLVQ (SCHNEIDER; BIEHL; HAMMER, 2009a) e GMLVQ (SCHNEIDER; BIEHL; HAMMER, 2009a). A Tabela 4 apresenta os resultados médios e desvios padrão da acurácia para todos os métodos nos conjuntos de dados de mineração de dados, categorização de objetos e categorização de cenas. A Tabela 5 apresenta o melhor resultado dos métodos para cada base de dados. A Tabela 6 fornece os tempos de treinamento para os métodos na melhor configuração de cada método para cada base de dados.

Os valores nas tabelas representam os resultados médios e seus desvios padrão, no qual os números em negrito indicam os melhores resultados para cada conjunto de dados. O resumo ao final de cada tabela analisa a quantidade de vezes que o método venceu, perdeu ou está estatisticamente empatado ou igual. O sinal de "+"é atribuído ao método que vence o MOCE-SOM, -"para o método que perder do MOCE-SOM e "≈"para os resultados que são estatísticamente equivalentes ou igual, assim a comparação é expressa por "$+/-/\approx$". Nas

próximas subseções, os resultados são comentados para cada grupo de bases de dados, além de apresentar análise dos resultados obtidos em cada camada do MOCE-SOM.

5.3.1 Mineração de Dados

Os conjuntos de dados de mineração de dados apresentam o menor nível de complexidade, dentre todos os grupos. Os resultados de acurácia média são apresentado na Figura 16 e o tempo de treinamento na Figura 17. Metade dessas bases contém menos de mil amostras, a quantidade máxima de categorias é 26, e a dimensão dos atributos não ultrapassa 700. Os métodos da família k-NN, em geral, apresentam resultados inferiores ao MOCE-SOM, com algumas exceções. O k-NN perde em 8 ocasiões e empata em 2. O LMNN alcança resultados comparáveis, com uma análise de $3/3/4$, indicando equivalência nos resultados. O LANN apresenta um comparativo de $1/3/5$, indicando uma leve vantagem do modelo proposto sobre ele. Na avaliação do tempo de execução, o k-NN é consistentemente mais rápido, principalmente para bases menores. O LMNN tem resultados de tempo de processamento um pouco mais rápidos que o modelo proposto, ganhando na comparação por $7/3/0$, enquanto o LANN perde para o nosso modelo na análise de tempo.

A família LVQ apresenta desempenhos variados em comparação ao modelo proposto. O LVQ e o GRLVQ perdem em todos os casos. O GMLVQ apresenta resultados ligeiramente piores que os nossos, com uma contagem de $2/4/4$, e o LGMLVQ supera o MOCE-SOM para esses conjuntos de dados. Na avaliação de tempo, o LVQ perde para o MOCE-SOM em todas as instâncias, enquanto o GLVQ vence o modelo proposto em todas as comparações. O GRLVQ tem um bom desempenho de tempo, ficando à frente do modelo proposto. Já o GMLVQ e o LGMLVQ ficam empatados na comparação de tempo.

Em resumo, para as bases de mineração de dados, o LGMLVQ obteve uma acurácia superior ao MOCE-SOM e este empata em desempenho com o LMNN. Na análise de tempo, ocorre empate com o LGMLVQ e derrota para o LMNN.

5.3.2 Categorização de Objetos

Os conjuntos de dados de categorização de objetos são mais complexos quando comparados aos conjuntos de mineração de dados. Os resultados de acurácia média são apresentado na Figura 18 e o tempo de treinamento na Figura 19. As bases de dados de objetos apresentam

Tabela 4 – Tabela com os resultados para todas as bases de dados comparando os métodos. São apresentados todas as médias e o desvios padrões de acurácia. Comparação ao final da tabela: $+/-/\approx$.

Base de dados	KNN	LMNN	LANN	LVQ	GLVQ	GRLVQ	GMLVQ	LGMLVQ	MOCE-SOM
Iris	94,59 (0,27) -	96,49 (2,33) -	95,41 (2,37) -	94,05 (2,48) -	**98,67 (2,61)** $\approx$	95,41 (2,86) -	97,30 (2,55) $\approx$	96,22 (2,28) -	98,38 (1,40)
Wine	97,05 (2,41) $\approx$	98,86 (1,61) $\approx$	97,50 (2,50) $\approx$	97,05 (3,40) $\approx$	98,41 (3,91) $\approx$	98,86 (1,61) $\approx$	**99,55 (0,96)** $\approx$	99,32 (1,10) $\approx$	99,09 (1,17)
Heart	80,00 (3,16) -	80,15 (3,15) -	79,10 (5,07) -	83,13 (4,22) $\approx$	82,28 (3,85) $\approx$	**84,03 (4,45) +**	83,88 (4,33) $\approx$	83,28 (3,36) $\approx$	83,97 (3,02)
Vehicle	71,47 (1,73) -	79,19 (2,29) +	72,61 (2,94) $\approx$	59,43 (2,71) -	70,19 (2,91) -	72,56 (2,15) $\approx$	83,03 (2,47) +	**84,50 (1,90) +**	74,02 (2,65)
Letter	74,75 (2,86) -	83,30 (3,17) $\approx$	81,90 (3,27) $\approx$	57,50 (4,09) -	74,15 (2,93) -	74,45 (2,35) -	76,30 (3,81) -	**86,90 (2,20) +**	81,54 (2,31)
Ionosphere	87,33 (7,59) -	90,23 (4,03) -	86,14 (7,61) -	85,45 (4,68) -	87,95 (5,36) -	89,09 (5,55) -	92,27 (5,89) -	94,55 (5,38) -	**96,65 (1,59)**
Digits	97,94 (7,59) $\approx$	**98,72 (0,01) +**	98,22 (0,01) +	82,00 (1,61) -	94,41 (0,26) -	93,99 (0,51) -	96,17 (0,31) -	98,40 (0,01) +	97,83(0,14)
USPS	94,00 (0,26) -	95,12 (0,01) $\approx$	95,12 (0,01) $\approx$	87,89 (0,63) -	88,24 (0,39) -	88,25 (0,40) -	91,99 (0,15) -	**95,52 (0,16) +**	94,85 (0,32)
Motion Tracking	89,99 (0,63) -	**96,78 (0,01) +**	91,01 (0,01) $\approx$	75,82 (0,89) -	87,71 (0,05) -	89,72 (0,10) -	95,66 (0,18) +	95,15 (0,22) +	91,13 (0,20)
Isolet	92,17 (0,01) -	95,83 (0,01) $\approx$	92,88 (0,01) -	70,13 (1,76) -	92,89 (0,01) -	93,89 (0,20) -	95,72 (0,17) $\approx$	**96,47 (0,18) +**	95,62 (0,14)
CIFAR_10 Vectorized	35,39 (0,01) +	35,12 (0,93) $\approx$	35,60 (0,45) +	26,32 (0,98) -	28,21 (0,94) -	31,32 (0,76) -	33,35 (1,12) $\approx$	**35,86 (0,47) +**	34,24 (0,36)
CIFAR_10 HoG	50,46 (0,01) -	**57,40 (0,72) +**	50,06 (0,01) -	48,16 (0,23) -	24,11 (1,15) -	33,26 (1,50) -	13,73 (0,89) -	38,61 (1,49) -	51,27 (0,23)
CIFAR_100 Vectorized	**17,55 (0,01) +**	16,98 (0,18) +	16,41 (0,17) +	1,45 (0,28) -	9,43 (0,24) -	10,32 (0,25) -	7,80 (0,75) -	8,09 (0,16) -	15,83 (0,24)
CIFAR_100 HoG	22,29 (0,01) -	**22,40 (0,15)** $\approx$	22,05 (0,10) -	2,73 (0,37) -	6,30 (0,23) -	5,88 (0,26) -	7,81 (1,02) -	8,57 (0,20) -	22,38 (0,02)
Caltech_101	80,61 (1,02) -	89,83 (0,36) -	82,13 (0,43) -	46,24 (5,08) -	77,69 (2,13) -	85,92 (1,06) -	91,16 (0,86) -	77,16 (1,97) -	**93,02 (0,18)**
Scenes_15	73,79 (1,05) -	88,82 (1,15) -	76,28 (1,08) -	82,82 (0,90) -	84,69 (1,23) -	85,47 (2,17) -	90,16 (0,36) -	86,00 (3,09) -	**91,23 (1,13)**
SUN_RGBD	45,22 (0,66) $\approx$	**51,96 (0,02) +**	44,60 (0,01) $\approx$	45,59 (0,20) $\approx$	45,76 (1,36) $\approx$	45,61 (0,55) $\approx$	50,43 (0,44) -	46,30 (0,60) -	45,42 (0,40)
Indoors_67	56,57 (0,01) -	60,67 (0,01) -	57,46 (0,01) -	52,20 (0,37) -	56,14 (2,70) -	59,23 (1,81) -	60,72 (0,53) -	59,48 (0,80) -	**64,91 (0,33)**
SUN_397	37,98 (0,09) -	39,98 (0,67) -	39,23 (0,10) -	6,53 (0,70) -	34,08 (0,15) -	34,99 (1,08) -	1,49 (0,01) -	0,67 (0,11) -	**47,42 (0,13)**
	2/14/3	6/7/6	3/10/6	0/16/3	0/15/4	1/15/3	2/12/5	7/10/2	

Tabela 5 – Tabela com os resultados para todas as bases de dados comparando os métodos. São apresentados melhor acurácia para cada modelo. Comparação ao final da tabela: $+/-/\approx$.

