

**UNIVERSIDADE PRESBITERIANA MACKENZIE
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO**

Vinícius Amendoeira Ferraria

**Análise da Robustez de um Detector de Bordas Baseado em
Autômatos Celulares Otimizado por Enxame de Partículas**

Texto de Dissertação apresentado ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie como parte dos requisitos para a obtenção do título de Mestre.

Orientador: Prof. Dr. Eurico Luiz Prospero Ruivo

São Paulo, Brasil
2024

F368a

Ferraria, Vinícius Amendoeira

Mapeando o Análise da Robustez de um detector de bordas baseado em autômatos celulares otimizado por enxame de partículas / Vinícius Amendoeira Ferraria

69 f.; 30 cm

Dissertação (Mestrado em Engenharia Elétrica e Computação) –
Universidade Presbiteriana Mackenzie, São Paulo, 2024.
Orientador: Prof. Dr. Eurico Luiz Prospero Ruivo

Bibliografia: f. 58-63

1. Autômatos Celulares 2. Computação Natural 3. Detecção de Borda 4.
Processamento de Imagem 5. Transfer Learning I. Ruivo, Eurico Luiz
Prospero II. Título

CDD 621.3

Bibliotecária responsável: Maria Gabriela Brandi Teixeira – CRB 8 / 6339

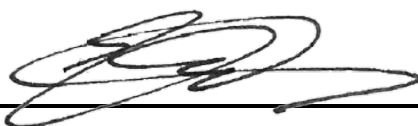
VINÍCIUS AMENDOEIRA FERRARIA

ANÁLISE DA ROBUSTEZ DE UM DETECTOR DE BORDAS BASEADO EM
AUTÔMATOS CELULARES OTIMIZADO POR ENXAME DE PARTÍCULAS

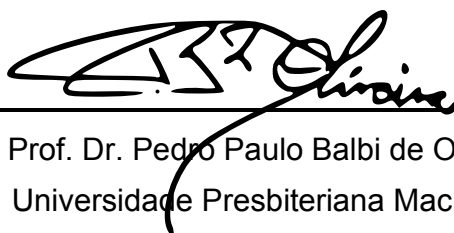
Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie, como requisito parcial à obtenção de título de Mestre em Engenharia Elétrica e Computação.

Aprovada em 8 de fevereiro de 2024


BANCA EXAMINADORA



Prof. Dr. Eurico Luiz Prospero Ruivo
Universidade Presbiteriana Mackenzie



Prof. Dr. Pedro Paulo Balbi de Oliveira
Universidade Presbiteriana Mackenzie

Documento assinado digitalmente
 FABRÍCIO OLIVETTI DE FRANÇA
Data: 08/02/2024 15:41:46-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Fabricio Olivetti de França
Universidade Federal do ABC

Agradecimentos

Gostaria de dedicar minha mais profunda gratidão à minha família por seu encorajamento inabalável e amor incondicional ao longo deste período de pesquisa, sem eles não estaria onde estou hoje. Sua compreensão durante os momentos difíceis foram fundamentais para que pudesse alcançar este marco importante em minha jornada. Agradeço por estarem sempre ao meu lado, oferecendo suporte emocional mesmo nos momentos mais desafiadores.

Agradeço também os meus colegas pesquisadores da PPGEEC que me motivaram durante a minha jornada acadêmica e no desenvolvimento desta dissertação.

Agradeço também a todos os professores, orientadores e colegas que contribuíram de alguma forma para este trabalho. Suas aulas, orientações e discussões foram de suma importância para o desenvolvimento das ideias apresentadas nesta dissertação.

Ao meu orientador, Prof. Dr. Eurico Luiz Prospero Ruivo, gostaria de agradecer pela sua experiência e apoio. Suas ideias e direcionamentos foram tremendamente importantes para o desenvolvimento desta dissertação, enriquecendo meu discernimento e perspectiva sobre o cenário acadêmico e também servindo como inspiração para o meu futuro como pesquisador.

Para o Prof. Dr. Leandro Nunes de Castro e Prof. Dr. Pedro Paulo Balbi, sou grato pelo suporte e mentoria que me ajudaram a moldar o tema desta pesquisa.

Ao meu irmão, Matheus Amendoeira Ferraria, expresso meu sincero apreço cujo apoio e contribuições têm sido indispensáveis. Suas sugestões, avaliações e contribuição constantes em relação à fase experimental aumentaram a profundidade e a qualidade deste trabalho.

Ao meu amigo, Lucas Cerqueira Figueira, que me ajudou durante toda esta pesquisa, fornecendo comentários que contribuíram significativamente para a qualidade deste trabalho.

Por fim, expresso minha gratidão aos meus amigos próximos, colegas de trabalho e a todas as pessoas que de alguma forma contribuíram para esta jornada acadêmica, diretamente ou indiretamente. Este trabalho não teria sido possível sem o apoio e contribuições de cada um de vocês.

RESUMO

A emergência de novas áreas de pesquisas e o avanço tecnológico permitem o desenvolvimento e a aplicação de detectores de bordas mais avançados que visam solucionar os problemas dos principais detectores desenvolvidos, como a dificuldade de detecção de bordas soltas e a falta de contexto para extrair informações relevantes de problemas específicos. Com o intuito de aprimorar a tarefa de identificação de bordas e de adaptar a tarefa às propriedades das categorias de imagens de um determinado problema, é estudado um modelo de detector descrito por autômatos celulares de duas dimensões que faz uso de uma fase de otimização abordada por técnicas de meta-heurísticas. A adaptabilidade do detector se dá através da exploração dos parâmetros do modelo e do espaço de busca dos autômatos celulares cujas propriedades e a regra adiram melhor às imagens de entrada. Esse estudo tem como objetivo analisar a robustez da regra obtida pela etapa de otimização, que faz uso do otimizador de enxame de partículas, e a adaptabilidade do modelo para identificar as bordas de um conjunto de imagens natural e de subconjuntos especializados extraídos da mesma base. Os resultados encontrados comprovam a adaptabilidade do modelo estudado para os casos propostos nesta pesquisa e demonstram que a ampliação do espaço de busca não foi efetivo para a base escolhida. As variações dos experimentos demonstraram que independente da validação, o modelo foi capaz de se adaptar à entrada e que as técnicas de *Transfer Learning* aplicadas ao modelo não apresentaram uma melhora significativa na performance.

Palavras-chave: *Autômatos Celulares, Computação Natural, Detecção de Borda, Processamento de Imagem, Transfer Learning.*

Abstract

The emergence of new research areas and technological advancement allow the development and application of more advanced edge detectors that aim to solve the problems of the main detectors developed, such as the difficulty in detecting loose edges and the lack of context to extract relevant information from specific problems. In order to improve the task of identifying edges and adapt the task to the properties of the image categories of a given problem, a detector model described by two-dimensional cellular automata is studied that makes use of an optimization phase addressed by meta-heuristic techniques. The adaptability of the detector occurs through the exploration of the model parameters and the search space of cellular automata whose properties and rule best adhere to the input images. This study aims to analyze the robustness of the rule obtained by the optimization stage, which makes use of the particle swarm optimizer, and the adaptability of the model to identify the edges of a set of natural images and specialized subsets extracted from the same database. The results found prove the adaptability of the model studied for the cases proposed in this research and demonstrate that expanding the search space was not effective for the chosen database. The variations of the experiments demonstrated that regardless of validation, the model was able to adapt to the input and that the *Transfer Learning* techniques applied to the model did not present a significant improvement in performance.

Keywords: *Cellular Automata, Natural Computing, Edge Detection, Image Processing, Transfer Learning.*

Sumário

1	INTRODUÇÃO	1
1.1	Apresentação do trabalho de referência	3
1.2	Justificativa	4
1.3	Organização do Trabalho	5
2	REFERENCIAL TEÓRICO	6
2.1	Autômato Celular de Duas Dimensões	6
2.2	Detecção de Bordas	9
2.2.1	Detector Proposto	9
2.2.2	Detector de Bordas Canny	11
2.3	Meta-heurísticas	12
2.3.1	Otimizador de Enxame de Partículas	13
2.3.2	Otimizador de Enxame de Partículas proposto	16
2.4	<i>Transfer Learning</i>	16
2.4.1	Aplicação no modelo proposto	17
2.5	Validação Cruzada	17
2.5.1	Validação Cruzada em k-pastas	18
2.5.2	Validação Cruzada em Categorias	19
3	METODOLOGIA	20
3.1	Conjunto de dados	20
3.2	Experimentos	25
3.2.1	Análise de robustez da regra encontrada pela etapa de otimização do modelo	25
3.3	Métricas de Avaliação	26
3.3.1	<i>Dice Similarity Score</i>	26
3.3.2	<i>Peak signal-to-noise-ratio</i>	27
3.3.3	<i>Structural Similarity</i>	28
4	RESULTADOS	30
4.1	Análise resultado validação cruzada	30
4.2	Avaliação cruzada nas categorias	30
4.3	Comparação entre os modelos e o Canny	32

5 CONCLUSÕES

56

REFERÊNCIAS BIBLIOGRÁFICAS

63

1 INTRODUÇÃO

A detecção de bordas pode parecer uma tarefa simples, entretanto, seu papel é de suma importância para a área de computação visual. Em muitas de suas aplicações, a tarefa de detecção de bordas é aplicada como uma etapa inicial para se extrair informações relevantes de imagens, servindo como base para aplicações de mapeamento *3D* (BASIRI; LINES; PEREIRA, 2023) e (ARORA et al., 2021), reconhecimento de objetos (RANI; GHAI; KUMAR, 2022) e (YASIR et al., 2022) e segmentação de imagem (AL-AMRI; KALYANKAR; KHAMITKAR, 2010) e (ZHOU; DU; WANG, 2021).

Os detectores de borda possuem duas classes, podendo serem classificados como detectores gerais, aqueles não usam conhecimento prévio para a tarefa e detectores contextuais, aqueles que fazem uso de informações relevantes ou pertinentes a contextos específicos para aprimorar a detecção (ZIOU; TABBONE, 1998). Almejando abordar o problema da tarefa de detecção, diversos modelos de detectores de bordas foram propostos. Os primeiros detectores desenvolvidos são relativamente simples e pertencem à categoria geral, mapeando as bordas através da extração de informações locais como a intensidade dos pixels adjacentes de uma determinada vizinhança. Um ponto fraco destes detectores e de sua classe é a sua necessidade da generalização da tarefa, evidenciado em cenários mais complexos e específicos que requerem a detecção de bordas soltas e a extração de características intrínsecas de categorias de imagens.

Com a evolução dos estudos na área e a presença de computadores potentes, surgiram novos modelos de detectores contextuais que propõem resolver os problemas de generalização enfrentados pelos modelos anteriores combinando o conhecimento de diversas outras áreas de pesquisa como inteligência artificial e autômatos celulares. Exemplos destes novos modelos incluem modelos baseados em redes neurais que aprendem a partir de dados previamente avaliados (WANG, 2016) (PU et al., 2022) e a utilização das propriedades dos autômatos celulares (MOHAMMED; NAYAK, 2014) que elevam a capacidade do processamento e a viabilidade de detectores de bordas contextuais.

O uso de autômatos celulares para a resolução de problemas, dos mais simples aos mais complexos, é um tópico bastante abordado na literatura. Este interesse é resultado de sua natureza discreta e intrinsecamente paralela, que utiliza apenas de informações

locais para se atingir um objetivo global, tornando-os propícios para o desenvolvimento dos mais diversos tipos de sistemas como circuitos (KHAN et al., 1997) (CHO; SWARTZLANDER, 2007) (TORRES et al., 2018), simuladores de eventos reais (SANTÉ et al., 2010), processamento de tarefas computacionais (DUMITRU et al., 2021), simulação do contágio de pânico em uma população dinâmica (WANG et al., 2022), modelagem e geo-simulação das mudanças globais no uso de solos urbanos (ADDAE; DRAGIĆEVIĆ, 2022) e a modelagem em tempo real de sistemas de evacuação de prédios e escolas (JI et al., 2022). A aplicação de autômatos celulares para o processamento de imagens é resultante da presença de características extremamente similares entre ambas áreas como a utilização de uma vizinhança de pixels (informação local) para tarefas como detecção de bordas (objetivo global).

Algumas das implementações de detectores bordas baseados em autômatos celulares podem ser descritos com a aplicação de regras lineares e utilização da vizinhança de Moore (MOHAMMED; NAYAK, 2014). A utilização das regras e da topologia dos autômatos celulares abriu um leque de possibilidades devido aos diversos tipos de comportamentos apresentados pela composição de cada regra, dado o raio de sua vizinhança e os estados de cada célula. Com o intuito de se explorar e usufruir de todas as propriedades e comportamentos dos autômatos, foram desenvolvidos modelos que combinam a aplicação de autômatos celulares com meta-heurísticas, técnicas de exploração baseadas e inspiradas por diversos fenômenos da natureza, com intuito de se explorar o espaço de soluções possíveis e encontrar a solução ótima que apresenta os melhores valores dado uma função objetivo e modelos que fazem uso da aprendizagem de máquina (AM) para se encontrar a melhor composição de regra, conjunto de estados e topologia de um autômato em um espaço de busca, considerando as características específicas de cada imagem (DUMITRU et al., 2020a) (DUMITRU et al., 2020b) (DUMITRU et al., 2021).

O objetivo dessa pesquisa é analisar quantitativamente a robustez e generalização de um modelo de detector de bordas de imagens de cinza adaptável, originalmente desenvolvido para problemas relacionados à imagens médicas, que faz uso de autômatos celulares de duas dimensões e é otimizado pelo Otimizador de Enxame de Partículas, também formalmente conhecido como *Particle Swarm Optimization (PSO)*. Tal análise será feita através da expansão do raio de vizinhança do modelo proposto, da otimização e validação do modelo utilizando técnicas de validação cruzada para encontrar regras ótimas para

a base de imagens *BSDS500* (ARBELAEZ et al., 2011), base que contém 500 imagens com mapas de bordas anotados a mão por mais de uma pessoa. Posteriormente à fase de otimização, o modelo ótimo encontrado será avaliado na mesma base e comparado com o Canny utilizando as métricas propostas, modelo que servirá de base para a análise do desempenho.

1.1 Apresentação do trabalho de referência

Este estudo é inspirado em (DUMITRU et al., 2020b) e (DUMITRU et al., 2020a). O modelo do detector apresentado no trabalho de referência é descrito por um autômato celular de duas dimensões com vizinhança de Moore de raio $r = 1$ otimizado pelo Otimizador de Enxame de Partículas para a tarefa de detecção de bordas em imagens de exames médicos sem anotações aplicando técnicas de processamento em lote e de conceitos de *Transfer Learning*, que consiste em treinar o modelo em um problema generalizado e depois aplicá-lo à um problema relacionado (WEISS; KHOSHGOFTAAR; WANG, 2016). A adaptabilidade do modelo proposto é definida pela otimização da regra do autômato que permite encontrar uma regra que adapte o detector às características de uma imagem. Almejando detecções com melhores qualidades e que consideram a variação de contraste e disparidades dos exames de imagem, o estudo decidiu segmentar as imagens em dois grupos baseados nos contrastes das imagens da base para auxiliar no processo de otimização da regra, gerando uma regra para cada segmento.