Base de dados	KNN	LMNN	LANN	LVQ	GLVQ	GRLVQ	GMLVQ	LGMLVQ	MOCE-SOM
Iris	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00**
Wine	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00** $\approx$	**100,00**
Heart	85,07 $\approx$	85,07 $\approx$	85,07 $\approx$	91,04 +	91,04 +	91,04 +	**94,03 +**	**94,03 +**	86,57
Vehicle	74,88 -	81,99 +	77,25 -	65,88 -	76,30 -	78,20 -	88,15 +	**90,05 +**	81,50
Letter	83,00 -	87,50 -	86,00 +	63,00 -	77,50 -	80,50 -	83,00 -	**92,50 +**	85,60
Ionosphere	97,14 -	**100,00** $\approx$	**100,00** $\approx$	95,45 -	95,45 -	97,73 -	97,73 -	100,00 $\approx$	**100,00**
Digits	98,64 +	98,39 +	98,22 +	83,14 -	94,56 -	94,88 -	96,33 -	**98,78 +**	98,11
USPS	95,12 -	95,12 -	95,12 -	88,34 -	88,95 -	88,64 -	92,20 -	**95,57 +**	95,20
Motion Tracking	90,74 -	**96,78 +**	91,01 -	77,47 -	87,73 -	89,74 -	95,68 -	95,45 +	91,31
Isolet	92,17 -	95,83 +	92,88 -	72,10 -	92,99 -	94,23 -	95,83 +	**96,50 +**	94,56
CIFAR_10 Vectorized	35,40 -	37,75 -	38,00 -	28,24 -	29,90 -	33,46 -	35,16 -	36,94 -	**38,76**
CIFAR_10 HoG	50,46 -	**57,50 +**	50,06 -	48,32 -	25,70 -	37,12 -	14,68 -	40,14 -	51,96
CIFAR_100 Vectorized	17,56 +	**17,05 +**	16,54 +	1,52 -	9,84 -	10,76 -	8,33 -	8,20 -	16,34
CIFAR_100 HoG	22,30 -	22,45 -	22,12 -	2,98 -	6,66 -	6,22 -	8,52 -	8,70 -	**22,62**
Caltech_101	83,26 -	89,87 -	82,56 -	50,35 -	80,17 -	87,18 -	92,08 -	78,86 -	**93,60**
Scenes_15	81,70 -	89,09 -	82,12 -	83,98 -	85,33 -	86,90 -	90,34 -	87,94 -	**92,42**
SUN_RGBD	45,60 +	**51,97 +**	44,60 -	45,62 +	46,90 +	46,32 +	50,80 +	47,56 +	45,55
Indoors_67	56,57 -	60,67 -	57,46 -	52,46 -	59,78 -	60,82 -	60,92 -	59,60 -	**65,11**
SUN_397	38,20 -	40,29 -	39,30 -	6,60 -	35,75 -	35,87 -	1,51 -	0,75 -	**47,50**
	3/13/3	7/8/4	3/12/4	2/15/2	2/15/2	2/15/2	4/13/2	8/8/3	

Tabela 6 – Tabela com os resultados para todas as bases de dados comparando os métodos. São apresentados todas as médias e desvios padrões dos tempos treinamento (em segundos) para os melhores resultados. Comparação ao final da tabela: $+/-/\approx$.

Bases de dados	KNN	LMNN	LANN	LVQ	GLVQ	GRLVQ	GMLVQ	LGMLVQ	MOCE-SOM
Iris	**0,04 (0,01) +**	0,35 (0,04) +	7,39 (0,41) -	28,21 (2,49) -	0,22 (0,05) +	0,25 (0,10) +	0,19 (0,05) +	0,10 (0,04) +	2,56 (0,52)
Wine	**0,005 (0,002) +**	0,08 (0,001) +	3,67 (0,06) +	18,71 (0,37) -	0,11 (0,06) +	0,29 (0,10) +	1,32 (0,18) +	0,24 (0,08) +	2,61 (0,35)
Heart	**0,005 (0,001) +**	2,58 (0,25) -	2,81 (0,06) -	40,01 (3,42) -	0,10 (0,01) +	0,23 (0,13) +	0,16 (0,03) +	0,07 (0,01) +	0,91 (0,06)
Vehicle	**0,007 (0,001) +**	25,33 (5,98) -	28,06 (0,33) -	87,36 (0,40) -	1,73 (0,33) +	2,63 (0,67) $\approx$	16,02 (1,97) -	10,3 (1,42) +	3,17 (0,09)
Letter	**0,12 (0,02) +**	4,69 (0,24) +	73,45 (6,81) -	314 (23) -	13,91 (0,58) +	6,13 (0,40) +	91,45 (13,03) -	51,02 (10,11) -	34,62 (0,86)
Ionosphere	**0,04 (0,01) +**	1,48 (0,14) +	8,90 (1,80) +	111,96 (3,4) -	0,51 (0,14) +	0,32 (0,04) +	1,19 (0,41) +	0,81 (0,04) +	14,21 (0,99)
Digits	**0,33 (0,06) +**	154 (7) +	579 (36) -	930 (156) -	6,40 (4,02) +	11,68 (0,71) +	65,97 (4,26) +	383 (25) -	282,45 (12,51)
USPS	**5,97 (0,45) +**	24,27 (1,98) +	1.512 (76) -	2.841 (238) -	13,31 (2,64) +	24,56 (2,61) +	1.832 (84) -	1.092 (237) -	281,56 (6,21)
Motion Tracking	**16,64 (0,88) +**	168,12 (9,46) -	378 (17) -	2.324 (186) -	23,13 (2,15) +	182 (13) -	1.328 (312) -	696 (58) -	124,19 (3,89)
Isolet	**11,42 (1,06) +**	72,89 (21,45) +	582 (48) -	2.376 (136) -	13,51 (1,99) +	323 (254) +	3.061 (412) -	1.532 (845) -	420,69 (10,78)
CIFAR_10 Vectorized	1.934 (386) +	1.003 (298) +	1.203 (189) +	5.998 (142) +	**154,03 (31,56) +**	4.742 (238) +	22.944 (2.398) +	33.156 (3.182) -	28.091 (956)
CIFAR_10 HoG	2.130 (428) +	1.014 (235) +	1.358 (224) +	5.010 (161) +	**141,45 (13,20) +**	4.920 (341) +	26.182 (2.418) -	36.864 (4.210) -	24.762 (1.123)
CIFAR_100 Vectorized	4.320 (382) +	1.403 (428) +	1.567 (234) +	3.690 (938) +	**316 (32) +**	28.247 (974) +	24.516 (1.956) +	39.704 (1.557) -	31.323 (1.281)
CIFAR_100 HoG	5.112 (890) +	1.608 (431) +	2.058 (381) +	9.660 (438) +	**636 (7) +**	8.345 (294)+	85.050 (1.239) -	28.502 (1.278) $\approx$	28.567 (1.239)
Caltech_101	**42,25 (6,36) +**	979 (52) +	366 (42) +	3.851 (297) +	43,22 (11,15) +	21.421 (1.980) -	8.198 (342) -	95.892 (5.592) -	6.872 (57)
Scenes_15	**7,74 (1,31) +**	720 (42) -	3.881 (123) -	162 (24) +	60,23 (4,98) +	460 (391) -	660,45 (51,62) -	1.402 (259) -	399,02 (9,81)
SUN_RGBD	**20,52 (5,27) +**	480 (84) +	761 (44) -	10.264 (1.239) -	24,56 (1,92) +	4.494 (315) -	2.923 (189) -	4.361 (312) +	662,22 (49,21)
Indoors_67	63,50 (4,17) +	728 (34) +	15.356 (2.398) -	9.068 (593) -	**48,87 (9,22) +**	1.032 (83) +	6.203 (292) -	2.045 (417) +	2.871 (45)
SUN_397	1.172 (14) +	126.436 (8.565) -	115.492 (21.341) -	174.381 (17.802) -	**492 (45) +**	12.973 (1782) +	17.422 (945) +	8.237 (210) +	78.345 (3.456)
	19/0/0	14/5/0	7/12/0	6/13/0	19/0/0	11/2/1	8/11/0	8/10/1	

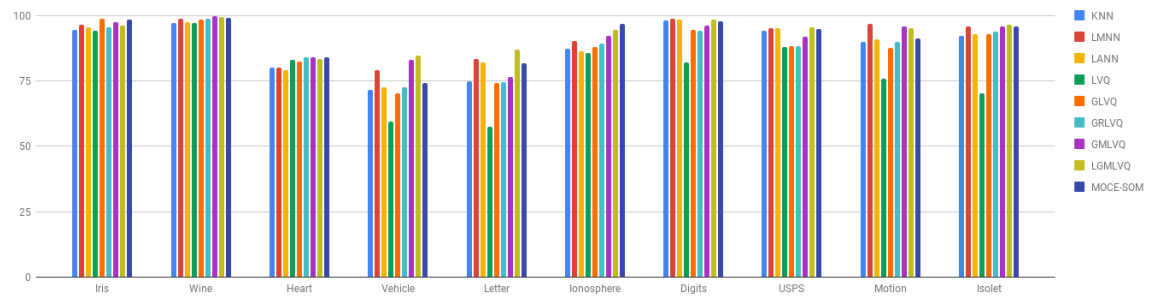

Figura 16 – Acurácia média para os conjuntos de mineração de dados.