A fase de otimização do modelo descrito tem como objetivo encontrar os melhores parâmetros do modelo e a regra linear que melhor se adaptem às características do conjunto de imagens. A otimização ocorre através da utilização do Otimizador de Enxame de Partículas para cada imagem do conjunto de treinamento, conjunto composto por imagens sintéticas geradas a partir da aplicação do filtro gaussiano combinado com a inserção de ruídos gaussianos e distorções que simulam as características das imagens originais. Durante a otimização, o otimizador recebe uma imagem ou um lote de imagens do conjunto de treinamento, otimiza a regra e os parâmetros para a entrada por uma quantidade de iterações parametrizável com o objetivo de encontrar os parâmetros que maximizem o valor da função objetivo para uma imagem ou a média para um lote de imagens. Em seguida, o melhor global é reiniciado para evitar partículas presas e o modelo recebe a

próxima entrada, repetindo o processo de otimização descrito até que o modelo receba todas as imagens.

A função objetivo utilizada para a otimização foi a métrica *Dice Similarity Score* e os parâmetros do modelo são mapeados e normalizados para um espaço contínuo para melhor aderir ao espaço de busca do *PSO*, que atua em espaços reais.

Após a otimização do modelo, o mesmo é avaliado e comparado ao Canny calculando as métricas *Peak Signal-To-Noise-Ratio* e *Structural Similarity* para o conjunto de testes, que é composto pelos exames de imagens originais inalterados. O estudo avalia a robustez da fase de otimização, o impacto do processamento em lotes na etapa de otimização e compara o modelo ao Canny em diversos cenários como integrando fases de pré e pós processamento ao modelo, sem a utilização de etapas de processamento e a adaptabilidade do detector ao incluir ruídos e distorções nas imagens do conjunto de testes.

1.2 Justificativa

Esse estudo tem como objetivo avaliar de maneira quantitativa a adaptabilidade do detector de bordas do modelo do trabalho de referência. A avaliação da adaptabilidade se dará através da análise do espaço de buscas do autômato, explorando uma ampliação do raio e subsequentemente do espaço de busca das regras e da análise da generalização do modelo sob uma base de imagens naturais geral que possui mapas de bordas anotados a mão por várias pessoas.

A avaliação será feita pela aplicação da validação cruzada em k-pastas e da validação cruzada das categorias de imagens definidas no estudo. Para a validação cruzada de categorias, o modelo fará uso de técnicas de *Transfer Learning*, em que um modelo geral será treinado utilizando a base completa e posteriormente, o modelo geral encontrado servirá como ponto de partida para se treinar modelos especializados nas categorias de imagens encontradas na base, como imagens de pessoas, animais, paisagens e objetos, com o intuito de se avaliar se o modelo final encontrado para cada categoria foi capaz de se adaptar melhor às características intrínsecas de cada categoria.

Os trabalhos (DUMITRU et al., 2020b) e (DUMITRU et al., 2020a) foram utilizados como referência para esta pesquisa, portanto, as métricas de avaliação utilizadas serão

as mesmas para permitir maior clareza na avaliação do modelo e na interpretação dos resultados obtidos.

1.3 Organização do Trabalho

Este trabalho está dividido em seis seções. A Seção 1 retrata as motivações da pesquisa e descreve o trabalho utilizado como referência. A Seção 2 apresenta uma contextualização através do referencial teórico introduzindo os principais conceitos de autômatos celulares, da tarefa de detecção de bordas, de técnicas de meta-heurísticas, *Transfer Learning* e validação cruzada. A Seção 3 descreve a metodologia e experimentos aplicados para avaliar o modelo, o conjunto de dados e o pré-processamento utilizado. A Seção 4 apresenta uma análise quantitativa dos resultados obtidos através dos experimentos propostos. A Seção 5 apresenta uma discussão mais abrangente dos resultados em torno da metodologia adotada. A Seção 6 contém as considerações finais, descrevendo pontos importantes observados na pesquisa e que podem ser explorados em trabalhos futuros. Por último, o trabalho é concluído com as referências bibliográficas utilizadas como base para a pesquisa.

2 REFERENCIAL TEÓRICO

2.1 Autômato Celular de Duas Dimensões

Um autômato celular é uma máquina de estado que opera por meio da aplicação de um conjunto de regras de transições de estado a uma célula baseada em sua entrada, nos seus estados anteriores e na sua vizinhança (SAYAMA, 2015). As células de um autômato evoluem com o tempo a partir de uma configuração inicial definida pelo estado inicial das células e cada período consiste na aplicação das regras de transição considerando o estado atual da célula e de seus vizinhos.

Autômatos celulares de duas dimensões são tipicamente representados por meio da vizinhança de Von Neumann e de Moore. Os autômatos com vizinhança de Von Neumann atualizam seu estado de acordo o estado da célula atual e o estados de seus vizinhos à esquerda, superior, inferior e à direita dentro de um raio r , parâmetro responsável por delimitar a distância das células vizinhas à célula central. A vizinhança de Moore (figuras 1a, 1c e 1e) é definida como uma extensão da vizinhança de Von Neumann (figuras 1b, 1d e 1f) ao incluir os vizinhos das posições diagonais à célula central.

Um autômato celular bidimensional tem uma topologia de uma grade retangular infinita. Cada célula da grade é representada por um quadrado de uma grade de dimensão d referenciadas por \mathbb{Z}^d (KARI, 2005) que para duas dimensões, $d = 2$. A definição matemática de um autômato se dá por $A = (S, N, f, d, Y)$, onde:

- S representa o conjunto distinto dos estados das células descrito como $S = \{0, 1, 2, \dots, k - 1\}$ onde $k \in \mathbb{Z}$ e $k \geq 1$; canais
- N representa o vetor da vizinhança, $N = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n)$ onde $\vec{v}_i \in \mathbb{Z}^d$ e n representa o tamanho do vetor de vizinhança;
- $f : S^n \rightarrow S$ é a regra de transição ou função local que recebe uma entrada de tamanho n e a mapeia para um estado;
- d representa a dimensão do autômato onde $d \in \mathbb{Z}$ e $d \geq 1$;
- Y representa o conjunto das células da grade bidimensional;

$(-1,1)$	$(0,1)$	$(1,1)$
$(-1,0)$	$(0,0)$	$(1,0)$
$(-1,-1)$	$(0,-1)$	$(1,-1)$

(a) Moore $r = 1$ (SAYAMA, 2015)

$(-2,2)$	$(-1,2)$	$(0,2)$	$(1,2)$	$(2,2)$
$(-2,1)$	$(-1,1)$	$(0,1)$	$(1,1)$	$(2,1)$
$(-2,0)$	$(-1,0)$	$(0,0)$	$(1,0)$	$(2,0)$
$(-2,-1)$	$(-1,-1)$	$(0,-1)$	$(1,-1)$	$(2,-1)$
$(-2,-2)$	$(-1,-2)$	$(0,-2)$	$(1,-2)$	$(2,-2)$

(c) Moore $r = 2$

$(-3,3)$	$(-2,3)$	$(-1,3)$	$(0,3)$	$(1,3)$	$(2,3)$	$(3,3)$
$(-3,2)$	$(-2,2)$	$(-1,2)$	$(0,2)$	$(1,2)$	$(2,2)$	$(3,2)$
$(-3,1)$	$(-2,1)$	$(-1,1)$	$(0,1)$	$(1,1)$	$(2,1)$	$(3,1)$
$(-3,0)$	$(-2,0)$	$(-1,0)$	$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$
$(-3,-1)$	$(-2,-1)$	$(-1,-1)$	$(0,-1)$	$(1,-1)$	$(2,-1)$	$(3,-1)$
$(-3,-2)$	$(-2,-2)$	$(-1,-2)$	$(0,-2)$	$(1,-2)$	$(2,-2)$	$(3,-2)$
$(-3,-3)$	$(-2,-3)$	$(-1,-3)$	$(0,-3)$	$(1,-3)$	$(2,-3)$	$(3,-3)$

(e) Moore $r = 3$

	$(0,1)$	
$(-1,0)$	$(0,0)$	$(1,0)$
	$(0,-1)$	

(b) Von Neumann $r = 1$ (SAYAMA, 2015)

		$(0,2)$		
	$(-1,1)$	$(0,1)$	$(1,1)$	
$(-2,0)$	$(-1,0)$	$(0,0)$	$(1,0)$	$(2,0)$
	$(-1,-1)$	$(0,-1)$	$(1,-1)$	
		$(0,-2)$		

(d) Von Neumann $r = 2$

			$(0,3)$			
		$(-1,2)$	$(0,2)$	$(1,2)$		
	$(-2,1)$	$(-1,1)$	$(0,1)$	$(1,1)$	$(2,1)$	
$(-3,0)$	$(-2,0)$	$(-1,0)$	$(0,0)$	$(1,0)$	$(2,0)$	$(3,0)$
	$(-2,-1)$	$(-1,-1)$	$(0,-1)$	$(1,-1)$	$(2,-1)$	
		$(-1,-2)$	$(0,-2)$	$(1,-2)$		
			$(0,-3)$			

(f) Von Neumann $r = 3$

Figura 1: Visualização da vizinhanças de Moore e de Von Neumann com as coordenadas indicando as distâncias dos vizinhos da célula central nos eixos x e y .

Um autômato celular de duas dimensões de Moore possui suas regras definidas pelos estados da célula atual e de seus vizinhos que devem ser considerados durante a aplicação da regra de transição. O número de cada regra é definido por uma convenção, as figuras (2a e 2b) exemplificam a numeração das regras para as vizinhanças de Moore de $r = 1$ e $r = 2$, respectivamente. A partir da definição proposta por (DUMITRU et al., 2021), cada célula do autômato é associada a um número que representa o número da regra e a regra é definida pelas células consideradas, ou seja, a regra de vizinhança onde $r \in \{1, 2\}$ que considera somente o elemento central é a regra 1. Para regras que são compostas pela combinação de várias células, o número da regra é definido pela soma dos valores das células em consideração. Por exemplo, para a vizinhança de $r = 1$, a regra com dependência nas células da vizinhança da diagonal esquerda, diagonal a esquerda, a direita e abaixo da célula central é representada como a regra de número 362 ($64+32+8+2+256$) (figura 2c).

64	128	256
32	1	2
16	8	4

1048576	2097152	4194304	8388608	16777216
524288	64	128	256	512
262144	32	1	2	1024
131072	16	8	4	2048
65536	32768	16384	8192	4096

(a) $r = 1$ (DUMITRU et al., 2021)

(b) $r = 2$

64	128	256
32	1	2
16	8	4

(c) Definição da regra 362 para $r = 1$, regra composta pela combinação de várias células.

Figura 2: Representação dos números das regras lineares do autômato de Moore.

2.2 Detecção de Bordas

A detecção de bordas consiste no processo de detectar bordas de uma imagem. Uma borda é detectada quando a diferença entre um pixel e sua vizinhança é considerada abrupta ou significativa, ou seja, é superior a um limiar previamente definido (ZIOU; TABBONE, 1998).

Um detector de bordas tem como entrada uma imagem, de preferência em preto e branco, e o mesmo realiza a análise da imagem em sua totalidade utilizando o critério do limiar mencionado anteriormente. Ao fim de sua execução, um mapa de bordas é gerado contendo informações sobre todas bordas encontradas.

O processo de detecção de bordas é resultado da aplicação de três operações: suavização, diferenciação e classificação da borda (JAIN; KASTURI; SCHUNCK, 1995). Os detectores de borda são classificados (ZIOU; TABBONE, 1998) a partir da maneira como são treinados:

- detectores sem contexto são aqueles que utilizam somente informações locais como os valores de sua vizinhança de pixels
- detectores contextuais são aqueles que utilizam algum conhecimento a priori da tarefa durante o processo de detecção

2.2.1 Detector Proposto

O detector de bordas proposto será referenciado como PSO-CA ao decorrer da pesquisa. O PSO-CA é um detector contextual que utiliza autômatos celulares com vizinhança de Moore onde $r \in \{1, 2\}$ e é representado por uma matriz, 3×3 para $r = 1$ e 5×5 para $r = 2$, com estados binários para extrair mapas de bordas de imagens em tons de cinza. O modelo proposto faz uso de regras de transições lineares que são definidas por operações *XOR* entre as células (KHAN et al., 1997).

O detector define os vizinhos de um pixel (x_0, y_0) em uma imagem bidimensional como:

$$N(x_0, y_0) = \{(x, y) : |x - x_0| \leq r, |y - y_0| \leq r\} \quad (1)$$

onde r representa o raio da vizinhança, nesta pesquisa $r \in \{1, 2\}$.

A partir da definição descrita no referencial teórico, o autômato do detector proposto, independente do raio de vizinhança, é descrito por:

- conjunto de estados definido como $S = \{0, 1\}$
- dimensão como $d = 2$;
- seu espaço celular $Y = \mathbb{Z}^2$;
- uma configuração é definida como uma função $c : \mathbb{Z}^2 \rightarrow \{0, 1\}$. Para uma imagem $A \times B$ temos $X = \{0, \dots, A-1\} \times \{0, \dots, B-1\}$ e uma configuração é uma função $c : \{0, \dots, A-1\} \times \{0, \dots, B-1\} \rightarrow \{0, 1\}^{(A \times B)}$

Seguindo a definição descrita, o autômato com raio de vizinhança $r = 1$ tem sua vizinhança descrita por:

- sua vizinhança de tamanho $n = 9$ definida como $N = ((0, 0), (1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0), (-1, 1), (0, 1), (1, 1))$

O autômato com raio de vizinhança $r = 2$ tem sua vizinhança descrita por:

- sua vizinhança de tamanho $n = 25$ definida como $N = ((0, 0), (1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0), (-1, 1), (0, 1), (1, 1), (2, 1), (2, 0), (2, -1), (2, -2), (1, -2), (0, -2), (-1, -2), (-2, -2), (-2, -1), (-2, 0), (-2, 1), (-2, 2), (-1, 2), (0, 2), (1, 2), (2, 2))$

Para o autômato, é utilizado uma regra de transição de duas fases definida em (UGUZ; SAHIN; SAHIN, 2015) que tem a sua primeira fase definida por:

$$\mu_{i,j} = \frac{\phi(X_{i,j})}{\Delta + \phi(X_{i,j})} \quad (2)$$

em que X_{ij} representa um pixel de uma imagem de escala de cinza X , uma matriz de dimensão $A \times B \times 1$, na linha i e coluna j , que são mapeados como células do autômato com X_{ij} como célula central, $\Delta \in \{0, \dots, 255\}$. $\phi(X_{i,j})$ é definido como:

$$\phi(X_{i,j}) = \sum_k \sum_q |X_{ij} - X_{i+k, j+q}| \quad (3)$$

onde k e q são todos valores no intervalo $[-r, r]$.