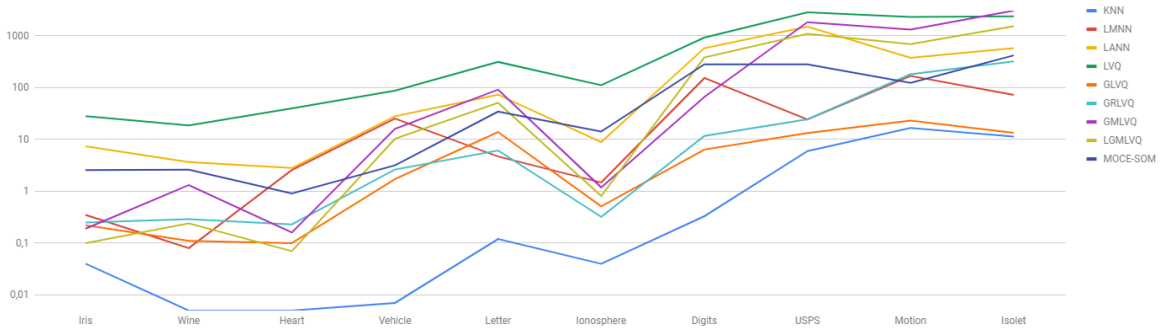

Figura 17 – Tempo de treinamento em escala logarítmica para mineração de dados.

alto número de amostras, chegando a 60.000, a quantidade de categorias pode atingir 100, e a dimensão dos atributos varia de 3.072 até 4.096. Os métodos da família k-NN, em geral, produzem resultados semelhantes ao MOCE-SOM, sendo eficazes para lidar com uma grande quantidade de amostras. O k-NN perde em 3 ocasiões e vence em 2. O LMNN obtém um comparativo de $1/2/2$, indicando que, nesse caso, nosso modelo foi ligeiramente superior. Em relação ao LANN, o comparativo é $2/3/0$, demonstrando uma pequena vantagem do MOCE-SOM. Avaliando os tempos para toda a família do k-NN, eles são consistentemente mais rápidos que o MOCE-SOM. Nesse contexto, para a categorização de objetos, o MOCE-SOM obtém acurácia superior aos modelos da família k-NN em todos os casos.

Os modelos baseados em LVQ não apresentam bons resultados para as bases de dados de categorização de objetos em comparação com o MOCE-SOM. O LVQ, GLVQ, GRLVQ, e GMLVQ perdem em todos os casos de comparação de acurácia. O LGMLVQ vence em apenas um dos conjuntos de dados, obtendo um comparativo de $1/4/0$. Na avaliação de tempo, o LVQ, GLVQ, e GRLVQ são sempre mais rápidos que o MOCE-SOM. Este é mais rápido que o GMLVQ e LGMLVQ, com comparações de $2/3/0$ e $0/4/1$, respectivamente. Isso sugere que o MOCE-SOM é superior aos métodos comparados da família LVQ. Os métodos GMLVQ e LGMLVQ sofrem tanto na queda da acurácia quanto no tempo. Portanto, para as bases de categorização de objetos, o MOCE-SOM obteve resultados superiores a todos os métodos

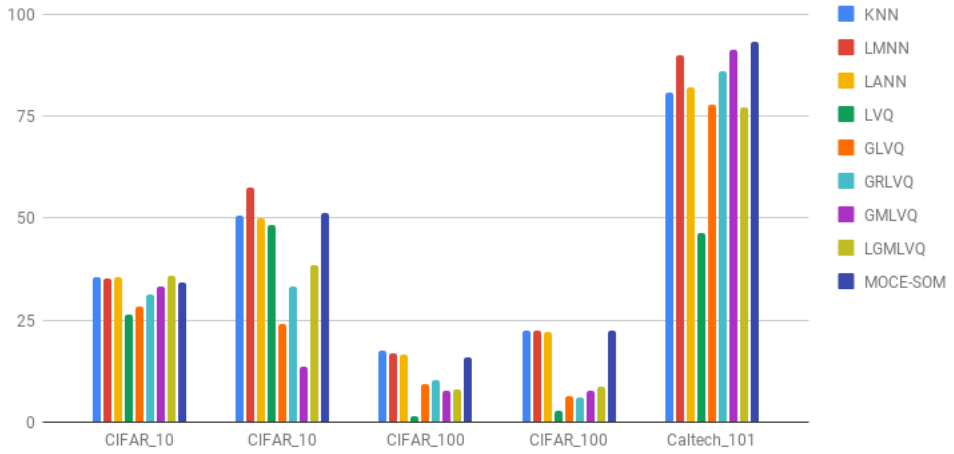

Figura 18 – Acurácia média para os conjuntos de categorização de objetos.

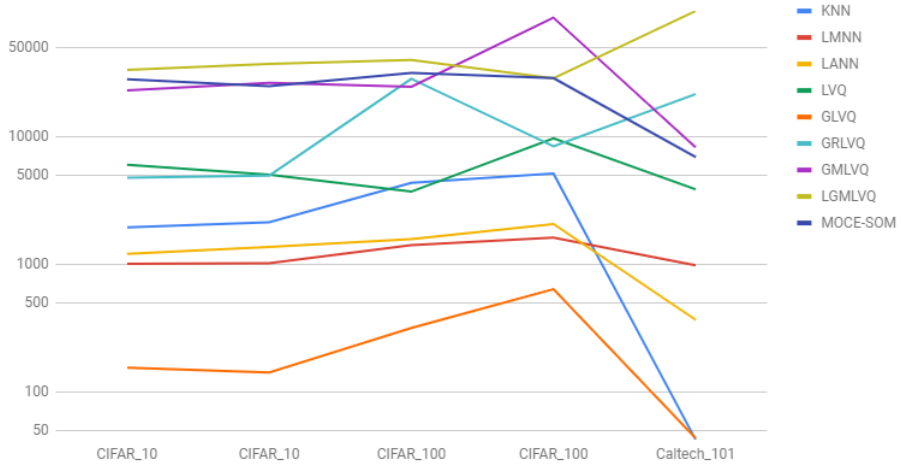

Figura 19 – Tempo de treinamento em escala logarítmica para categorização de objetos.

comparados, embora, na avaliação do tempo, foi mais rápido apenas que o GMLVQ e o LGMLVQ.

5.3.3 Categorização de Cenas

Os conjuntos de dados de categorização de cenas são os mais complexos dos três grupos. Os resultados de acurácia média são apresentado na Figura 20 e o tempo de treinamento na Figura 21. Essas bases de dados apresentam um número substancial de amostras, alcançando centenas de milhares, a quantidade de categorias pode atingir 397, e a dimensão dos atributos é de 4.096. A combinação de uma quantidade significativa de amostras e alta dimensionalidade pode apresentar muitos desafios para os algoritmos, que precisam estimar as relevâncias dos atributos. Os métodos da família k-NN, em sua maioria, resultam em desempenho inferior ao

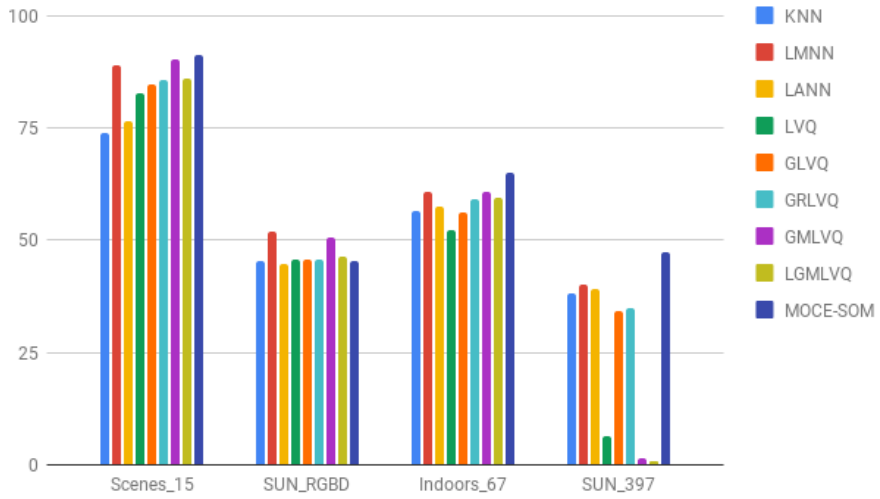

Figura 20 – Acurácia média para os conjuntos de categorização de cenas.

MOCE-SOM, perdendo em todos os comparativos. O k-NN perde em 3 ocasiões e empata em 1. O LMNN obtém um comparativo de $1/3/0$, demonstrando que nosso modelo escala melhor do que o LMNN em situações mais desafiadoras. Em relação ao LANN, o comparativo é $0/3/1$, indicando que o MOCE-SOM alcança melhores resultados em comparação com métodos da família k-NN para casos desafiadores, como a categorização de cenas. Avaliando os tempos para toda a família do k-NN, o k-NN é consistentemente melhor do que o MOCE-SOM, o LMNN empata, e o LANN perde. Esses resultados ressaltam o desempenho do modelo proposto quando os problemas escalam.

Já para os modelos baseados no LVQ apresentam uma degradação nos resultados para as bases de dados de categorização de cenas, ampliando a distância do desempenho para o MOCE-SOM. O LVQ, GLVQ, GRLVQ, GMLVQ, LGMLVQ perdem em todas as situações de comparação de acurácia. No caso da SUN_397, que é a base mais complexa de todas, os resultados do GMLVQ e LGMLVQ sofrem uma queda importante ou até catastrófica, pois esses dois métodos, para serem viáveis em execução, precisam passar por uma redução de dimensionalidade severa, proposta pelos autores dos algoritmos, que leva a perda de desempenho. Os resultados sugerem superioridade de desempenho do método proposto com respeito aos métodos comparados da família LVQ em acurácia e desempenho temporal quando as bases se tornam muito complexas. Os métodos GMLVQ e LGMLVQ sofrem intensamente tanto na queda da acurácia quanto no aumento do tempo. Dessa forma, para as bases de categorização de cenas, o MOCE-SOM obteve resultados consistentemente superiores a todos os métodos comparados.