O segundo passo da regra de transição é definida por uma função de limiar $F : X \mapsto \{0, 1\}$ que retorna o estado binário, descrita por:

$$F(X_{i,j}) = \begin{cases} 1, & \text{se } \mu(X_{i,j}) > \tau \\ 0, & \text{se } \mu(X_{i,j}) \leq \tau \end{cases} \quad (4)$$

em que $\tau \in [0, 1]$.

O modelo tem 3 hiper-parâmetros que precisam ser ajustados para cada problema. Δ um parâmetro inversamente proporcional ao número de bordas detectadas, a regra linear z que define os vizinhos que a regra deve considerar e τ que controla quantos pontos são definidos como bordas. Tal modelo é combinado com o algoritmo de otimização *Particle Swarm Optimization* para otimizá-lo para as características das imagens.

Com o intuito de se preservar as dimensões originais das imagens, o modelo fará uso do preenchimento de bordas com valor 0 para cada dimensão antes da aplicação da primeira regra de transição. As bordas acrescentadas às imagens resultarão em imagens cujo o tamanho final será equivalente ao acréscimo de r nos eixos X e Y das imagens.

2.2.2 Detector de Bordas Canny

O detector de bordas Canny é um detector geral de referência (CANNY, 1986) que serve como base e métrica para o desenvolvimento de novos modelos (DUMITRU et al., 2020b). O Canny adere a três critérios ótimos para a tarefa de detecção: uma boa detecção das bordas presentes, localização das bordas e uma única resposta a uma borda. O detector consiste de várias etapas: a aplicação do filtro gaussiano na imagem original para a redução de ruídos e cálculo da intensidade dos gradientes e depois a detecção de pontos máximos do gradiente da imagem.

A partir dessa detecção, tais pontos máximos passam por uma função de supressão de duplo limiar. Valores acima dos limiares são classificados como bordas fortes, valores que se encontram no meio dos limiares são classificados como bordas fracas e valores menores são descartados. Após a classificação das máximas, o mapa é gerado selecionando somente as bordas fortes e as bordas fracas próximas ou conectadas as bordas fortes. (CANNY, 1986).

O detector Canny possui 3 parâmetros ajustáveis, a largura do filtro gaussiano, representado por σ , o limiar inferior e o limiar superior para a função de duplo limiar.

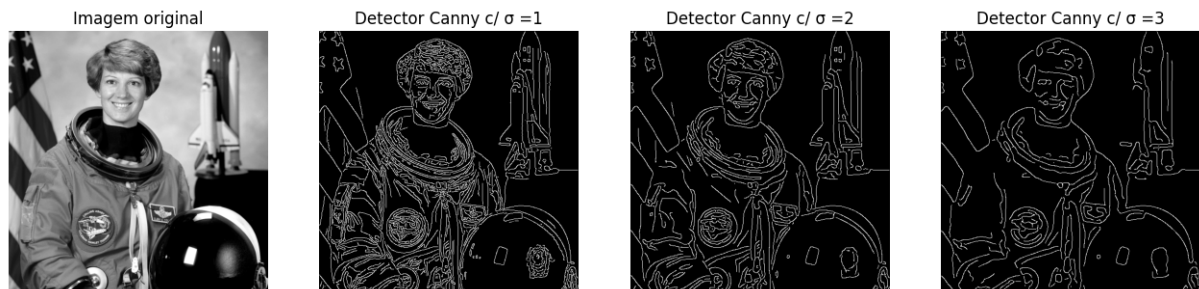


Figura 3: Aplicação da implementação do Canny da biblioteca *skimage* (WALT et al., 2014a) (WALT et al., 2014b) com a variação do parâmetro σ à imagem do Astronauta do *skimage* em tons de cinza.

2.3 Meta-heurísticas

Meta-heurísticas são um conjunto de heurísticas abstratas, independente do domínio do problema, que trabalham no nível meta para otimizar uma variedade de problemas de otimização combinatorial. O conjunto é descrito pela busca iterativa para encontrar a melhor solução de um problema em um espaço de soluções. As técnicas de meta-heurística não possuem uma norma ou padrão de categorização, entretanto, existe um consenso de categorias (BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

As meta-heurísticas são formalizadas por $P = (E, f, \Omega)$, em que P representa o problema de otimização, E o espaço de busca do domínio do problema, f a função objetivo, também conhecida como *fitness*, e Ω o conjunto de limites do problema. O espaço de busca é definido por múltiplas variáveis $\{var_1, var_2, \dots, var_n\}$, representando as dimensões ou variáveis do problema que precisam ser exploradas na busca. As restrições restringem as soluções que devem ser exploradas ao avaliar se a mesma é válida para o problema. A função objetivo f é usada para avaliar a qualidade de cada solução e direcionar o processo de busca (PERES; CASTELLI, 2021).

Dois processos são fundamentais na definição de uma técnica de meta-heurísticas, a exploração e diversificação. Uma técnica é apropriada para a otimização e procura de soluções ótimas se conseguir balancear a sua exploração e diversificação. A exploração é

responsável por intensificar a busca em uma área considerada ótima e a diversificação é responsável por identificar e explorar áreas com possíveis soluções ótimas. O balanceamento dos processos ajuda a explorar áreas promissoras e instigar a intensificação para evitar que o espaço de buscas fique concentrado em um único lugar e acabe ficando preso nele.

Uma técnica de otimização é o otimizador de enxame de partículas, uma heurística bio-inspirada que faz uso do conceito da inteligência de enxame de uma população e é bastante aplicada ao problema de busca em autômatos (DUMITRU et al., 2020a).

2.3.1 Otimizador de Enxame de Partículas

O Otimizador de Enxame de Partículas mais conhecido do seu termo em inglês *Particle Swarm Optimization* será referido ao longo do artigo como (*PSO*) para facilitar a leitura. O *PSO* é um algoritmo de otimização bio-inspirada no comportamento social de uma população de enxame de partículas, em que cada partícula representa uma possível solução. Uma partícula i é definida pela sua posição x_i no espaço e por sua velocidade v_i e ambas influenciam na movimentação da mesma na fase de otimização (KENNEDY; EBERHART, 1995).

O *PSO* é executado por uma quantidade parametrizável de iterações e a cada iteração, as partículas tem suas posições e velocidades influenciadas por uma função objetivo que avalia a solução de cada partícula.

A otimização é obtida a partir da exploração de soluções definida pela movimentação das partículas no espaço de busca através de sua experiência e de interações entre partículas vizinhas.

Uma partícula i tem sua posição em um tempo t da iteração definida pela sua velocidade que é calculada usando a seguinte fórmula:

$$v_i(t + 1) = w * v_i(t) + r_1 * c_1 * (p_{best_i} - x_i(t)) + r_2 * c_2 * (g_{best} - x_i(t)) \quad (5)$$

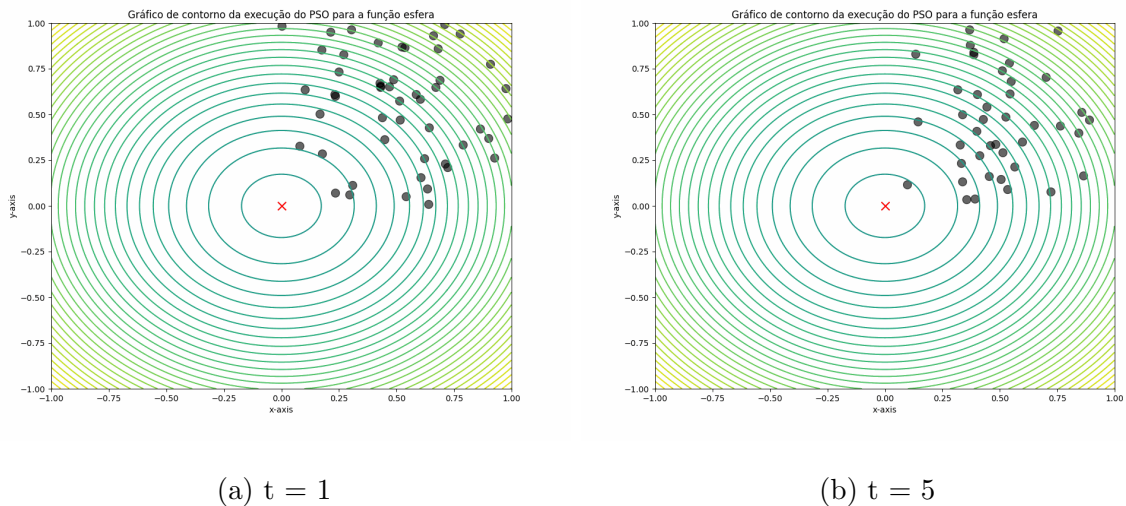
onde w controla a oscilação da partícula p , p_{best} é a melhor posição da partícula i , g_{best} é a melhor posição global do enxame, c_1 é o coeficiente referente a influência própria da partícula, c_2 é o coeficiente do fator de influência do enxame e $r_1, r_2 \in (0, 1)$ são variáveis uniformes aleatórias. $x_i(t)$ representa a posição da partícula i no tempo t e é definida

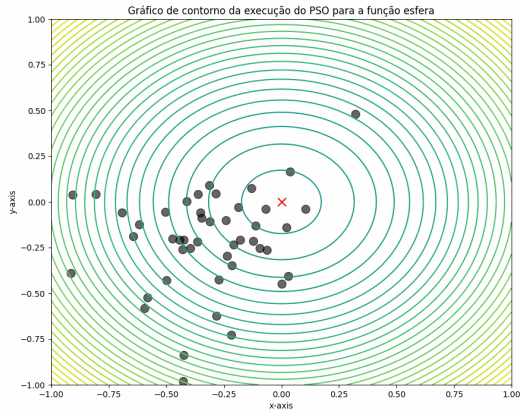
pela fórmula:

$$x_i(t + 1) = v_i(t + 1) + x_i(t) \quad (6)$$

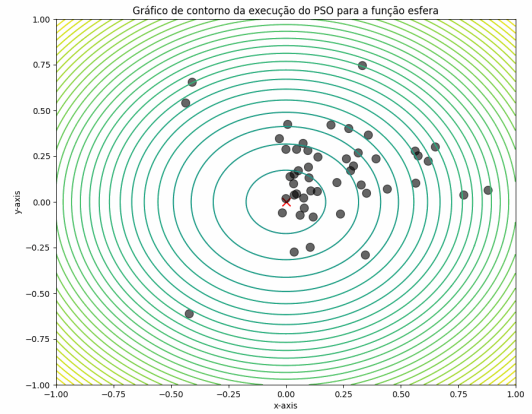
As partículas vão se movendo pelo espaço de busca ao longo das iterações e tendem a se aproximar da posição que representa o ótimo global quando otimizadas. É possível observar a evolução da população de enxame ao se aplicar o algoritmo a uma função que se deseja otimizar. Para a visualização, foi executado um exemplo utilizando a biblioteca *pyswarm* que implementa o *PSO* e sua documentação (MIRANDA, 2018). A execução buscou otimizar a função da esfera, uma função que também é implementada pela biblioteca e que tem como objetivo encontrar o melhor valor em um espaço bidimensional que tem como mínima $x = 0$ e $y = 0$, denotados com um 'x' vermelho na figura (4a). O exemplo da otimização da visualização foi executado com os parâmetros e iterações utilizados na própria documentação da biblioteca: $iter = 100$, $c_1 = 0.5$, $c_2 = 0.3$, $w = 0.9$.

Figura 4: Visualização dos passos descritos por t da execução do *PSO* com 100 iterações para a otimização da função objetivo da esfera.

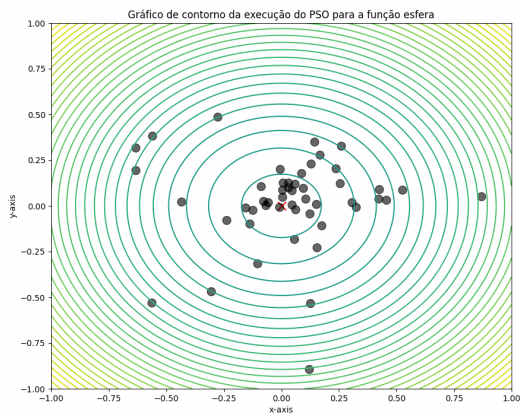




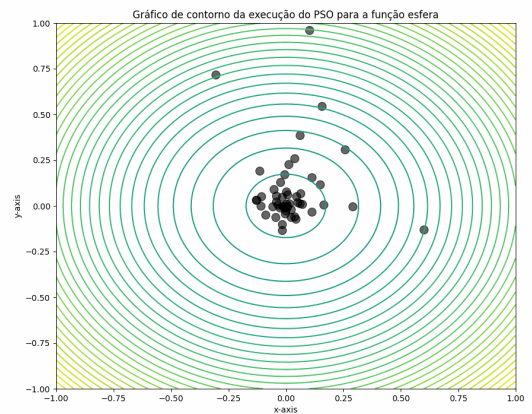
(c) $t = 10$



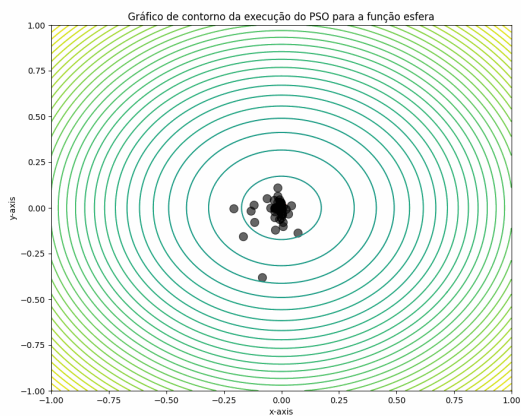
(d) $t = 15$



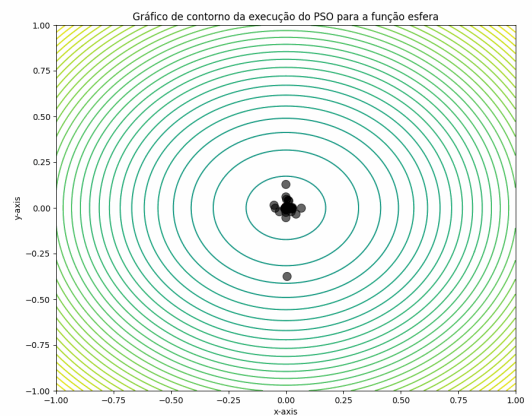
(e) $t = 25$



(f) $t = 50$



(g) $t = 75$



(h) $t = 100$

2.3.2 Otimizador de Enxame de Partículas proposto

O *PSO* proposto é uma adaptação do otimizador do modelo de referência. O *PSO* vai otimizar os parâmetros do modelo visando encontrar o melhor global através da função objetivo definida na seção de métricas. Para isso, para cada amostra da base, o objetivo é otimizar a tripla de parâmetros do autômato (Δ, τ, z) onde Δ e τ são definidos na seção do modelo proposto e o número da regra linear, $z \in \{0, \dots, z_{r_{max}}\}$ em que $z_{r_{max}}$ equivale ao número máximo da regra, seguindo a convenção definida na seção do modelo proposto, para o raio r e pode ser calculado através da fórmula $(2^{(1+2 \times r)^2}) - 1$. Para adequar os valores aos domínios contínuos do *PSO*, a tripla é normalizada no intervalo $[0, 1]$ e a representação final é descrita por (Δ', τ, z') onde $z' = z/z_{r_{max}}$ e $\Delta' = \Delta/255$.