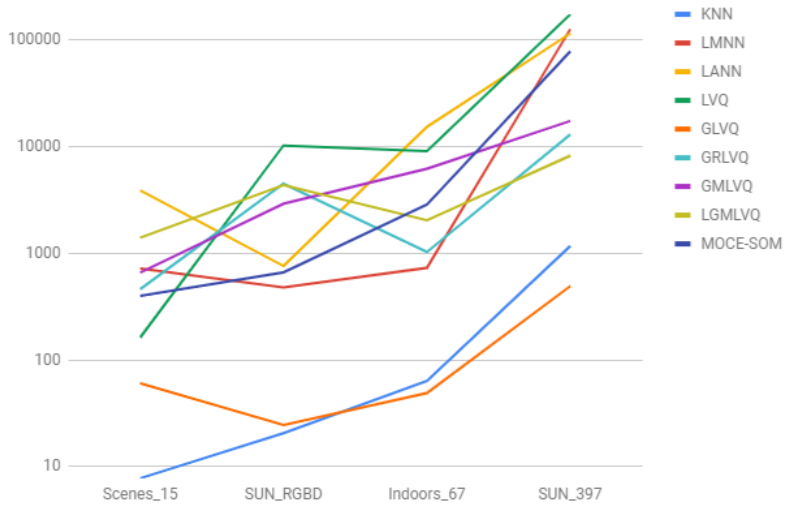

Figura 21 – Tempo de treinamento em escala logarítmica para categorização de cenas.

5.3.4 Análise por Camadas do MOCE-SOM

O modelo MOCE-SOM incorpora diversos recursos ao longo de suas camadas para aprimorar a acurácia em várias situações. A Tabela 7 exibe a acurácia em diferentes etapas do MOCE-SOM, permitindo a observação do impacto de cada nova contribuição adicionada ao MOCE-SOM, análise *ablation*. A Figura 22 mostra de forma gráfica o ablation. As configurações analisadas incluem: SOM padrão, Camada 1 (sem relevâncias), Camada 1, Camada 3, Camada 1 + relevâncias inter, Camada 3 + Inter e Camada 3 + Inter + Inferência Variacional. O SOM padrão e a Camada 1 (sem relevâncias) não existe de fato no modelo MOCE-SOM, ela somente foi simulada para nivel de comparação. Os termos adicionados na legenda informam a funcionalidade incorporada com respeito ao primeiro caso, Camada 1 (sem relevâncias). As diferenças entre essas cinco etapas destacam: o treinamento por somente uma categoria, a inclusão das relevâncias intra-categorias, a repercussão da inclusão das relevâncias inter-categoria; a influência das diversas camadas do modelo, e, por fim, o papel do refinamento.

A primeira avaliação compara o SOM padrão com o treinamento por múltiplos módulos. Essa modificação tem um impacto inicial bastante positivo, especialmente em bases de dados com muitas categorias. Observa-se que essa abordagem de múltiplos módulos já induz, desde o início, uma significativa separação entre as categorias. Em geral, o aumento da acurácia da Camada 1 (sem relevâncias) analisada do MOCE-SOM, Camada 1 (sem relevâncias), até a camada final geralmente resulta em um aumento de 1% a 4% na acurácia do modelo para diversas bases. No caso do Ionosphere chegou a 8% de ganho entre todas as camadas. Isso

estima o ganho do modelo ao longo de sua propagação pelas camadas. A transição entre a situação da Camada 1 (sem relevâncias) e a Camada 1, melhora até em 3%. A transição da Camada 1 para Camada 3 apresentou uma pequeno acréscimo nos resultados, principalmente os de baixa dimensionalidade. A Camada 1 + Inter apresentou um acréscimo na acurácia, com ganhos que chegam a ser superiores a 2% em diferentes bases.

O efeito das três camadas com Inter do modelo pode ser observado na comparação entre Camada 1 + Inter e Camada 3 + Inter. Esse ganho foi mais significativo em bases de dados que possuem uma quantidade maior de categorias. Quando a quantidade de categorias é muito alta, o ganho entre as múltiplas camadas torna-se insignificante. Destaca-se o conjunto de dados Ionosphere, que possui apenas 2 categorias, apresentando um ganho entre as camadas de 3%, CIFAR_10_HoG, que alcança um aumento de 2%, e Scenes_15, também com um aumento de 2%.

A última análise aborda a influência do refinamento pela Inferência Variacional no final do modelo. Esse ganho foi variado nos casos analisados. Foi mais expressivo quando as bases tinham poucas categorias e a acurácia percentual do modelo era relativamente baixa. No entanto, esse ganho de refinamento mostrou-se disperso entre os casos, com magnitude geralmente entre 0.5% e 3%. Os ganhos de acurácia nas diversas características do modelo não acontecem todas ao mesmo tempo, os ganhos se alternam para cada situação.

5.3.5 Discussão

Este capítulo abordou os experimentos conduzidos com o MOCE-SOM, comparando-o com métodos selecionados da literatura. Os experimentos foram conduzidos em três grupos de conjuntos de dados: mineração de dados, categorização de objetos e categorização de cenas. A apresentação dos experimentos procurou seguir a ordem de complexidade dos conjuntos de dados, permitindo a avaliação do comportamento dos métodos em diversas situações, considerando a complexidade em termos de quantidade de amostras, categorias e de atributos dos dados. A avaliação de desempenho dos métodos considerou acurácia e tempo. Buscou-se avaliar a escalabilidade dos métodos para problemas desafiadores de categorização.

Para problemas de mineração de dados, o MOCE-SOM evidenciou ser uma alternativa competitiva, alcançando desempenho comparável aos métodos em comparação como LMNN e LGMLVQ. No âmbito da categorização de objetos, o método proposto apresentou precisão superior em comparação com os métodos avaliados, embora tenha registrado um pequeno

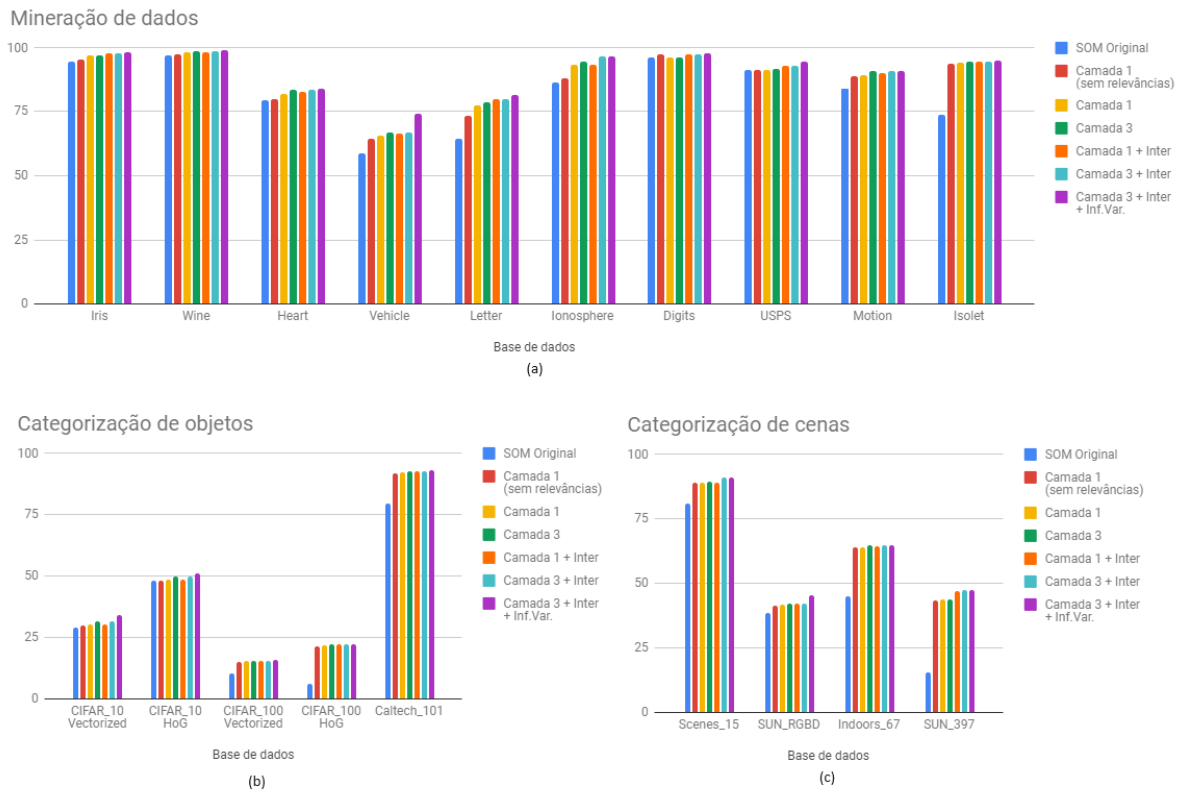

Figura 22 – Análise de ablation: (a) Minaração de dados, (b) categorização de objetos e (c) categorização de cenas.

Tabela 7 – Tabela com os resultados para todas as bases de dados comparando a principais etapas do MOCE-SOM. São apresentados todas as médias e desvios padrões da acurácia em cada etapa principal do modelo e na linha final é comparado o ganho médio entre as configurações.