Com o intuito de se generalizar para a base inteira, o *PSO* proposto fará uso do conjunto de processamento em lotes tal como no modelo de referência. Entretanto, como a pesquisa não tem como objetivo analisar o impacto do tamanho do lote, o mesmo será restringido a usar todas as imagens recebidas como entrada como um lote único e cada partícula terá seu valor calculado como a média do resultado da função objetivo aplicada ao lote de imagens.

2.4 *Transfer Learning*

Transfer Learning é uma técnica de aprendizagem de máquina que permite transferir o conhecimento de um modelo base já treinado em um domínio para um modelo alvo que possui um domínio relacionado. A transferência de conhecimento tem como inspiração a aprendizagem humana e tem como objetivo complementar o conhecimento de um modelo e mitigar o problema de bases escassas para a fase de treinamento. Além dos problemas mencionados anteriormente, a técnica também visa melhorar a performance de um modelo alvo, uma vez que os domínios são relacionados, o modelo alvo pode continuar o treinamento do modelo base para se especializar no domínio (SAMMUT; WEBB, 2011) (WEISS; KHOSHGOFTAAR; WANG, 2016)

2.4.1 Aplicação no modelo proposto

Assim como proposto pelo modelo de referência (DUMITRU et al., 2020a) e (DUMITRU et al., 2021), a aplicação do *Transfer Learning* se dará através de um processo de otimização em duas etapas. A primeira etapa consiste em treinar um modelo, referenciado como modelo geral, na base de dados por inteira. Após o treinamento, a população final do *PSO* será exportada e utilizada como população inicial em uma segunda etapa de treinamento, em que o modelo, a partir da população final, será treinado para as categorias específicas encontradas na etapa de pré-processamento da base de dados.

2.5 Validação Cruzada

A validação cruzada é um conjunto de técnicas de amostragem que tem como objetivo avaliar o desempenho geral de modelos supervisionados, otimizar os parâmetros de modelos para uma tarefa e de se aumentar o conjunto de amostras de uma base de dados.

Sua aplicação tem como propósito evitar os problemas de *overfitting* e *underfitting* que um modelo pode sofrer ao se utilizar uma base pequena com pouca variação e com problemas de divisão de amostras para as etapas de treinamento, validação e testes (BERRAR, 2019), (BROWNE, 2000) e (BROADHURST; KELL, 2006).

A amostragem é um processo que recebe um conjunto de dados de uma base por inteiro e a divide em conjuntos de treinamento, validação e testes, aplicando técnicas que podem considerar a estratificação das classes e repetições de dados na divisão dos conjuntos. O conjunto de treinamento e validação são utilizados durante a etapa de treinamento e otimização do modelo e o conjunto de testes é utilizado para se testar a performance do modelo final. Essa divisão de conjuntos visa auxiliar e remediar os problemas mencionados anteriormente, *overfitting* e *underfitting*, ao gerar conjuntos diferentes para cada fase.

O *overfitting* é um problema que um modelo apresenta resultados questionáveis. Isso pode ocorrer nos seguintes cenários: ao se treinar um modelo em uma base pequena, ao se utilizar a mesma base tanto para o treinamento quanto para a etapa de avaliação e ao se utilizar uma base com pouca variação. Tal problema faz com o que o modelo final apresente resultados bons somente para dados conhecidos e o tornando incapaz de

generalizar a tarefa para dados não conhecidos.

O problema de *underfitting* também impacta a generalização de um modelo e ocorre quando a base não é o suficiente para descrever o problema e os relacionamentos das propriedades dos dados, gerando um modelo final que é incapaz de apresentar resultados bons para quaisquer cenários.

2.5.1 Validação Cruzada em k-pastas

A Validação Cruzada em k-pastas é uma técnica de validação que consiste em dividir aleatoriamente uma base de dados em k pastas mutuamente exclusivas e de aproximadamente mesmo tamanho (BERRAR, 2019) (BROADHURST; KELL, 2006). O termo “pastas” do nome da técnica se refere as subconjuntos que a base será particionada e tem sua quantidade representada pelo parâmetro k . A partição de um conjunto em k “pastas” ocorre através da seleção de amostras aleatórias sem reposição. A validação de um modelo por essa técnica se dá através do treinamento do mesmo utilizando $k - 1$ subconjuntos, que juntos formam o conjunto de treinamento. Depois da fase de treinamento, o modelo tem seu desempenho avaliado no subconjunto restante, representado como o conjunto de validação. O processo de validação é repetido por k iterações até que todas as k pastas tenham sido utilizadas como conjunto de validação. A performance final da validação é a media entre os resultados obtidos na etapa de validação para cada k pasta.



Figura 5: Visualização da validação cruzada em k-pastas com $k = 10$. Imagem de Karl Rosaen (Karl Kosaen, 2016)

2.5.2 Validação Cruzada em Categorias

A Validação Cruzada em Categorias mencionada na pesquisa é descrita pelo treinamento de um modelo utilizando uma categoria e depois validado nas categorias restantes. O processo é repetido até que cada categoria tenha sido utilizado como conjunto de treinamento.

3 METODOLOGIA

Nesta pesquisa, os experimentos realizados visam estudar o espaço de busca do detector proposto e como uma ampliação do raio de vizinhança de Moore, tal que $r \in \{1, 2\}$, pode auxiliar na busca de uma regra ótima e como seu resultado se compara com o espaço do modelo originalmente proposto. Também será analisada a robustez da adaptabilidade do detector utilizando técnicas de *Transfer Learning*, o modelo será treinado e validado utilizando a base selecionada e após encontrar uma regra ótima geral, o modelo do detector passará por uma segunda fase de treinamento, na qual utilizará a população final do *PSO* da regra ótima encontrada como população inicial para subseqüentes treinamentos com o objetivo de se adaptar o modelo para as subcategorias da base anotadas para esta pesquisa. O impacto do *Transfer Learning* no modelo também será avaliado, serão comparados os desempenhos de modelos especializados em cada categoria com modelos treinados exclusivamente em uma categoria específica. A avaliação do modelo será feita utilizando as métricas informadas em conjunto com técnicas de validação cruzada em categorias e da validação cruzada em k-pastas.

3.1 Conjunto de dados

O conjunto de dados utilizado é o BSDS500 (ARBELAEZ et al., 2011), um conjunto de imagens naturais coloridas com mapas de bordas e segmentação rotulados a mão bastante popular em estudos de detecção de bordas e segmentação de imagem. Uma amostra da base pode ser vista na figura 6.

O BSDS500 é composto por 500 imagens, sendo separadas originalmente em 200 imagens para treinamento, 100 para validação e 200 para testes. Cada imagem da base tem um mapa de bordas rotulado de 4 a 9 pessoas como demonstrado para uma imagem da base na figura 7. Os mapas de bordas anotados da base são armazenados em arquivos *MATLAB* e agrupados por foto. Para a pesquisa, os mapas foram extraídos com o auxílio da biblioteca *scipy* (VIRTANEN et al., 2020) e convertidos para *PNG*.

Devido à natureza das imagens, dos rótulos da base e das variações nas anotações por anotador, é comum na literatura se gerar um mapa de probabilidade para cada imagem a partir da média da classificação entre todos os classificadores. A probabilidade das bordas

variam entre 0 e 1, sendo 0 para um consenso geral de que o pixel não é uma borda e 1 de que o pixel é uma borda. Através do mapa de probabilidade, é possível definir um limiar, bordas classificadas com probabilidades acima de tal limiar são consideradas bordas e abaixo são desconsideradas (PU et al., 2022), (LIU et al., 2017), (HE et al., 2019).



(a)



(b)



(c)



(d)

Figura 6: Visualização de uma amostra de 4 imagens inalteradas da base BSDS500.

Além da extração do mapa de bordas das anotações pelo limiar de probabilidade, as imagens e anotações passarão por uma fase de pré-processamento com o intuito de adequá-las à entrada do modelo e de agilizar e facilitar o processamento do mesmo, permitindo que todas as imagens sejam carregadas em lotes de processamento por bibliotecas como *pytorch* (PASZKE et al., 2019). O pré-processamento é descrito pela conversão das imagens

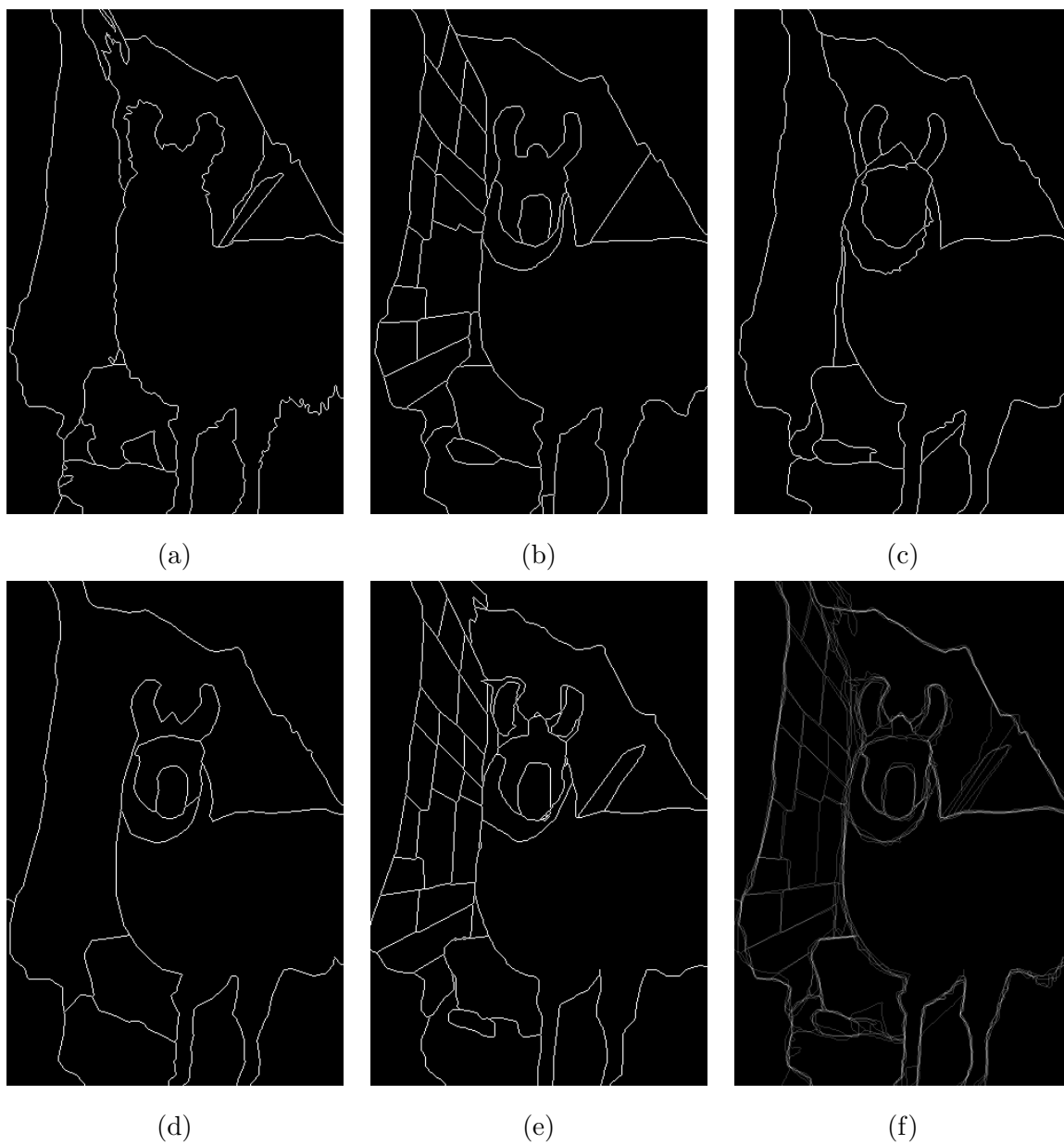


Figura 7: Visualização dos mapas de bordas da base anotados a mão por 5 pessoas (figuras 7a, 7b, 7c, 7d, 7e) e o mapa de probabilidade de bordas extraído das diferentes anotações (figura 7f).



Figura 8: Visualização das 4 imagens da base ilustradas anteriormente após a etapa de pré-processamento

coloridas para imagens em tons de cinza, seguida pelo redimensionamento para 128×128 mantendo a proporção original e restringindo a dimensão do lado maior, e a padronização da orientação das imagens. Apesar das imagens da base possuírem a mesma resolução final, parte da base está com a orientação de retrato e parte com a orientação de paisagem. Uma amostra das imagens da base após a etapa de processamento é exibida na figura 8 e suas respectivas anotações do mapa de bordas extraídos para as probabilidades 0.02 e 0.3 são exibidas na figura 9.

Com o objetivo de explorar a adaptabilidade do modelo às características intrínsecas de imagens, a divisão de conjuntos da base original será desconsiderada e a base passará por uma nova etapa de divisão em que a mesma será dividida em subconjuntos considerando o que está em foco na imagem. As quatro categorias encontradas na base foram: animais, pessoas, objetos e paisagens. A divisão das 500 imagens entre as categorias encontradas ficou como: 195 imagens de animais, 100 imagens de paisagens, 54 imagens de objetos e 151 imagens de pessoas.



(a)



(b)



(c)



(d)



(e)



(g)



(h)

Figura 9: Visualização da extração do mapa de probabilidades das imagens anteriores para os limiares de probabilidade 0.02 (figuras 9a, 9b, 9c, 9d) e 0.3 (figuras 9e, 9f, 9g, 9h)

3.2 Experimentos

3.2.1 Análise de robustez da regra encontrada pela etapa de otimização do modelo

A partir da definição do modelo proposto pela pesquisa, o detector foi implementado utilizando a biblioteca *pytorch* (PASZKE et al., 2019) e foi otimizado utilizando a implementação do *PSO* global da biblioteca (MIRANDA, 2018) para a base *BSDS500* pré-processada. Para a validação cruzada em k-pastas foi utilizada a implementação da biblioteca *scikit-learn* (PEDREGOSA et al., 2011).

Para todos os experimentos definidos nessa pesquisa, os mapas de bordas da base pré-processada utilizados serão extraídos para os limiares de probabilidade igual à 0,02. O limiar de probabilidade será referido ao decorrer do texto como p .

O modelo será avaliado usando os seguintes casos de estudos:

1. buscar o τ , o Δ e a regra ótima do autômato para $r \in \{1, 2\}$ em conjunto com a validação cruzada em k-pastas onde $k = 10$;
2. buscar o τ , o Δ e a regra otimizada para $r = 1$ usando como conjunto de treinamento todas as imagens e guardar a população final e a melhor regra;
 - (a) testar a regra ótima encontrada do passo 2 no conjunto de todas as imagens;
 - (b) testar a regra ótima encontrada do passo 2 no conjunto de cada categoria separadamente;
3. buscar o τ , o Δ e a regra otimizada para $r = 2$ usando como conjunto de treinamento todas as imagens e guardar a população final e a melhor regra.
 - (a) testar a regra ótima encontrada do passo 3 no conjunto de todas as imagens;
 - (b) testar a regra ótima encontrada do passo 3 no conjunto de cada categoria separadamente;
4. usando como população inicial a população final do passo 2, buscar separadamente para cada categoria o τ , o Δ e a regra do autômato ótimos do mesmo espaço e avaliar para todas as categorias;

5. usando como população inicial a população final do passo 3, buscar separadamente para cada categoria o τ , o Δ e a regra do autômato ótimos do mesmo espaço e avaliar para todas as categorias;
6. fazer uma busca em cada categoria (separadamente, e sem usar as regras encontradas anteriormente na busca) por uma regra tal que o raio $r \in \{1, 2\}$ e testar a regra encontrada em cada categoria para todas as categorias;

O objetivo desse processo é usar diversas amostras da mesma base para avaliar a robustez da tarefa de detecção de borda. Após as execuções dos testes, os resultados das métricas de avaliações selecionadas serão comparados entre o detector do modelo proposto otimizado para cada valor de r e o detector de bordas Canny implementado pela biblioteca *skimage* usando os parâmetros base de sua implementação, em que $\sigma = 1$, o limiar inferior e superior equivalem à 10% e à 20% do valor máximo do tipo da imagem, que para a pesquisa é *int8* e equivale à 255, o valor máximo de imagens em tom de cinza.

3.3 Métricas de Avaliação

3.3.1 *Dice Similarity Score*

Para se avaliar o *fitness* das partículas quando aplicadas a uma regra, calcula-se o *Dice Similarity Score (DSC)* a partir do mapa de bordas anotado e do mapa de bordas extraído pela regra atual. A métrica é pode ser descrita pela fórmula:

$$DSC = (2 \cdot VP) / (2 \cdot VP + FP + FN) \quad (7)$$

Para o problema de detecção de borda, a classe positiva é definida como uma borda e a negativa como não borda. Os termos da fórmula anterior são descritos como:

- *VP* (Verdadeiro Positivo) representa que o detector classificou corretamente o pixel da imagem como uma borda;
- *FP* (Falso Positivo) representa que o detector classificou incorretamente o pixel da imagem como uma borda;
- *FN* (Falso Negativo) representa que o detector não classificou o pixel da imagem como uma borda e na anotação da imagem o pixel é classificado como uma borda;

Essa métrica, originalmente desenvolvida na área de ecologia para medir a associação entre espécies (DICE, 1945), é bastante utilizada em pesquisas na área de segmentação e detecção de bordas em imagens com o intuito de se avaliar a similaridade entre dois conjuntos de dados, o mapa extraído pelo modelo e o mapa real. Tal fórmula se torna interessante ao problema uma vez que diferente da métrica de acurácia, ela considera o desbalanceamento das classes do problema de detecção. A métrica foi implementada usando funções e manipulações providas pela biblioteca *pytorch*.

3.3.2 *Peak signal-to-noise-ratio*

A qualidade das bordas detectadas é avaliada através da métrica *Peak signal-to-noise-ratio* que sinaliza a proporção de ruído à quantidade de informação da imagem original (HORÉ; ZIOU, 2013). A fórmula é definida por:

$$PSNR(f, g) = 10 \log_{10} \left(\frac{R^2}{EQM(f, g)} \right) \quad (8)$$

em que (EQM) é descrito por:

$$EQM(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \quad (9)$$

Na fórmula descrita, f corresponde à imagem original e g ao mapa de bordas detectado, ambas com dimensões $M \times N$. R corresponde ao valor máximo de variação da imagem. Para esse estudo que trabalha somente com imagens de tons de cinza R será considerado como 255.

O cálculo da Erro Quadrático Médio (EQM) compara os valores da imagem original com a variação da imagem e o valor resultante representa a quantidade de ruído na variação da imagem. Valores altos da métrica indicam que a taxa de erro é baixa e existe pouco ruído entre as informações da imagem. Apesar de ser uma métrica bastante utilizada na área de compressão e recuperação de informações em imagens, a mesma também é aplicada para avaliar detectores de bordas (PADMAVATHI; SUBASHINI; LAVANYA, 2009) (POOBATHY; CHEZIAN, 2014) (DUMITRU et al., 2021) e segmentações de imagem (XESS; AGNES, 2014).

Para esta pesquisa, a métrica foi implementada usando funções auxiliares da biblioteca *numpy* (HARRIS et al., 2020).

3.3.3 Structural Similarity

A métrica *Structural Similarity (SSIM)* avalia a similaridade entre o resultado do detector e o mapa das bordas (WANG et al., 2004). O valor da métrica é definida no intervalo $[-1, 1]$, valores igual a 1 indicam que ambas imagem e o mapa das bordas são similares, 0 representa que não existe similaridade e -1 indica a existência de uma correlação negativa na similaridade. A fórmula da métrica é descrita por:

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) \quad (10)$$

em que l (eq. 11) corresponde a comparação da luminância, c (eq. 12) ao contraste e s (eq. 13) a estrutura das imagens x e y .

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (11)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (12)$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (13)$$

onde $\mu_{x,y}$ correspondem a média local de x e y , valor médio de um conjunto:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (14)$$

A fórmula também serve para y , basta substituir os termos x por y .

σ_{xy} representa a covariância local entre x e y , a interdependência e o relacionamento linear entre duas variáveis:

$$\sigma_{xy} = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \right) \quad (15)$$

$\sigma_{x,y}^2$ descreve a variância, a dispersão dos valores de um conjunto em a relação a sua média, de x e y . A fórmula para ambos, vide a substituição da variável, pode ser definida como:

$$\sigma_x^2 = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right) \quad (16)$$

$\sigma_{x,y}$ é o desvio padrão e é definido como a raiz quadrada da variância de um conjunto:

$$\sigma_x = \sqrt{\sigma_x^2} \quad (17)$$

Os termos C_1 , C_2 e C_3 são constantes. Para esta pesquisa, a implementação utilizada da métrica foi a da biblioteca *skimage* (WALT et al., 2014b).

4 RESULTADOS

4.1 Análise resultado validação cruzada

A figura (10) e tabela (1) apresentam os resultados da validação cruzada de k-pastas para o caso de estudo 1. Pelos resultados, é possível perceber que existe uma variação considerável da média para cada iteração e a presença de alguns *outliers*, valores bem distantes da média, para ambas as métricas escolhidas.

Apesar das variações nos valores nas duas métricas, uma constante foi o modelo com $r = 1$ apresentar valores médios melhores quando comparado ao modelo com $r = 2$ (figura 10a). Ao avaliar a variação do resultado da função objetivo para partícula encontrada no final do treinamento para cada iteração do k (figura 11 e tabela 3), é possível perceber que apesar das diferenças nos valores de *DSC*, ambos os raios apresentaram as mesmas variações para cada pasta. A tabela 3 destaca a flexibilidade que os parâmetros podem causar na regra encontrada, as pastas das iterações 0, 6 e 7 encontraram a mesma regra (350) e a variação dos valores de Δ e τ fez com que o modelo se adaptasse as imagens de cada pasta.

As figuras (12b e 12a) apresentam a comparação entre o modelo Canny e o modelo proposto, o PSO-CA. O modelo proposto teve uma variação maior para a métrica *PSNR* quando comparado ao Canny, entretanto, ainda assim apresentou resultados melhores para a maior parte dos casos. Já para a métrica *SSIM*, o modelo proposto foi capaz de apresentar resultados consideravelmente melhores em todas as iterações.

4.2 Avaliação cruzada nas categorias

Avançando para os casos de estudos 4 e 5 que são relacionados à exploração da regra e à adaptação do modelo para as categorias da base preprocessada. As figuras (13 e 14) e tabela 4 apresentam os resultados das métricas escolhidas, comparando o modelo geral e o modelo especializado de cada categoria para cada subconjunto da base de dados e a tabela 5 apresenta os resultados obtidos pelo Canny. Para os casos de estudos 4 e 5, os resultados da aplicação do *Transfer Learning* foram representados pelo prefixo *TF* no nome de cada categoria.

Ao comparar os modelos encontrados com o Canny (tabela 5), os modelos propostos foram capazes de apresentar resultados melhores tanto para o *PSNR* quanto para o *SSIM*, indicando que os mesmos foram capazes de se adaptar as categorias individuais das imagens.

Os resultados obtidos também indicam que o modelo de categoria geral e o modelo especializado de cada categoria foram capazes de encontrar regras diferentes (tabela 6) e em alguns casos, a nova regra se adaptou melhor à categoria como para os modelos de animais e objetos. Ao comparar os valores de *DSC* da regra encontrada na etapa de otimização do modelo geral com os modelos especializados, percebe-se que o modelo especializado em animais apresentou valores do *DSC* abaixo do modelo geral e mesmo assim teve resultados bons para as métricas de avaliações. Também é possível perceber pela tabela (6) que os modelos com $r = 2$ geral e os especializados nas categorias animais e objetos apresentaram valores próximos para os 3 parâmetros, indicando que a partícula pode ter ficado presa durante a fase de otimização e não conseguiu encontrar regras melhores. Outro ponto que pode ter influenciado esse resultado é a proporção das bases, os subconjuntos da categoria de animais e de pessoas compõem 39% e 30% da base.

Avaliando a adaptabilidade do modelo, os modelos de animal e objetos foram capazes de se adaptar um pouco melhor a própria categoria, apresentando resultados melhores nas métricas de avaliação para própria categoria e também para as categorias restantes quando comparado ao modelo geral (figuras 14 e 13).

Os resultados do caso de estudo 6 (figuras 13 e 14) apresentam a avaliação dos experimentos dos modelos individuais, modelos que não fazem o uso de *Transfer Learning*. A partir dos resultados encontrados, é possível perceber que mesmo sem a transferência de conhecimento do domínio comum e a presença de menos dados de treinamento, os modelos também foram capazes de se adaptar as características das imagens (tabela 8), apresentando bons resultados de *PSNR* e *SSIM* (tabela 7).

Para esse experimento, o modelo de animal também apresentou os melhores resultados em ambas as métricas para todas as bases e o modelo de paisagens também apresentou os piores resultados.

Similar aos modelos especializados, os modelos com $r = 1$ também apresentaram melhores resultados no geral e foram capazes de performar melhor que o Canny independente

do raio e categoria de treinamento do modelo (figura 21a).

Um ponto interessante deste experimento é que os modelos encontrados neste experimento tiveram resultados bem próximos quando comparados com o experimento 4 e 5. Tal comportamento pode ser evidenciado através das figuras (22 e 23) e tabela (7).

Os modelos especializados com técnicas de *Transfer Learning* quando comparados aos modelos individuais apresentaram poucas variações (figuras 22 e 23 e tabela 9). Os resultados da fase de otimização reafirmam esse comportamento, apresentando valores próximos para o *DSC* (tabela 10).

Um ponto de observação é que modelo especializado e individual treinados na categoria de paisagens apresentou os piores resultados médios das métricas para todas as categorias para ambos os raios avaliados. Tais resultados dos modelos de paisagem podem ser explicados pelas características da categoria. Paisagens tendem a apresentar mapas de bordas com características bem diferentes de rostos, animais e objetos.

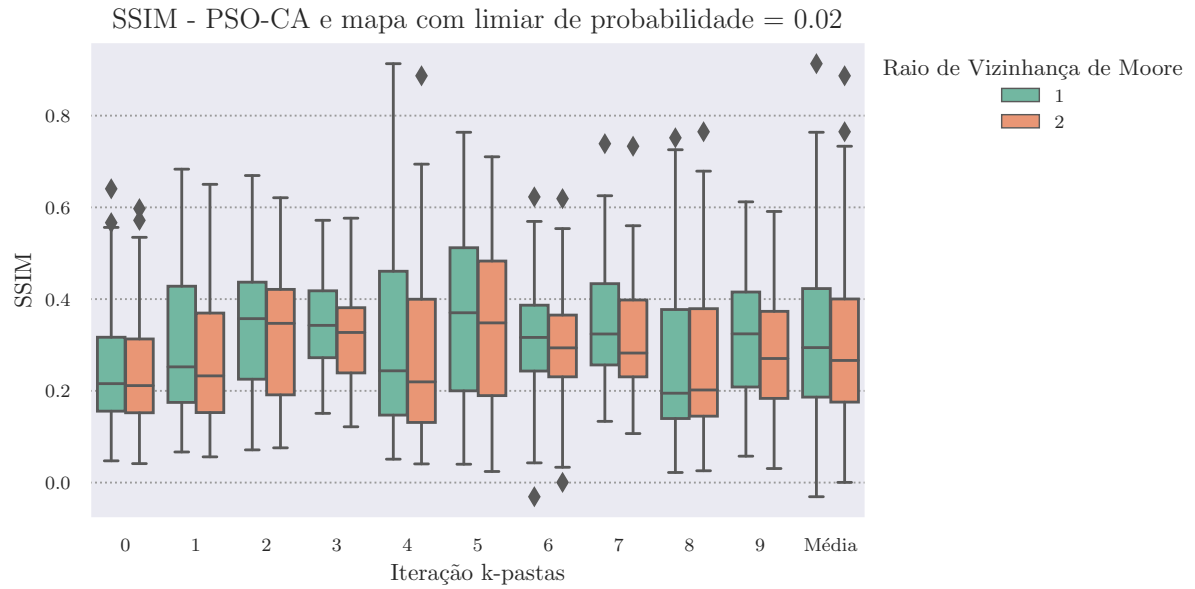
Ao se utilizar a base completa, o modelo geral apresentou resultados próximos aos modelos especializados e conseguiu contornar o problema encontrado nos modelos que utilizaram a categoria de paisagem no treinamento (tabela 9 e figuras 14 e 21b).