Base de dados	SOM Original	Camada 1 sem relevâncias	Camada 1	Camada 3	Camada 1 + Inter	Camada 3 + Inter	Camada 3 + Inter + Inf.Var.
Iris	94,59 (2,85)	95,41 (2,63)	97,03 (1,99)	97,30 (2,21)	97,84 (2,13)	97,84 (2,13)	98,38 (1,40)
Wine	97,27 (2,35)	97,50 (2,92)	98,18 (1,79)	98,64 (1,59)	98,18 (1,79)	98,64 (1,59)	99,09 (1,17)
Heart	79,40 (3,43)	79,85 (3,46)	81,94 (3,88)	83,43 (3,33)	82,54 (5,56)	83,43 (3,33)	83,97 (3,02)
Vehicle	58,82 (3,60)	64,22 (2,24)	65,69 (1,86)	66,68 (4,60)	66,29 (3,34)	66,68 (4,60)	74,02 (2,65)
Letter	64,60 (3,31)	73,45 (2,08)	77,40 (1,81)	78,40 (1,71)	79,75 (2,07)	79,85 (2,07)	81,54 (2,31)
Ionosphere	86,59 (5,30)	88,18 (5,85)	93,41 (3,78)	94,55 (3,89)	93,64 (3,68)	96,59 (1,61)	96,65 (1,59)
Digits	96,19 (0,41)	97,50 (0,12)	96,51 (0,01)	96,51 (0,01)	97,36 (0,15)	97,36 (0,15)	97,83(0,14)
USPS	91,46 (0,50)	91,47 (0,28)	91,67 (0,30)	91,83 (0,17)	93,20 (0,26)	93,25 (0,36)	94,85 (0,32)
Motion Tracking	84,14 (1,02)	89,20 (0,21)	89,39 (1,16)	90,95 (0,21)	90,42 (0,19)	90,95 (0,21)	91,13 (0,20)
Isolet	73,89 (1,16)	93,98 (0,23)	94,39 (0,23)	94,71 (0,15)	94,59 (0,23)	94,71 (0,15)	95,12 (0,14)
CIFAR_10 Vectorized	28,93 (0,46)	30,10 (0,11)	30,20 (0,21)	31,70 (0,32)	30,50 (0,24)	31,70 (0,32)	34,24 (0,36)
CIFAR_10 HoG	48,04 (0,25)	48,32 (0,30)	48,55 (0,32)	50,02 (0,25)	48,65 (0,37)	50,02 (0,25)	51,27 (0,23)
CIFAR_100 Vectorized	10,43 (0,31)	15,08 (0,20)	15,30 (0,25)	15,40 (0,26)	15,37 (0,27)	15,40 (0,26)	15,83 (0,24)
CIFAR_100 HoG	6,13 (0,18)	21,40 (0,05)	21,80 (0,02)	22,17 (0,01)	22,12 (0,01)	22,17 (0,01)	22,32 (0,02)
Caltech_101	79,54 (3,17)	92,01 (0,01)	92,47 (0,21)	92,94 (0,19)	92,81 (0,20)	92,94 (0,19)	93,02 (0,18)
Scenes_15	80,85 (0,51)	88,98 (1,14)	89,07 (0,59)	89,28 (0,89)	89,10 (0,76)	91,21 (1,17)	91,23 (1,13)
SUN_RGBD	38,68 (1,43)	41,43 (0,17)	41,98 (0,40)	42,34 (0,22)	42,34 (0,22)	42,34 (0,22)	45,42 (0,40)
Indoors_67	44,85 (0,53)	63,93 (0,11)	64,13 (0,35)	64,75 (0,34)	64,33 (0,31)	64,75 (0,34)	64,91 (0,33)
SUN_397	15,45 (0,51)	43,46 (0,01)	43,64 (0,01)	43,67 (0,90)	46,86 (0,02)	47,37 (0,30)	47,42 (0,13)
Ganho médio	-	7,14	0,91	0,42	0,27	0,60	1,11

aumento de tempo de processamento. Em relação à categoria mais desafiadora de categorização de cenas, o MOCE-SOM alcançou uma acurácia nitidamente melhor mesmo em cenários complexos. Esse resultado é particularmente difícil de se obter devido à arduidade intrínseca em estimar as relevâncias dos atributos, devido à considerável disparidade entre a quantidade de atributos e o número de amostras por categoria, além da elevada quantidade de categorias envolvidas. Vale ressaltar que, no contexto específico do LMNN, sua acurácia em comparação com o MOCE-SOM é reduzida. No caso do LGMLVQ, para torná-lo viável, é necessário realizar redução na dimensionalidade, implicando na perda de capacidade discriminativa em casos mais extremos. Além disso, o LGMLVQ apresenta um aumento no custo computacional de forma quadrática em relação à quantidade de atributos.

A análise comparativa entre as Tabelas 4 e 5 proporciona uma visão da estabilidade e robustez dos modelos. Alguns modelos podem exibir uma acurácia muito elevada, mas carecer de estabilidade, como é o caso do LGMLVQ. Na comparação com nosso modelo, o LGMLVQ apresenta um empate com $8/8/3$ para o melhor resultado; entretanto, ao considerar a média e o desvio padrão, essa comparação diminui para $7/10/2$. Ao comparar os melhores resultados de cada modelo, o MOCE-SOM demonstra uma ligeira superioridade sobre o LMNN, que, por sua vez, evidencia ser tambem um modelo robusto. O MOCE-SOM e o LMNN não variam muito do melhor resultado em relação a melhor média. Em relação a todos os outros métodos analisados para o melhor resultado, o MOCE-SOM mostrou-se significativamente superior.

Assim, o MOCE-SOM emerge como uma alternativa competitiva para a categorização de problemas simples de mineração, destacando-se como um método altamente eficaz para enfrentar desafios complexos e exigentes na categorização de bases de dados. As características incorporadas ao modelo, como o treinamento de apenas uma categoria por módulo, as relevâncias intra e inter, o processamento em múltiplas camadas e a inferência variacional, contribuem para tornar o modelo uma opção eficaz na resolução de problemas de categorização principalmente para dados com alta dimensionalidade como aqueles envolvendo reconhecimento de cenas em ambientes internos e externos.

6 CONCLUSÃO

O modelo designado como MOCE-SOM é constituído por múltiplos módulos especializados, cada um treinado exclusivamente para uma categoria específica. Dentro desses módulos, encontram-se SC-SOMs organizados em diversas camadas, destinadas a realizar o agrupamento de dados com diferentes níveis de complexidade. Durante a fase de treinamento de cada SC-SOM, são determinados seus protótipos, empregando a relevância de diferentes atributos para lidar eficientemente com dados de alta dimensionalidade. O MOCE-SOM é um algoritmo da agrupamento em subespaço flexível pois atribui valores contínuos entre zero e um para as relevâncias, de maneira a dar peso maior para atributos relevantes na caracterização de agrupamentos em detrimento das características não influentes. A composição dos dados de entrada para os blocos SC-SOM causa uma ligeira modificação na distribuição de amostras, aproximando-as de seus protótipos que propicia uma pequena melhoria na precisão da categorização. Ao final do treinamento de cada SC-SOM, é incorporada uma segunda ponderação dos atributos, agora proveniente da influência das relevâncias inter-categorias. Essas relevâncias reforçam a distinção entre as categorias no modelo, enfatizando as dimensões que levam a tais diferenças. O sinal propagado entre os blocos SC-SOM de um módulo resulta de combinação linear entre o protótipo vencedor da camada que emite o sinal e o sinal de entrada, a instância sob processamento. Adicionalmente, ao término de todas as camadas, um processo de refinamento é executado por meio de inferência variacional.

Todas essas características foram incorporadas ao MOCE-SOM com o objetivo de elevar a acurácia do modelo sem comprometer seu desempenho em comparação com outras abordagens que empregam o cálculo da distância por vetores de referência. Estas abordagens frequentemente apresentam um custo computacional que crescem significativamente com o aumento de complexidade dos dados ao mesmo tempo que perdem desempenho devido a simplificações para serem executados ou perda de propriedades.

Os experimentos abrangeram 17 conjuntos de dados (em 19 representações) com características diversas, incluindo tamanhos de espaços amostrais, número de categorias e de dimensões. O MOCE-SOM destacou-se ao atingir desempenho consistentemente competitivo nesses conjuntos de dados diversos, competindo com métodos comparativos em termos de precisão a custo computacional melhor. Em particular, o desempenho foi especialmente promissor para os conjuntos de dados com inúmeros atributos, categorias e alta dimensionalidade. Além

disso, o MOCE-SOM apresentou um tempo de treinamento mais curto para as bases de dados mais complexas em comparação com métodos comparáveis, sem comprometer a precisão, tornando-se uma escolha competitiva. Para bases de dados mais simples em complexidade, o MOCE-SOM se mostrou um modelo que possui resultados comparáveis com os métodos avaliados nesta tese.

Os aprimoramentos proporcionados por cada etapa do MOCE-SOM foram detalhados na Tabela 7. Uma das características introduzidas pelo MOCE-SOM é a implementação de módulos especialistas no treinamento de cada categoria individualmente e o aprendizado de relevâncias intra-categorias, o que se reflete em resultados satisfatórios já observados na Camada 1. A forma de treinar cada módulo com somente uma categoria cria um distanciamento entre as amostras de cada categoria que não tendem a se misturar durante o processo de treinamento evitando a confusão e sobreposição de amostras de categorias diferentes entre elas.

Já a inserção de relevâncias intra no modelo melhora os resultados em até 2%. Diversas abordagens para aprender as relevâncias intra foram avaliadas, sendo constatado que o uso de relevâncias para cada nó individualmente não trouxe melhorias significativas, pois as diferenças entre as relevâncias eram sutis e não se mostraram recompensadoras. Portanto, as relevâncias são calculadas por grupo de uma forma menos custosa e eficaz. Aplicação das relevâncias intra-categorias busca evidência o que mais é importante para cada categoria isoladamente, contribuindo para um agrupamento mais coeso.

Apesar da aplicação do aprendizado de relevâncias intra, observou-se que a relação de relevâncias entre as categorias não estava sendo considerada da mellhor forma, o que poderia contribuir para a discriminação de amostras em situações de confusão entre categorias. Ao analisar a variância entre as variâncias das categorias, procuramos destacar atributos que fossem distintamente importantes entre as categorias. Foram exploradas várias abordagens para incorporar o conceito de relevâncias inter durante o treinamento. Essas tentativas incluíram a introdução de informações no processo de treinamento por meio de composição de intra e inter, convolução; no entanto, observou-se que a informação inter prejudicava a convergência do treinamento ou essa informação era minimizada no processo. A alternativa adotada foi inserir a informação inter após o treinamento. Além disso, a variância das relevâncias intra mostrou-se mais eficaz do que a variação direta nos pesos. A incorporação das relevâncias intercategorias resultou em um aumento na acurácia, variando em média de 0.2% a 2%, conforme observado na Camada 1 + Inter.

A característica de múltiplas camadas contribui para resolver problemas em mais de uma

etapa, inserindo informações importantes da camada anterior na camada posterior. Surgiu, assim, o conceito de composição das amostras com os protótipos vencedores para as camadas subsequentes. Essa composição visa modificar a distribuição dos dados, fazendo com que amostras inicialmente categorizadas incorretamente sejam reatribuídas à categoria correta, enquanto as categorizadas corretamente permaneçam atribuídas à categoria correta. Os resultados para as múltiplas camadas proporcionaram um aumento adicional na acurácia, variando de 0.2% a 3%, conforme apresentado na Camada 3 + Inter.

A adoção da abordagem de inferência variacional ao final do modelo, em conjunto com as relevâncias, tem o objetivo de capturar a distribuição dos dados. Dado que esse método é sensível à inicialização dos centros, a determinação dos centros pelas camadas da rede e a contribuição das relevâncias são cruciais para uma boa inicialização do modelo. Sem essa boa inicialização, o processo de inferência variacional pode ficar preso em soluções locais. A inclusão da inferência variacional na última etapa do modelo também contribuiu para um aumento na acurácia, com ganhos médios entre 0.2% e 3%. Essa combinação de melhorias na acurácia, embora nem todas ocorram simultaneamente ou em todas as situações, resultou em um desempenho promissor do MOCE-SOM nos experimentos realizados nesta tese.

Analisando o MOCE-SOM na comparação da robustez dos seus resultados foi calculada a média dos desvios padrões de seus resulados. A média dos desvios padrões de cada métodos para todas as bases de dados são: 1,55 para o k-NN, de 1,10 para o LMNN, de 1,38 para o LANN de 1,87 para LVQ, de 1,71 para GLVQ, de 1,55 para o GRLVQ, de 1,41 para o GMLVQ, de 1,35 para o LGMLVQ e, finalmente, de 0,84 para o MOCE-SOM. O nosso método apresentou a menor média de desvio padrão em relação a todos os métodos, inclusive o LMNN e o LGMLVQ que se mostraram os métodos mais competitivos. Essas informações demonstram a robustez e confiabilidade do nosso método proposto.

6.1 LIMITAÇÕES DO MODELO

Embora tenha alcançado avanços substanciais, o modelo proposto ainda apresenta algumas limitações notáveis:

- O modelo atual adota SC-SOMs para cada categoria com uma quantidade fixa de nodos, determinada por um parâmetro predefinido. Para conjuntos de dados com um grande número de categorias, essa abordagem possui um grande impacto, uma vez que muitos

nodos podem estar subutilizados. Além disso, a consideração da existência de diferentes quantidades de subcategorias para cada categoria não é abordada no modelo proposto, pressupondo o mesmo número de subcategorias para todas as categorias, o que pode não ser adequado em muitos casos;

- Atualmente, todas as camadas do modelo processam as amostras com a mesma dimensionalidade presente na camada inicial. Essa característica implica que a adição de mais camadas à rede é um processo computacionalmente custoso;
- O MOCE-SOM foi apresentado com três camadas, e não foram observadas melhorias significativas na acurácia além dessa quantidade de camadas. Desafios adicionais surgem ao tentar tornar eficazes camadas adicionais no processo.

6.2 TRABALHOS FUTUROS

A pesquisa dedicada à categorização de lugares e cenas é uma área promissora que enfrenta desafios significativos, com diversas aplicações ainda em estágios iniciais de desenvolvimento. Para futuras investigações, sugere-se:

- Explorar modificações no processo de combinação para possibilitar as camadas mais profundas a continuarem aprimorando os resultados mesmo com um grande número de camadas;
- Investigar um processo de redução de dimensionalidade das amostras nas camadas subsequentes da rede, baseado nas relevâncias inter-categorias;
- Abordar a incorporação de novos tipos de categorias de cenas não inicialmente aprendidas pelo modelo, através de estratégias de aprendizagem contínua (ZHONG et al., 2017; SOUSA; BASSANI, 2018);
- Realizar a integração e validação dos algoritmos em um robô real, aplicando o mapeamento e navegação semântica no contexto prático;
- Adaptar o treinamento da rede para uma versão que possa ser facilmente executada em GPUs, aproveitando assim o poder de processamento desses recursos computacionais.

REFERÊNCIAS

ARAÚJO, A. F.; ANTONINO, V. O.; PONCE-GUEVARA, K. L. Self-organizing subspace clustering for high-dimensional and multi-view data. *Neural Networks*, Elsevier, v. 130, p. 253–268, 2020.

ARAÚJO, A. R.; COSTA, D. C. Local adaptive receptive field self-organizing map for image color segmentation. *Image and Vision Computing*, Elsevier, v. 27, n. 9, p. 1229–1239, 2009.

ASTUDILLO, C. A.; OOMMEN, B. J. Topology-oriented self-organizing maps: a survey. *Pattern Analysis and Applications*, p. 223–248, 2014.

ATTAOUI, M. O.; AZZAG, H.; LEBBAH, M.; KESKES, N. Subspace data stream clustering with global and local weighting models. *Neural Computing and Applications*, Springer, v. 33, n. 8, p. 3691–3712, 2021.

BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 39, n. 12, p. 2481–2495, 2017.

BALASKA, V.; BAMPIS, L.; BOUDOURIDES, M.; GASTERATOS, A. Unsupervised semantic clustering and localization for mobile robotics tasks. *Robotics and Autonomous Systems*, Elsevier, v. 131, p. 103567, 2020.

BANSAL, M.; KUMAR, M.; SACHDEVA, M.; MITTAL, A. Transfer learning for image classification using vgg19: Caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing*, Springer, p. 1–12, 2021.

BASSANI, H. F.; ARAÚJO, A. F. Dimension selective self-organizing maps for clustering high dimensional data. In: IEEE. *The 2012 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2012. p. 1–8.

BASSANI, H. F.; ARAUJO, A. F. Dimension selective self-organizing maps with time-varying structure for subspace and projected clustering. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 26, n. 3, p. 458–471, 2014.

BATISTA, G.; SILVA, D. F. et al. How k-nearest neighbor parameters affect its performance. In: CITESEER. *Argentine symposium on artificial intelligence*. [S.l.], 2009. p. 1–12.

BAY, H.; ESS, A.; TUYTELAARS, T.; GOOL, L. V. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, p. 346–359, 2008.

BEHZADAN, A. H.; KAMAT, V. R. Integrated information modeling and visual simulation of engineering operations using dynamic augmented reality scene graphs. *Journal of Information Technology in Construction (ITcon)*, v. 16, n. 17, p. 259–278, 2011.

BENABDESLEM, K.; LEBBAH, M. Feature selection for self-organizing map. In: IEEE. *International Conference on Information Technology Interfaces*. [S.l.], 2007. p. 45–50.

BEYER, K.; GOLDSTEIN, J.; RAMAKRISHNAN, R.; SHAFT, U. When is "nearest neighbor" meaningful? In: SPRINGER. *International Conference on Database Theory (ICDT)*. [S.l.], 1999. p. 217–235.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4.

BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, IOP Publishing, v. 2008, n. 10, p. P10008, 2008.

BONIN-FONT, F.; ORTIZ, A.; OLIVER, G. Visual Navigation for Mobile Robots: A Survey. *Journal of Intelligent and Robotic Systems ((JINT))*, p. 263–296, 2008.

BOUTELL, M. R.; LUO, J.; SHEN, X.; BROWN, C. M. Learning multi-label scene classification. *Pattern recognition*, Elsevier, v. 37, n. 9, p. 1757–1771, 2004.

CHATTERJEE, S.; HADI, A. S. *Sensitivity analysis in linear regression*. [S.l.]: John Wiley & Sons, 2009.

CHENG, X.; LU, J.; FENG, J.; YUAN, B.; ZHOU, J. Scene recognition with objectness. *Pattern Recognition*, Elsevier, v. 74, p. 474–487, 2018.

CHOUDHARY, S.; TREVOR, A. J.; CHRISTENSEN, H. I.; DELLAERT, F. Slam with object discovery, modeling and mapping. In: IEEE. *International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2014. p. 1018–1025.

CIMPOI, M.; MAJI, S.; VEDALDI, A. Deep filter banks for texture recognition and segmentation. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 3828–3836.

COOMANS, D.; MASSART, D. L. Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, Elsevier, v. 136, p. 15–27, 1982.

COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IEEE, v. 13, n. 1, p. 21–27, 1967.

CRESPO, J.; CASTILLO, J. C.; MOZOS, O. M.; BARBER, R. Semantic information for robot navigation: A survey. *Applied Sciences*, MDPI, v. 10, n. 2, p. 497, 2020.

CSURKA, G.; DANCE, C.; FAN, L.; WILLAMOWSKI, J.; BRAY, C. Visual categorization with bags of keypoints. In: PRAGUE. *Workshop on statistical learning in computer vision (ECCV)*. [S.l.], 2004. v. 1, n. 1-22, p. 1–2.

CSURKA, G.; DANCE, C.; FAN, L.; WILLAMOWSKI, J.; BRAY, C. Visual categorization with bags of keypoints. In: PRAGUE. *Workshop on Statistical Learning in Computer Vision (ECCV)*. [S.l.], 2004. v. 1, n. 1-22, p. 1–2.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2005. v. 1, p. 886–893.

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

DIXIT, M.; CHEN, S.; GAO, D.; RASIWASIA, N.; VASCONCELOS, N. Scene classification with semantic fisher vectors. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 2974–2983.

DOMENICONI, C.; GUNOPULOS, D.; PENG, J. Large margin nearest neighbor classifiers. *IEEE Transactions on Neural Networks*, IEEE, v. 16, n. 4, p. 899–909, 2005.

DONG, Y.; GAO, S.; TAO, K.; LIU, J.; WANG, H. Performance evaluation of early and late fusion methods for generic semantics indexing. *Pattern Analysis and Applications*, Springer, v. 17, n. 1, p. 37–50, 2014.