4.3 Comparação entre os modelos e o Canny

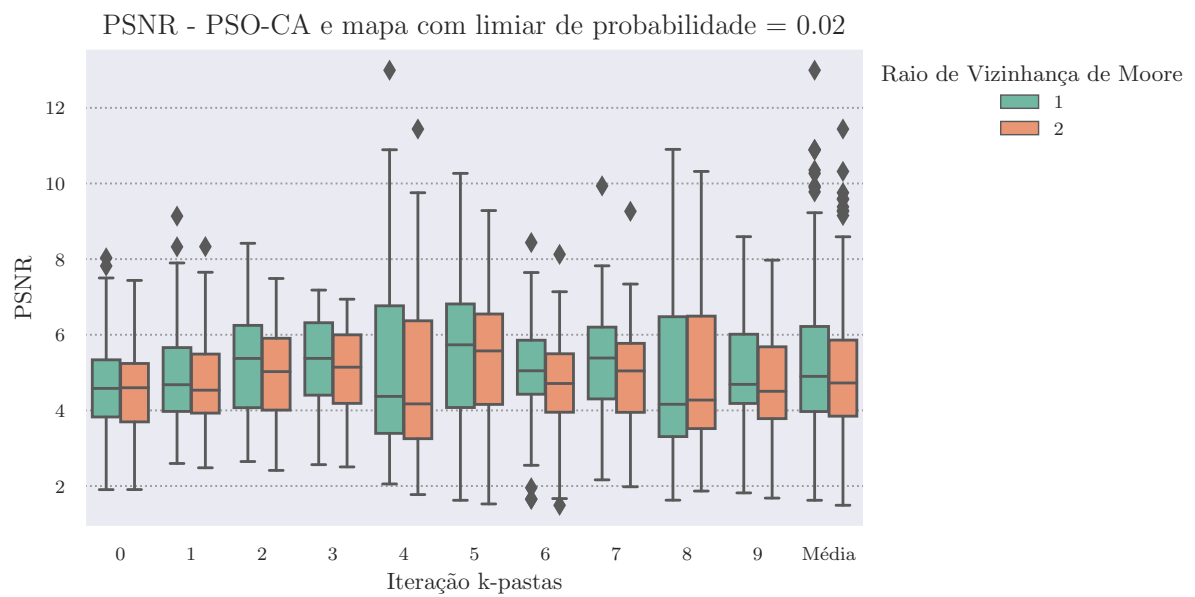
Para todos os experimentos propostos nesse estudo, o modelo estudado foi capaz de aderir às propriedades da imagem e a apresentar resultados melhores que o Canny. Para todos os casos, a métrica *SSIM* foi a que apresentou valores melhores para o modelo proposto e com maiores diferenças quando comparado ao Canny (figura 21b). A métrica *PSNR* também apresentou resultados melhores para o modelo proposto, mas sua diferença não foi tão significativa e para alguns casos apresentou resultados próximos aos do Canny (figura 21b).

Tabela 1: Resultados das métricas de avaliação para cada iteração do k-pastas para o modelo PSO-CA e mapa com limiar de probabilidade = 0.02.

Modelo	Iteração k-pastas	r	p	$PSNR$		$SSIM$	
PSO-CA	0	1	0,02	4,684	$\pm 1,486$	0,251	$\pm 0,139$
		2	0,02	4,583	$\pm 1,407$	0,248	$\pm 0,136$
	1	1	0,02	4,932	$\pm 1,459$	0,289	$\pm 0,155$
		2	0,02	4,715	$\pm 1,299$	0,265	$\pm 0,145$
	2	1	0,02	5,242	$\pm 1,384$	0,333	$\pm 0,146$
		2	0,02	4,909	$\pm 1,219$	0,307	$\pm 0,138$
	3	1	0,02	5,278	$\pm 1,181$	0,339	$\pm 0,111$
		2	0,02	5,065	$\pm 1,108$	0,317	$\pm 0,106$
	4	1	0,02	5,257	$\pm 2,498$	0,293	$\pm 0,199$
		2	0,02	4,957	$\pm 2,303$	0,280	$\pm 0,196$
	5	1	0,02	5,601	$\pm 2,018$	0,367	$\pm 0,194$
		2	0,02	5,414	$\pm 1,745$	0,344	$\pm 0,183$
	6	1	0,02	5,143	$\pm 1,515$	0,325	$\pm 0,137$
		2	0,02	4,772	$\pm 1,448$	0,296	$\pm 0,133$
	7	1	0,02	5,295	$\pm 1,418$	0,338	$\pm 0,129$
		2	0,02	4,889	$\pm 1,337$	0,308	$\pm 0,128$
	8	1	0,02	4,893	$\pm 2,151$	0,262	$\pm 0,173$
		2	0,02	4,903	$\pm 1,913$	0,263	$\pm 0,172$
	9	1	0,02	5,126	$\pm 1,481$	0,320	$\pm 0,137$
		2	0,02	4,764	$\pm 1,359$	0,288	$\pm 0,131$
Média	1	0,02	5,145	$\pm 1,707$	0,312	$\pm 0,157$	
	2	0,02	4,897	$\pm 1,554$	0,292	$\pm 0,150$	



(a) *SSIM*



(b) *PSNR*

Figura 10: Resultados da métrica *PSNR* e *SSIM* para cada iteração e média da validação cruzada em k-pastas com o limiar de probabilidade do mapa de bordas de 0.02.

Tabela 2: Resultados das métricas de avaliação de todas as iterações do k-pastas e sua média para o modelo PSO-CA e o modelo Canny com mapa com limiar de probabilidade = 0.02.

Modelo	Iteração k-pastas	p	$PSNR$		$SSIM$	
Canny	0	0,02	4,935	$\pm 0,531$	0,084	$\pm 0,038$
	1	0,02	4,874	$\pm 0,572$	0,082	$\pm 0,036$
	2	0,02	4,961	$\pm 0,455$	0,102	$\pm 0,045$
	3	0,02	4,899	$\pm 0,454$	0,115	$\pm 0,049$
	4	0,02	4,962	$\pm 0,585$	0,071	$\pm 0,038$
	5	0,02	5,162	$\pm 0,648$	0,092	$\pm 0,047$
	6	0,02	4,897	$\pm 0,524$	0,104	$\pm 0,046$
	7	0,02	4,989	$\pm 0,520$	0,103	$\pm 0,041$
	8	0,02	4,941	$\pm 0,677$	0,077	$\pm 0,040$
	9	0,02	4,857	$\pm 0,591$	0,101	$\pm 0,040$
	Média	0,02	4,948	$\pm 0,561$	0,093	$\pm 0,044$

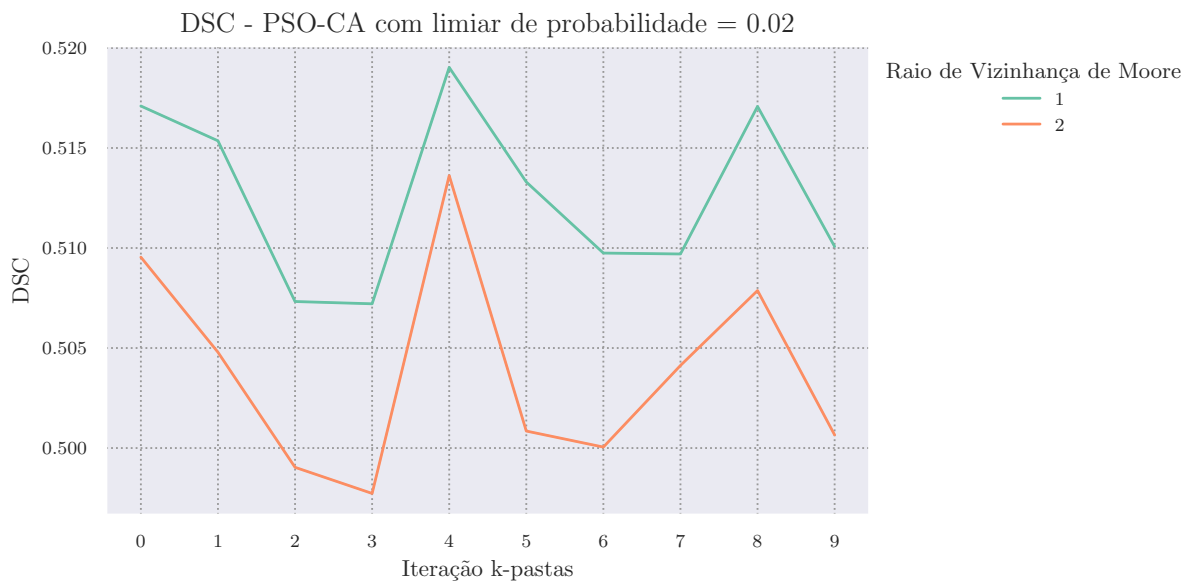
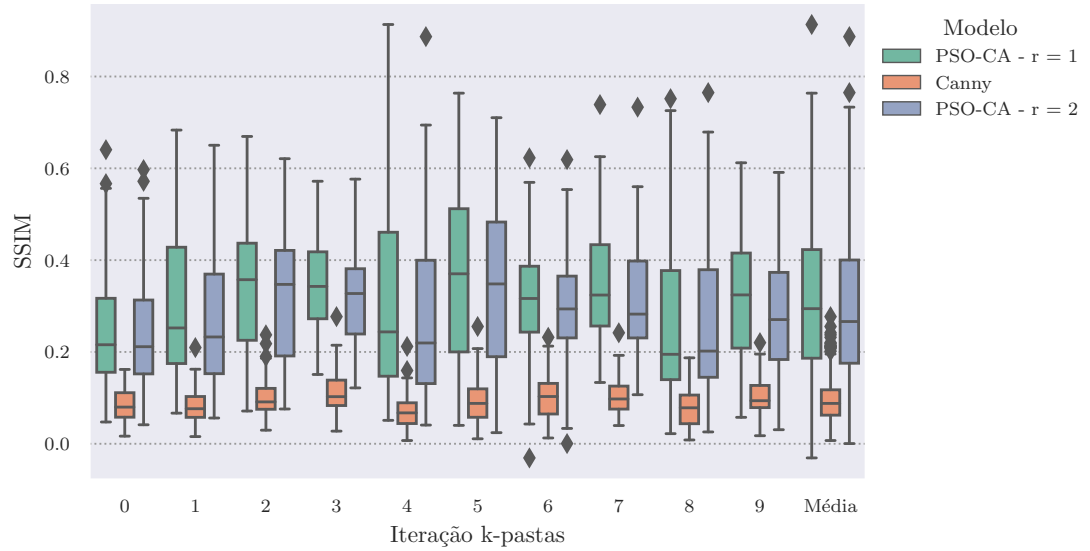


Figura 11: Valor da métrica DSC para a partícula encontrada durante a fase de otimização de cada iteração da validação cruzada em k-pastas com o limiar de probabilidade do mapa de bordas de 0.02.

Tabela 3: Resultados da função objetivo da etapa de otimização e os valores dos parâmetros encontrados de todas as iterações do k-pastas para o modelo PSO-CA e mapa com limiar de probabilidade = 0.02.

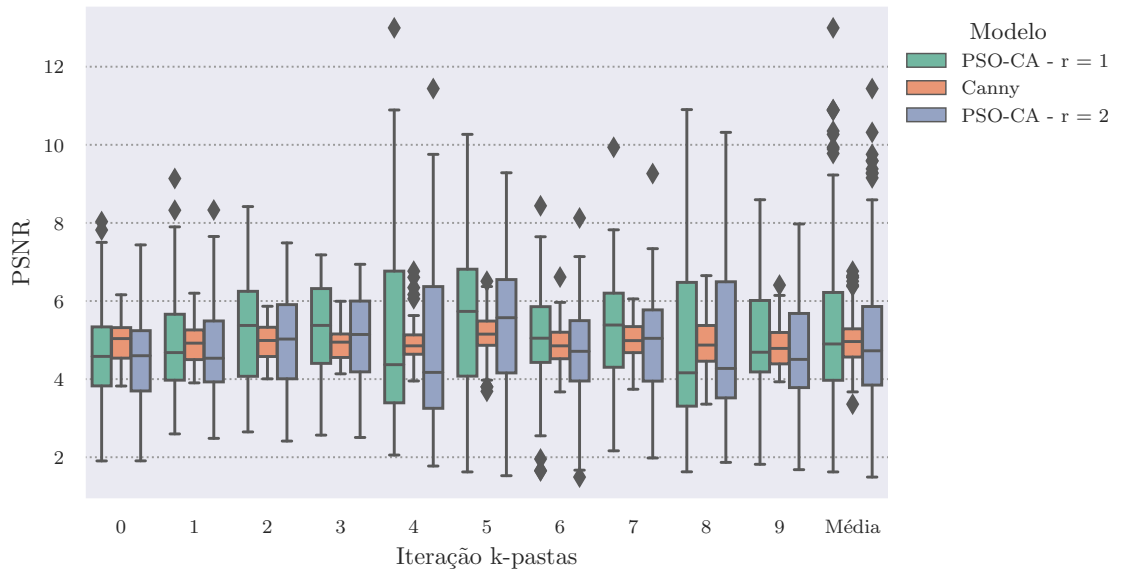
Modelo	r	p	Iteração k-pastas	Δ	τ	z	DSC
PSO-CA	1	0,02	0	20	0,744077	350	0,517
			1	127	0,338653	447	0,515
			2	126	0,267417	158	0,507
			3	125	0,325631	351	0,507
			4	146	0,280605	414	0,519
			5	120	0,353144	446	0,513
			6	121	0,330819	350	0,510
			7	234	0,206608	350	0,510
			8	231	0,214776	447	0,517
	9	253	0,206034	446	0,510		
	2	0,02	0	198	0,379419	566719	0,510
			1	65	0,751934	4390783	0,505
			2	86	0,590864	4246846	0,499
			3	129	0,535537	8435166	0,498
			4	80	0,543195	36317	0,514
			5	197	0,432928	12642622	0,501
			6	108	0,490182	535990	0,500
			7	51	0,677149	44508	0,504
8			158	0,483424	8447358	0,508	
9	217	0,398464	556538	0,501			

SSIM - Melhor PSO-CA comparado com o Canny e mapa com limiar de probabilidade = 0.02



(a) *SSIM*

PSNR - PSO-CA comparado com o Canny e mapa com limiar de probabilidade = 0.02



(b) *PSNR*

Figura 12: Comparação dos valores da métrica *SSIM* e *PSNR* entre melhor modelo da iteração do k-pastas e o detector Canny para cada iteração do k-pastas.

Tabela 4: Resultados das métricas de avaliação para os experimentos 2, 3, 4 e 5 para o modelo PSO-CA e mapa com limiar de probabilidade = 0.02.