DONOHO, D. L.; MALEKI, A.; MONTANARI, A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences (PNAS)*, National Acad Sciences, v. 106, n. 45, p. 18914–18919, 2009.

DU, D.; WANG, L.; WANG, H.; ZHAO, K.; WU, G. Translate-to-recognize networks for rgb-d scene recognition. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. p. 11836–11845.

DURAND, T.; THOME, N.; CORD, M. Weldon: Weakly supervised learning of deep convolutional neural networks. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 4743–4752.

EZUGWU, A. E.; IKOTUN, A. M.; OYELADE, O. O.; ABUALIGAH, L.; AGUSHAKA, J. O.; EKE, C. I.; AKINYELU, A. A. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 110, p. 104743, 2022.

FEI-FEI, L.; FERGUS, R.; PERONA, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: IEEE. *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2004. p. 178–178.

GAO, B.-B.; WEI, X.-S.; WU, J.; LIN, W. Deep spatial pyramid: The devil is once again in the details. *arXiv preprint arXiv:1504.05277*, 2015.

GIRSHICK, R. Fast r-cnn. In: *International Conference on Computer Vision (CVPR)*. [S.l.: s.n.], 2015. p. 1440–1448.

GONG, Y.; WANG, L.; GUO, R.; LAZEBNIK, S. Multi-scale orderless pooling of deep convolutional activation features. In: SPRINGER. *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.], 2014. p. 392–407.

GÖPFERT, J. P.; WERSING, H.; HAMMER, B. Interpretable locally adaptive nearest neighbors. *Neurocomputing*, Elsevier, v. 470, p. 344–351, 2022.

GOREN-BAR, D.; KUFLIK, T. Can automatic personal categorization deal with user inconsistency? *Recommendation and Personalization in eCommerce*, Citeseer, p. 116, 2002.

GRAVES, A. Long short-term memory. *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer, p. 37–45, 2012.

GREENE, M. R.; OLIVA, A. The briefest of glances: The time course of natural scene understanding. *Psychological Science*, SAGE Publications Sage CA: Los Angeles, CA, v. 20, n. 4, p. 464–472, 2009.

GUO, S.; HUANG, W.; WANG, L.; QIAO, Y. Locally supervised deep hybrid model for scene recognition. *IEEE Transactions on Image Processing (TIP)*, IEEE, v. 26, n. 2, p. 808–820, 2016.

GUZEL, M. Autonomous vehicle navigation using vision and mapless strategies: a survey. *Advances in Mechanical Engineering*, p. 263–296, 2013.

HAJJAR, C.; HAMDAN, H. Interval data clustering using self-organizing maps based on adaptive mahalanobis distances. *Neural Networks*, Elsevier, v. 46, p. 124–132, 2013.

HAMMER, B.; VILLMANN, T. Generalized relevance learning vector quantization. *Neural Networks*, Elsevier, v. 15, n. 8-9, p. 1059–1068, 2002.

HAYAT, M.; KHAN, S. H.; BENNAMOUN, M.; AN, S. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing (TIP)*, IEEE, v. 25, n. 10, p. 4829–4841, 2016.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.

HERMANS, A.; FLOROS, G.; LEIBE, B. Dense 3d semantic mapping of indoor scenes from rgb-d images. In: IEEE. *International Conference on Robotics and Automation (ICRA)*. [S.l.], 2014. p. 2631–2638.

HERRANZ, L.; JIANG, S.; LI, X. Scene recognition with cnns: objects, scales and dataset bias. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 571–579.

HOU, J.; ZENG, H.; ZHU, J.; HOU, J.; CHEN, J.; MA, K.-K. Deep quadruplet appearance learning for vehicle re-identification. *IEEE Transactions on Vehicular Technology*, IEEE, v. 68, n. 9, p. 8512–8522, 2019.

JÉGOU, H.; DOUZE, M.; SCHMID, C.; PÉREZ, P. Aggregating local descriptors into a compact image representation. In: IEEE. *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2010. p. 3304–3311.

JÚNIOR, M. R. da S.; ARAÚJO, A. F. Subspace clustering multi-module self-organizing maps with two-stage learning. In: SPRINGER. *International Conference on Artificial Neural Networks (ICANN)*. [S.l.], 2022. p. 285–296.

KARPATHY, A.; TODERICI, G.; SHETTY, S.; LEUNG, T.; SUKTHANKAR, R.; FEI-FEI, L. Large-scale video classification with convolutional neural networks. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2014. p. 1725–1732.

KELLY, M.; LONGJOHN, R.; NOTTINGHAM, K. The uci machine learning repository. *URL https://archive. ics. uci. edu*, 2023.

KHAN, S. H.; HAYAT, M.; BENNAMOUN, M.; TOGNERI, R.; SOHEL, F. A. A discriminative representation of convolutional features for indoor scene recognition. *IEEE Transactions on Image Processing (TIP)*, IEEE, v. 25, n. 7, p. 3372–3383, 2016.

KHAN, S. S.; MADDEN, M. G. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, Cambridge University Press, v. 29, n. 3, p. 345–374, 2014.

KIM, H. J.; FRAHM, J.-M. Hierarchy of alternating specialists for scene recognition. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 451–467.

KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

KLEINBERG, J. An impossibility theorem for clustering. *Advances in Neural Information Processing Systems*, v. 15, 2002.

KOHONEN, T. Improved versions of learning vector quantization. In: IEEE. *International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 1990. p. 545–550.

KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, IEEE, v. 78, n. 9, p. 1464–1480, 1990.

KOHONEN, T. Learning vector quantization. *Self-organizing maps*, Springer, p. 175–189, 1995.

KOHONEN, T. *Self-organization and associative memory*. [S.l.]: Springer Science & Business Media, 2012. v. 8.

KOHONEN, T.; HARI, R. Where the abstract feature maps of the brain might come from. *Trends in Neurosciences*, p. 135–139, 1999.

KOHONEN, T.; SCHROEDER, M. R.; HUANG, T. S. (Ed.). *Self-Organizing Maps*. [S.l.]: Springer-Verlag New York, Inc., 2001.

KOSTAVELIS, I.; CHARALAMPOUS, K.; GASTERATOS, A.; TSOTSOS, J. K. Robot navigation via spatial and temporal coherent semantic maps. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 48, p. 173–187, 2016.

KOSTAVELIS, I.; GASTERATOS, A. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, Elsevier B.V., p. 86–103, 2015.

KOSTAVELIS, I.; GASTERATOS, A. Semantic maps from multiple visual cues. *Expert Systems with Applications*, Elsevier, v. 68, p. 45–57, 2017.

KRISHNAN, A. K.; KRISHNA, K. M. A visual exploration algorithm using semantic cues that constructs image based hybrid maps. In: IEEE. *International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2010. p. 1316–1321.

KRIZHEVSKY, A.; HINTON, G. et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, v. 25, 2012.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, AcM New York, NY, USA, v. 60, n. 6, p. 84–90, 2017.

LAZEBNIK, S.; SCHMID, C.; PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE. *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2006. v. 2, p. 2169–2178.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LI, J.; LIN, D.; WANG, Y.; XU, G.; ZHANG, Y.; DING, C.; ZHOU, Y. Deep discriminative representation learning with attention map for scene classification. *Remote Sensing*, MDPI, v. 12, n. 9, p. 1366, 2020.

LI, Y.; ZHANG, Z.; CHENG, Y.; WANG, L.; TAN, T. Mapnet: Multi-modal attentive pooling network for rgb-d indoor scene classification. *Pattern Recognition*, Elsevier, v. 90, p. 436–449, 2019.

LIN, M.; CHEN, Q.; YAN, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

LIN, T.-Y.; DOLLÁR, P.; GIRSHICK, R.; HE, K.; HARIHARAN, B.; BELONGIE, S. Feature pyramid networks for object detection. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 2117–2125.

LIU, B.; LIU, J.; WANG, J.; LU, H. Learning a representative and discriminative part model with deep convolutional features for scene recognition. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2014. p. 643–658.

LIU, C.; XIE, J.; ZHAO, Q.; XIE, Q.; LIU, C. Novel evolutionary multi-objective soft subspace clustering algorithm for credit risk assessment. *Expert Systems with Applications*, Elsevier, v. 138, p. 112827, 2019.

LIU, L.; CHEN, J.; FIEGUTH, P.; ZHAO, G.; CHELLAPPA, R.; PIETIKÄINEN, M. From bow to cnn: Two decades of texture representation for texture classification. *International Journal of Computer Vision (IJCV)*, Springer, v. 127, n. 1, p. 74–109, 2019.

LIU, L.; OUYANG, W.; WANG, X.; FIEGUTH, P.; CHEN, J.; LIU, X.; PIETIKÄINEN, M. Deep learning for generic object detection: A survey. *International Journal of Computer Vision (IJCV)*, Springer, v. 128, n. 2, p. 261–318, 2020.

LIU, Y.; CHEN, Q.; CHEN, W.; WASSELL, I. Dictionary learning inspired deep network for scene recognition. In: *Conference on Artificial Intelligence (AAAI)*. [S.l.: s.n.], 2018. v. 32, n. 1.

LÓPEZ-CIFUENTES, A.; ESCUDERO-VIÑOLO, M.; BESCÓS, J.; GARCÍA-MARTÍN, Á. Semantic-aware scene recognition. *Pattern Recognition*, Elsevier, v. 102, p. 107256, 2020.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, p. 91–110, 2004.

MANN, H. B.; WHITNEY, D. R. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, JSTOR, p. 50–60, 1947.

MOUSAVIAN, A.; KOŠECKÁ, J.; LIEN, J.-M. Semantically guided location recognition for outdoors scenes. In: IEEE. *International Conference on Robotics and Automation (ICRA)*. [S.l.], 2015. p. 4882–4889.