Modelo	r	p	Base	Treinamento	$PSNR$		$SSIM$					
PSO-CA	1	0,02	Animais	Geral	4,903	$\pm 2,038$	0,270	$\pm 0,173$				
				TF-	5,075	$\pm 2,031$	0,279	$\pm 0,175$				
				Animais								
				TF-Objetos	4,990	$\pm 2,035$	0,274	$\pm 0,174$				
				TF-	4,514	$\pm 1,998$	0,241	$\pm 0,166$				
				Paisagens								
			Geral	TF-Pessoas	4,990	$\pm 2,035$	0,274	$\pm 0,174$				
				Geral	5,105	$\pm 1,723$	0,311	$\pm 0,157$				
				TF-	5,245	$\pm 1,706$	0,319	$\pm 0,157$				
				Animais								
				TF-Objetos	5,176	$\pm 1,715$	0,315	$\pm 0,157$				
				TF-	4,789	$\pm 1,720$	0,289	$\pm 0,156$				
			Objetos	Paisagens								
				TF-Pessoas	5,176	$\pm 1,715$	0,315	$\pm 0,157$				
				Geral	5,316	$\pm 1,600$	0,348	$\pm 0,151$				
				TF-	5,434	$\pm 1,571$	0,353	$\pm 0,149$				
				Animais								
				TF-Objetos	5,376	$\pm 1,587$	0,351	$\pm 0,150$				
			Paisagens	Paisagens								
				TF-	5,044	$\pm 1,609$	0,328	$\pm 0,152$				
				TF-Pessoas	5,376	$\pm 1,587$	0,351	$\pm 0,150$				
				Geral	5,350	$\pm 1,660$	0,348	$\pm 0,161$				
				TF-	5,463	$\pm 1,620$	0,353	$\pm 0,159$				
				Animais								
	Pessoas	TF-Objetos	5,407	$\pm 1,638$	0,350	$\pm 0,160$						
		TF-	5,106	$\pm 1,713$	0,333	$\pm 0,163$						
		Paisagens										
		TF-Pessoas	5,407	$\pm 1,638$	0,350	$\pm 0,160$						
		Geral	5,127	$\pm 1,293$	0,328	$\pm 0,121$						
		TF-	5,254	$\pm 1,291$	0,336	$\pm 0,121$						
	2	0,02		Animais	Geral	4,818	$\pm 1,830$	0,261	$\pm 0,168$			
					TF-	4,818	$\pm 1,830$	0,261	$\pm 0,168$			
					Animais							
					TF-Objetos	4,963	$\pm 1,841$	0,270	$\pm 0,170$			
					TF-	4,494	$\pm 1,842$	0,241	$\pm 0,164$			
					Paisagens							
				Geral	TF-Pessoas	4,942	$\pm 1,819$	0,267	$\pm 0,169$			
					Geral	4,968	$\pm 1,569$	0,297	$\pm 0,152$			
					TF-	4,968	$\pm 1,569$	0,297	$\pm 0,152$			
					Animais							
					TF-Objetos	5,101	$\pm 1,572$	0,306	$\pm 0,153$			
					TF-	4,710	$\pm 1,596$	0,281	$\pm 0,152$			
				Objetos	Paisagens							
					TF-Pessoas	5,081	$\pm 1,555$	0,303	$\pm 0,152$			
					Geral	5,130	$\pm 1,439$	0,328	$\pm 0,141$			
					TF-	5,130	$\pm 1,439$	0,328	$\pm 0,141$			
					Animais							
					TF-Objetos	5,260	$\pm 1,447$	0,338	$\pm 0,143$			
Paisagens				Paisagens								
				TF-	4,913	$\pm 1,480$	0,315	$\pm 0,144$				
				TF-Pessoas	5,232	$\pm 1,420$	0,334	$\pm 0,140$				
				Geral	5,216	$\pm 1,532$	0,334	$\pm 0,156$				
				TF-	5,216	$\pm 1,532$	0,334	$\pm 0,156$				
				Animais								
Pessoas	TF-Objetos	5,332	$\pm 1,518$	0,342	$\pm 0,156$							
	TF-	5,014	$\pm 1,600$	0,324	$\pm 0,160$							
	Paisagens											
	TF-Pessoas	5,313	$\pm 1,505$	0,339	$\pm 0,156$							
	Geral	4,941	$\pm 1,226$	0,307	$\pm 0,119$							
	TF-	4,941	$\pm 1,226$	0,307	$\pm 0,119$							
Animais												
TF-Objetos	5,069	$\pm 1,227$	0,317	$\pm 0,119$								
TF-	4,715	$\pm 1,219$	0,292	$\pm 0,118$								
Paisagens												
TF-Pessoas	5,052	$\pm 1,218$	0,314	$\pm 0,119$								

SSIM - Comparação das métricas entre os modelos PSO-CA TF treinados em cada categoria e mapa com limiar de probabilidade = 0.02

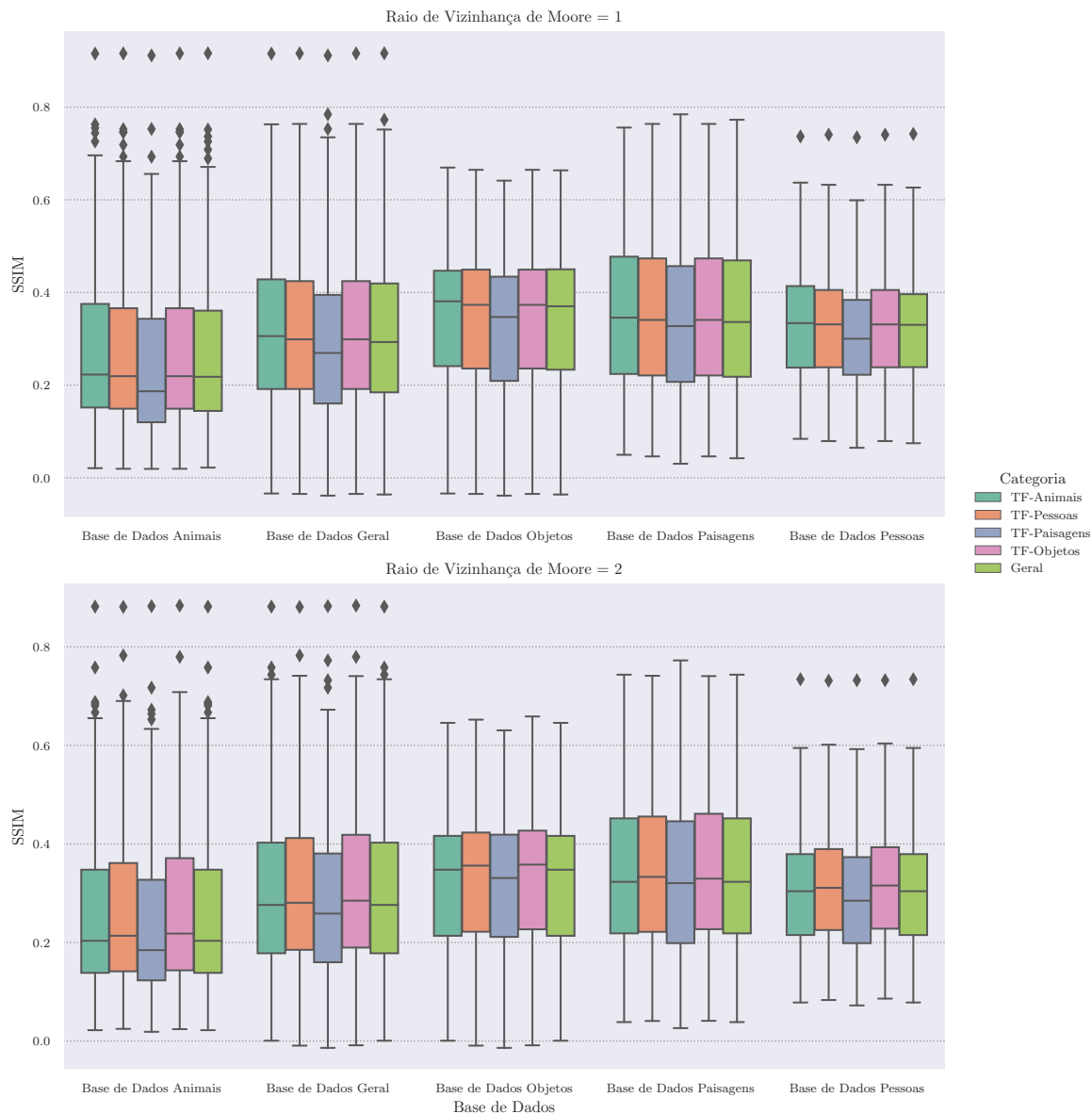


Figura 13: Comparação dos valores da métrica $SSIM$ para os casos de estudo: 2, 3, 4 e 5. Para cada valor de r e cada categoria da base de dados, os modelos gerais são comparados com modelos especializados.

PSNR - Comparação das métricas entre os modelos PSO-CA TF treinados em cada categoria e mapa com limiar de probabilidade = 0.02

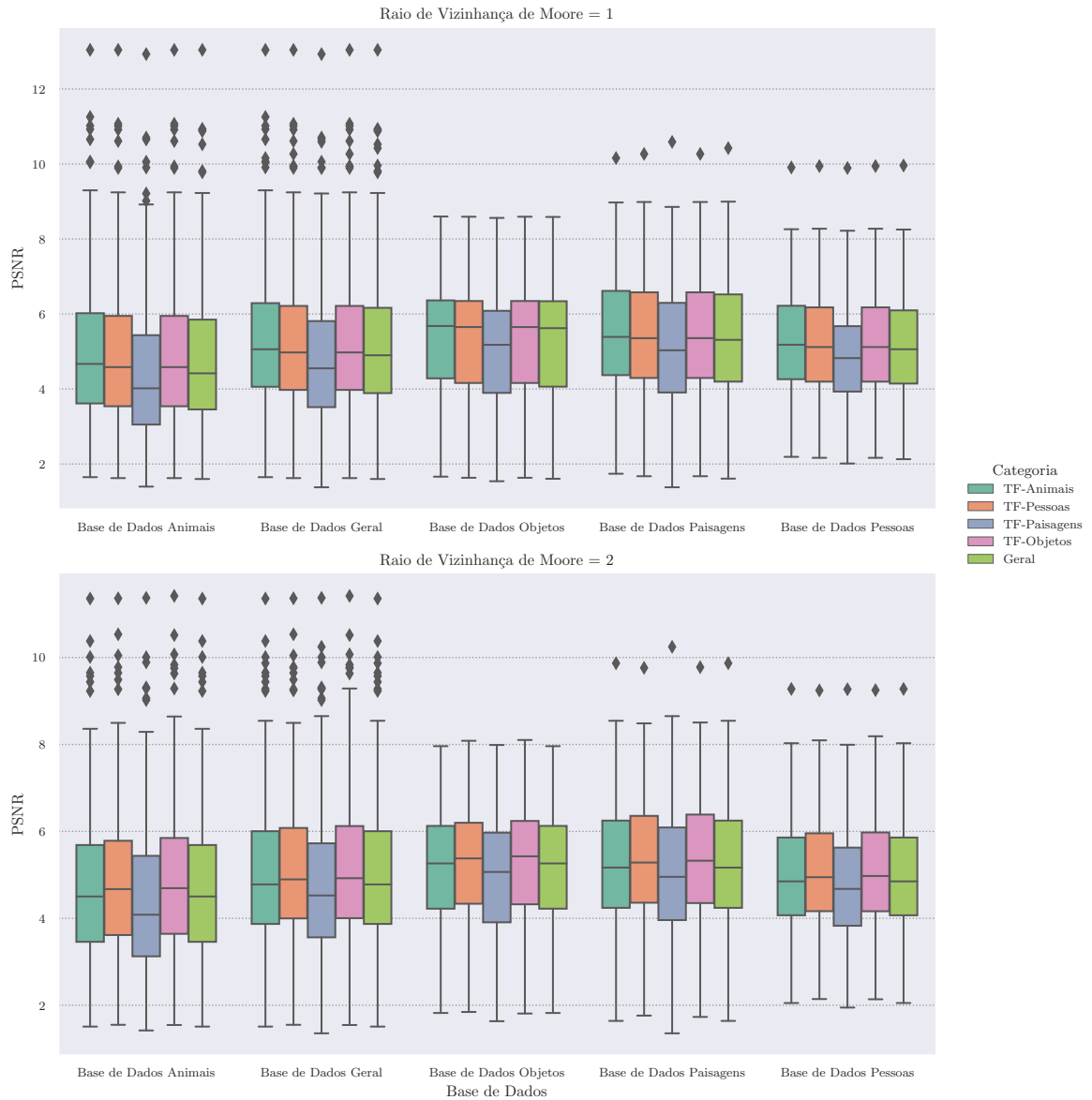


Figura 14: Comparação dos valores da métrica $PSNR$ para os casos de estudo: 2, 3, 4 e 5. Para cada valor de r e cada categoria da base de dados, os modelos gerais são comparados com modelos especializados.

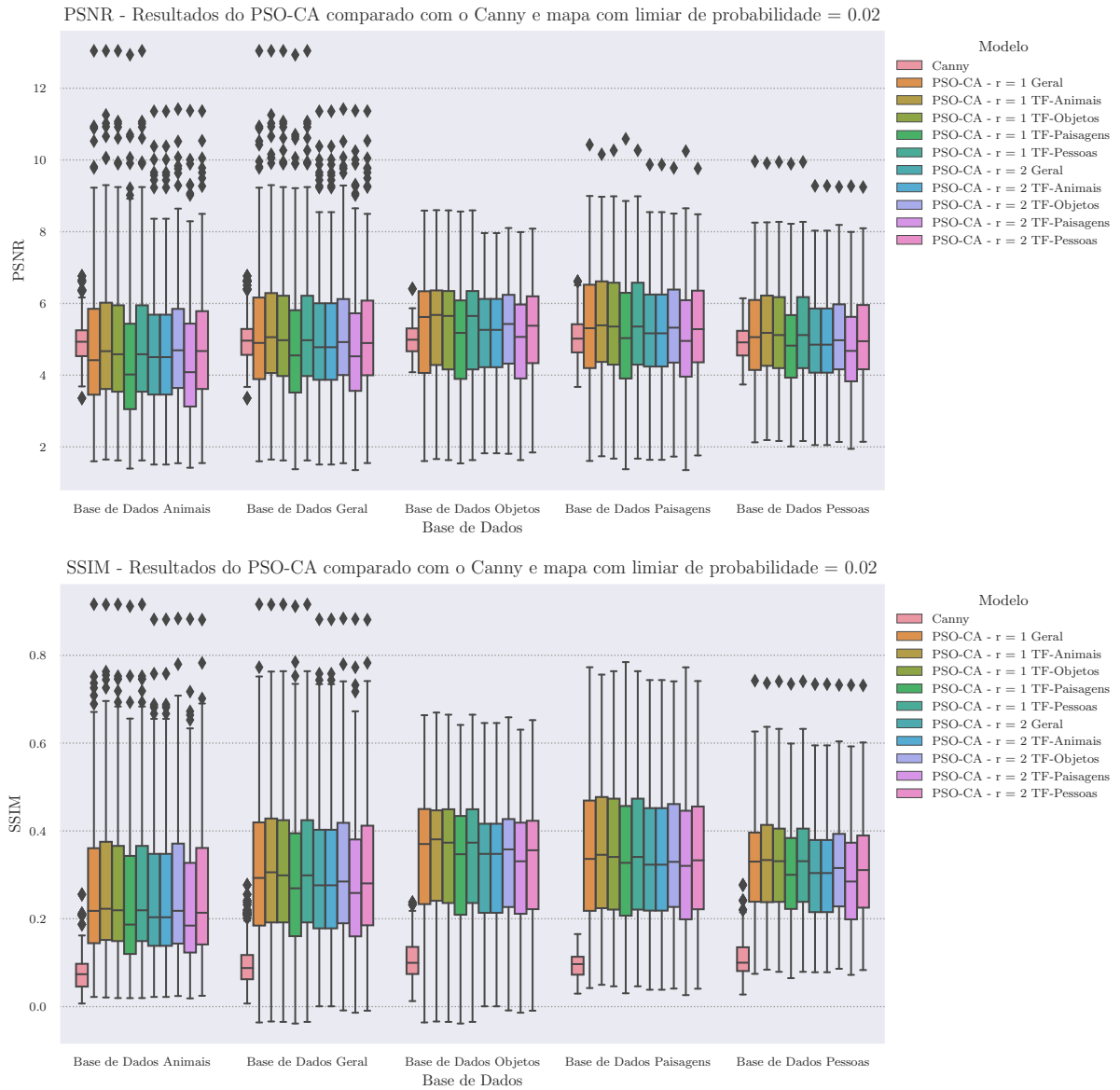


Figura 15: Comparação dos valores da métrica *SSIM* para os casos de estudo: 2, 3, 4 e 5. Para cada valor de r e cada categoria da base de dados, os modelos gerais são comparados com modelos especializados.