MOZOS, O. M.; TRIEBEL, R.; JENSFELT, P.; ROTTMANN, A.; BURGARD, W. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, Elsevier, v. 55, n. 5, p. 391–402, 2007.

NANNI, L.; GHIDONI, S.; BRAHNAM, S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern recognition*, Elsevier, v. 71, p. 158–172, 2017.

NEE, A. Y.; ONG, S.; CHRYSSOLOURIS, G.; MOURTZIS, D. Augmented reality applications in design and manufacturing. *CIRP Annals*, Elsevier, v. 61, n. 2, p. 657–679, 2012.

NOVA, D.; ESTÉVEZ, P. A. A review of learning vector quantization classifiers. *Neural Computing and Applications*, Springer, v. 25, n. 3, p. 511–524, 2014.

OLIVA, A.; TORRALBA, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer vision*, Springer, v. 42, p. 145–175, 2001.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

PAUL, R.; TRIEBEL, R.; RUS, D.; NEWMAN, P. Semantic categorization of outdoor scenes with uncertainty estimates using multi-class gaussian process classification. In: IEEE. *International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2012. p. 2404–2410.

PRONOBIS, A. *Semantic Mapping with Mobile Robots*. Tese (Doutorado) — KTH Royal Institute of Technology, 2011.

PRONOBIS, A.; CAPUTO, B. Confidence-based cue integration for visual place recognition. In: IEEE. *International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2007. p. 2394–2401.

PRONOBIS, A.; CAPUTO, B.; JENSFELT, P.; CHRISTENSEN, H. I. A discriminative approach to robust visual place recognition. In: IEEE. *International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2006. p. 3829–3836.

PRONOBIS, A.; MOZOS, O. M.; CAPUTO, B.; JENSFELT, P. Multi-modal semantic place classification. *International Journal of Robotics Research (SJR)*, SAGE Publications Sage UK: London, England, v. 29, n. 2-3, p. 298–320, 2010.

QUATTONI, A.; TORRALBA, A. Recognizing indoor scenes. In: IEEE. *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2009. p. 413–420.

RANGANATHAN, A. Pliss: labeling places using online changepoint detection. *Autonomous Robots*, Springer, v. 32, n. 4, p. 351–368, 2012.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, v. 28, 2015.

RODRÍGUEZ-SOTELO, J. L.; PELUFFO-ORDONEZ, D.; CUESTA-FRAU, D.; CASTELLANOS-DOMÍNGUEZ, G. Unsupervised feature relevance analysis applied to improve ecg heartbeat clustering. *Computer Methods and Programs in Biomedicine*, Elsevier, v. 108, n. 1, p. 250–261, 2012.

SÁNCHEZ, J.; PERRONNIN, F.; MENSINK, T.; VERBEEK, J. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision (IJCV)*, Springer, v. 105, n. 3, p. 222–245, 2013.

SATO, A.; YAMADA, K. Generalized learning vector quantization. *Advances in Neural Information Processing Systems (NIPS)*, v. 8, 1995.

SCHNEIDER, P.; BIEHL, M.; HAMMER, B. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, MIT Press, v. 21, n. 12, p. 3532–3561, 2009.

SCHNEIDER, P.; BIEHL, M.; HAMMER, B. Distance learning in discriminative vector quantization. *Neural Computation*, MIT Press, v. 21, n. 10, p. 2942–2969, 2009.

SHIH, K. J.; ENDRES, I.; HOIEM, D. Learning discriminative collections of part detectors for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 37, n. 8, p. 1571–1584, 2014.

SILVA, A. F. D.; JÚNIOR, M. R. D. S.; ARAÚJO, A. F.; DURAND-PETITEVILLE, A. Visual place recognition under a high-dimensional subspace clustering perspective. In: IEEE. *Latin American Robotics Symposium (LARS)*. [S.l.], 2023. p. 508–513.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2014.

SONG, S.; LICHTENBERG, S. P.; XIAO, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 567–576.

SONG, X.; JIANG, S.; HERRANZ, L. Combining models from multiple sources for rgb-d scene recognition. In: *International Joint Conference on Artificial Intelligence (IJCAI)*. [S.l.: s.n.], 2017. p. 4523–4529.

SOUSA, Y. C.; BASSANI, H. F. Incremental semantic mapping with unsupervised on-line learning. In: IEEE. *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2018. p. 1–8.

SOUSA, Y. C.; BASSANI, H. F. Topological semantic mapping by consolidation of deep visual features. *IEEE Robotics and Automation Letters (RA-L)*, IEEE, v. 7, n. 2, p. 4110–4117, 2022.

SREENU, G.; DURAI, S. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, SpringerOpen, v. 6, n. 1, p. 1–27, 2019.

SUN, N.; LI, W.; LIU, J.; HAN, G.; WU, C. Fusing object semantics and deep appearance features for scene recognition. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, IEEE, v. 29, n. 6, p. 1715–1728, 2018.

SUN, Y.; WANG, X.; TANG, X. Deeply learned face representations are sparse, selective, and robust. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 2892–2900.

SUTSKEVER, I.; MARTENS, J.; DAHL, G.; HINTON, G. On the importance of initialization and momentum in deep learning. In: PMLR. *International Conference on Machine Learning (ICML)*. [S.l.], 2013. p. 1139–1147.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 1–9.

TANG, P.; WANG, H.; KWONG, S. G-ms2f: Googlenet based multi-stage feature fusion of deep cnn for scene recognition. *Neurocomputing*, Elsevier, v. 225, p. 188–197, 2017.

VOGEL, J.; SCHIELE, B. Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision*, Springer, v. 72, n. 2, p. 133–157, 2007.

WALTHER, D. B.; CHAI, B.; CADDIGAN, E.; BECK, D. M.; FEI-FEI, L. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 108, n. 23, p. 9661–9666, 2011.

WANG, L.; GUO, S.; HUANG, W.; XIONG, Y.; QIAO, Y. Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns. *IEEE Transactions on Image Processing (TIP)*, IEEE, v. 26, n. 4, p. 2055–2068, 2017.

WANG, L.; WANG, Z.; DU, W.; QIAO, Y. Object-scene convolutional neural networks for event recognition in images. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 30–35.

WANG, Z.; WANG, L.; WANG, Y.; ZHANG, B.; QIAO, Y. Weakly supervised patchnets: Describing and aggregating local patches for scene recognition. *IEEE Transactions on Image Processing (TIP)*, IEEE, v. 26, n. 4, p. 2028–2041, 2017.

XIAO, J.; HAYS, J.; EHINGER, K. A.; OLIVA, A.; TORRALBA, A. Sun database: Large-scale scene recognition from abbey to zoo. In: IEEE. *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2010. p. 3485–3492.

XIE, G.-S.; ZHANG, X.-Y.; YAN, S.; LIU, C.-L. Hybrid cnn and dictionary-based models for scene recognition and domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, IEEE, v. 27, n. 6, p. 1263–1274, 2015.

XIONG, Z.; YUAN, Y.; WANG, Q. Rgb-d scene recognition via spatial-related multi-modal feature learning. *IEEE Access*, IEEE, v. 7, p. 106739–106747, 2019.

XIONG, Z.; YUAN, Y.; WANG, Q. Msn: Modality separation networks for rgb-d scene recognition. *Neurocomputing*, Elsevier, v. 373, p. 81–89, 2020.

YANG, S.; RAMANAN, D. Multi-scale recognition with dag-cnns. In: *International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1215–1223.

YOO, D.; PARK, S.; LEE, J.-Y.; KWEON, I. S. Multi-scale pyramid pooling for deep convolutional representation. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 71–80.

YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, v. 27, 2014.

ZENG, D.; LIAO, M.; TAVAKOLIAN, M.; GUO, Y.; ZHOU, B.; HU, D.; PIETIKÄINEN, M.; LIU, L. Deep learning for scene classification: A survey. *Computer Vision and Pattern Recognition (CVPR)*, 2021.

ZHANG, M.-L.; ZHOU, Z.-H. Multi-label learning by instance differentiation. In: *Conference on Artificial Intelligence (AAAI)*. [S.l.: s.n.], 2007. v. 7, p. 669–674.

ZHANG, R.; CANDRA, S. A.; VETTER, K.; ZAKHOR, A. Sensor fusion for semantic segmentation of urban scenes. In: IEEE. *International Conference on Robotics and Automation (ICRA)*. [S.l.], 2015. p. 1850–1857.

ZHANG, T.; LIU, S.; XU, C.; LU, H. Mining semantic context information for intelligent video surveillance of traffic scenes. *IEEE Transactions on Industrial Informatics*, IEEE, v. 9, n. 1, p. 149–160, 2012.

ZHANG, W.; YU, X.; HE, X. Learning bidirectional temporal cues for video-based person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 28, n. 10, p. 2768–2776, 2017.

ZHENG, L.; YANG, Y.; TIAN, Q. Sift meets cnn: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 40, n. 5, p. 1224–1244, 2017.

ZHONG, C.; LIU, S.; LU, Q.; ZHANG, B. Continuous learning route map for robot navigation using a growing-on-demand self-organizing neural network. *International Journal of Advanced Robotic Systems (IJARS)*, SAGE Publications Sage UK: London, England, v. 14, n. 6, p. 1729881417743612, 2017.

ZHOU, B.; KHOSLA, A.; LAPEDRIZA, A.; OLIVA, A.; TORRALBA, A. Learning deep features for discriminative localization. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 2921–2929.

ZHOU, B.; LAPEDRIZA, A.; KHOSLA, A.; OLIVA, A.; TORRALBA, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 40, n. 6, p. 1452–1464, 2017.

ZHOU, B.; LAPEDRIZA, A.; XIAO, J.; TORRALBA, A.; OLIVA, A. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, v. 27, 2014.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. D. Semi-supervised learning using gaussian fields and harmonic functions. In: *International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2003. p. 912–919.