Tabela 5: Resultados das métricas de avaliação para a execução do modelo Canny para cada categoria da base de dados e mapa com limiar de probabilidade = 0.02

Modelo	Base	p	$PSNR$		$SSIM$	
Canny	Animais	0,02	4,922	$\pm 0,610$	0,077	$\pm 0,041$
	Geral	0,02	4,948	$\pm 0,561$	0,093	$\pm 0,044$
	Objetos	0,02	4,993	$\pm 0,478$	0,107	$\pm 0,050$
	Paisagens	0,02	5,035	$\pm 0,587$	0,094	$\pm 0,031$
	Pessoas	0,02	4,907	$\pm 0,494$	0,108	$\pm 0,044$

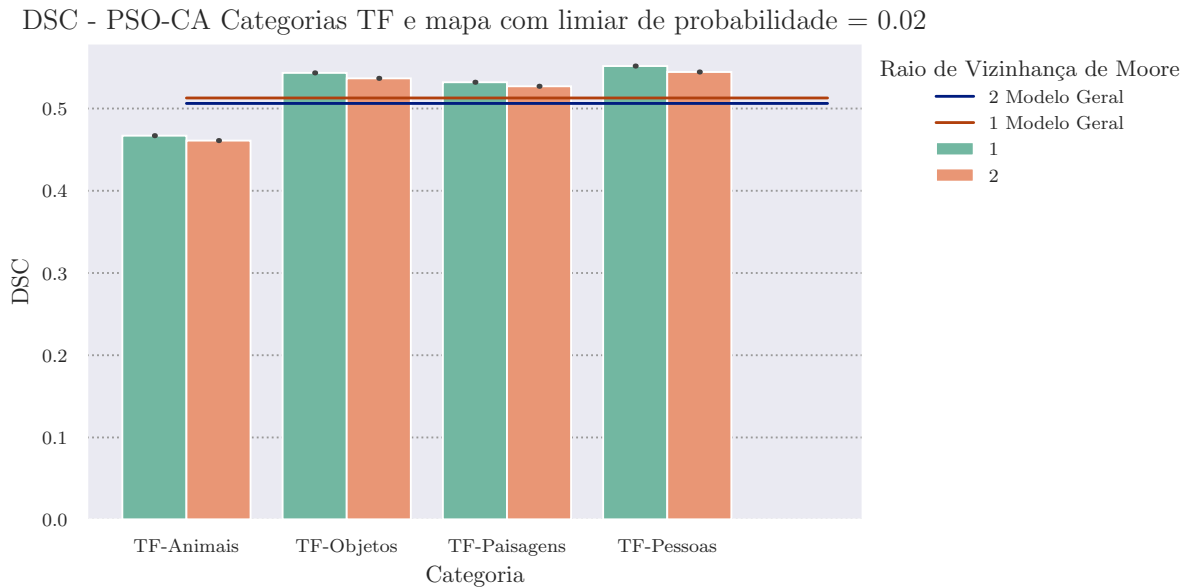


Figura 16: Comparação dos valores da métrica DSC do melhor indivíduo para os casos de estudo: 2 3, 4 e 5. No gráfico, a linha representa os valores da métrica para modelo geral, servindo como base da comparação.

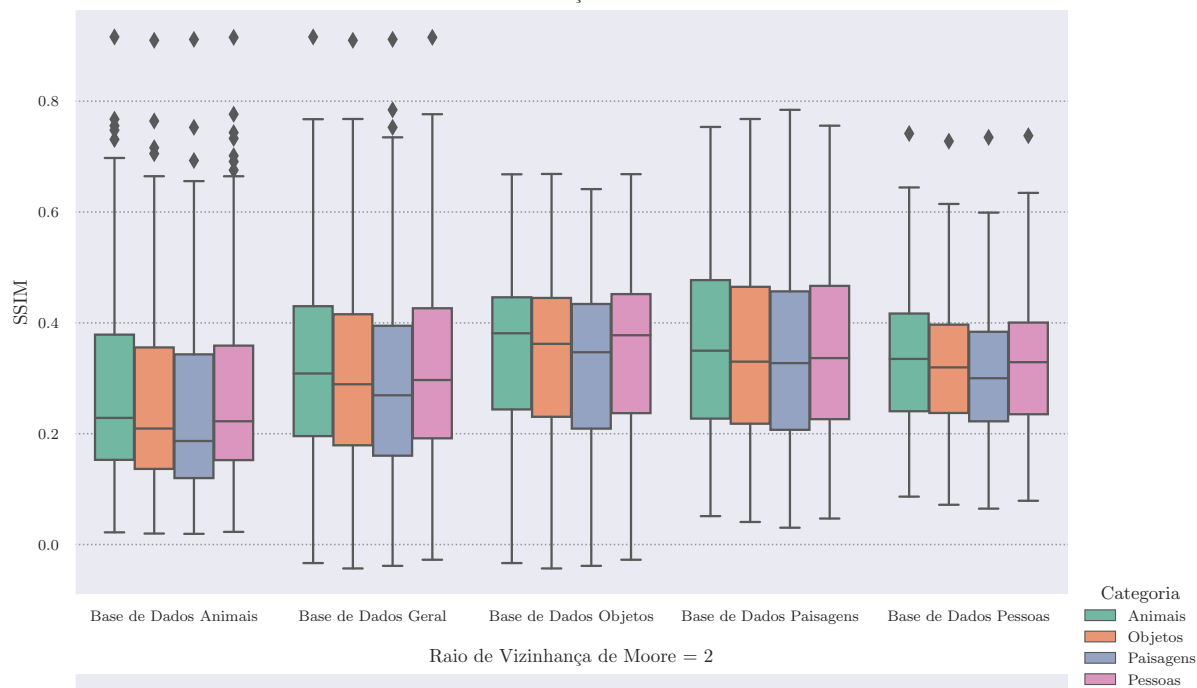
Tabela 6: Resultados da função objetivo da etapa de otimização e os valores dos parâmetros encontrados para os experimentos 2, 3, 4 e 5 para o modelo PSO-CA e mapa com limiar de probabilidade = 0.02.

Modelo	r	p	Treinamento	Δ	τ	z	DSC	
PSO-CA	1	0,02	Geral	102	0,382719	447	0,513	
			TF-	89	0,431699	446	0,467	
			Animais					
			TF-	82	0,445122	447	0,543	
			Objetos					
			TF-	74	0,398369	415	0,532	
			Paisagens					
	TF-	91	0,419048	446	0,552			
	Pessoas							
	2	0,02	Geral	146	0,490741	61307	0,506	
			TF-	146	0,490741	61307	0,461	
			Animais					
			TF-	146	0,491368	44990	0,537	
			Objetos					
TF-			108	0,488158	44508	0,527		
Paisagens								
TF-	147	0,498451	61367	0,544				
Pessoas								

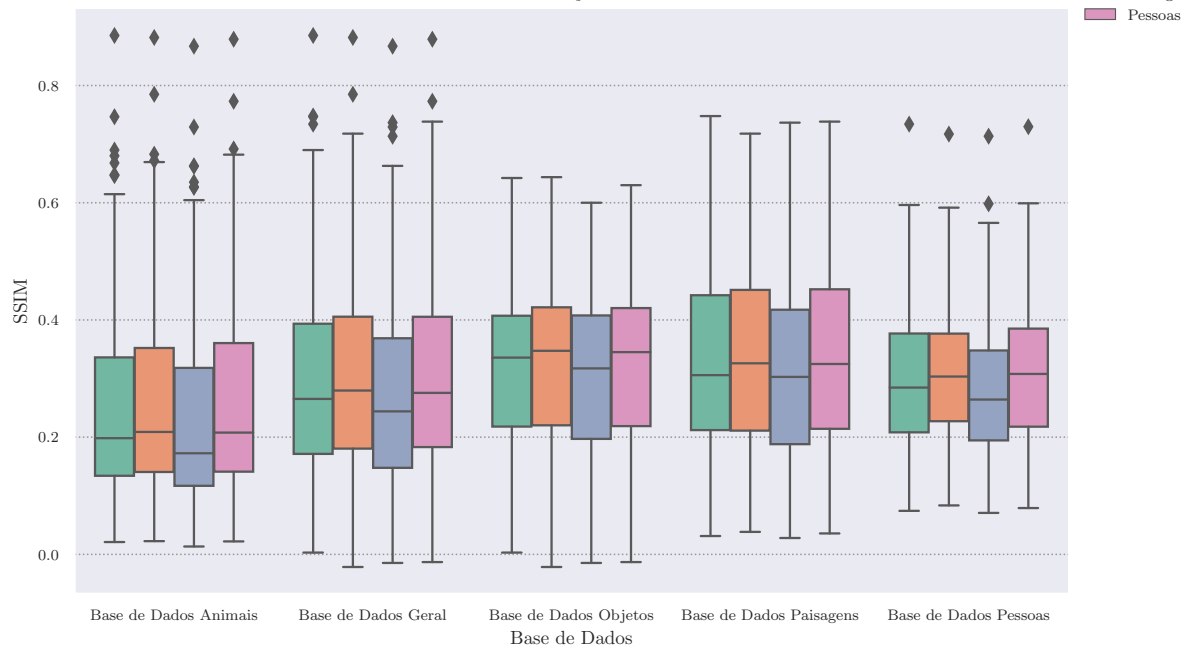
Tabela 7: Resultados das métricas de avaliação para o experimento 6 com o modelo PSO-CA e mapa com limiar de probabilidade = 0.02

Modelo	r	p	Base	Treinamento	$PSNR$		$SSIM$	
PSO-CA	1	0,02	Animais	Animais	5,116	$\pm 2,028$	0,282	$\pm 0,175$
				Objetos	4,865	$\pm 1,996$	0,260	$\pm 0,170$
				Paisagens	4,514	$\pm 1,998$	0,241	$\pm 0,166$
				Pessoas	5,050	$\pm 2,007$	0,274	$\pm 0,173$
			Geral	Animais	5,278	$\pm 1,702$	0,321	$\pm 0,157$
				Objetos	5,094	$\pm 1,697$	0,305	$\pm 0,157$
				Paisagens	4,789	$\pm 1,720$	0,289	$\pm 0,156$
				Pessoas	5,217	$\pm 1,688$	0,315	$\pm 0,156$
			Objetos	Animais	5,463	$\pm 1,564$	0,355	$\pm 0,148$
				Objetos	5,330	$\pm 1,586$	0,345	$\pm 0,150$
				Paisagens	5,044	$\pm 1,609$	0,328	$\pm 0,152$
				Pessoas	5,406	$\pm 1,554$	0,350	$\pm 0,149$
			Paisagens	Animais	5,489	$\pm 1,610$	0,354	$\pm 0,158$
				Objetos	5,356	$\pm 1,641$	0,345	$\pm 0,160$
				Paisagens	5,106	$\pm 1,713$	0,333	$\pm 0,163$
				Pessoas	5,443	$\pm 1,603$	0,349	$\pm 0,158$
	Pessoas	Animais	5,283	$\pm 1,292$	0,338	$\pm 0,122$		
		Objetos	5,130	$\pm 1,275$	0,324	$\pm 0,122$		
		Paisagens	4,844	$\pm 1,276$	0,307	$\pm 0,121$		
		Pessoas	5,215	$\pm 1,279$	0,331	$\pm 0,121$		
	2	0,02	Animais	Animais	4,685	$\pm 1,818$	0,254	$\pm 0,168$
				Objetos	4,916	$\pm 1,789$	0,262	$\pm 0,167$
				Paisagens	4,374	$\pm 1,775$	0,232	$\pm 0,163$
				Pessoas	4,843	$\pm 1,802$	0,264	$\pm 0,168$
			Geral	Animais	4,838	$\pm 1,560$	0,288	$\pm 0,151$
				Objetos	5,049	$\pm 1,530$	0,297	$\pm 0,150$
				Paisagens	4,566	$\pm 1,523$	0,267	$\pm 0,148$
				Pessoas	4,975	$\pm 1,541$	0,297	$\pm 0,151$
			Objetos	Animais	4,994	$\pm 1,419$	0,318	$\pm 0,139$
				Objetos	5,214	$\pm 1,399$	0,329	$\pm 0,140$
				Paisagens	4,740	$\pm 1,384$	0,299	$\pm 0,138$
				Pessoas	5,105	$\pm 1,387$	0,325	$\pm 0,138$
Paisagens			Animais	5,101	$\pm 1,529$	0,327	$\pm 0,156$	
			Objetos	5,280	$\pm 1,482$	0,333	$\pm 0,154$	
			Paisagens	4,828	$\pm 1,500$	0,305	$\pm 0,154$	
			Pessoas	5,198	$\pm 1,501$	0,331	$\pm 0,155$	
Pessoas			Animais	4,807	$\pm 1,217$	0,296	$\pm 0,118$	
			Objetos	5,010	$\pm 1,199$	0,307	$\pm 0,116$	
			Paisagens	4,577	$\pm 1,175$	0,276	$\pm 0,114$	
			Pessoas	4,951	$\pm 1,213$	0,307	$\pm 0,118$	

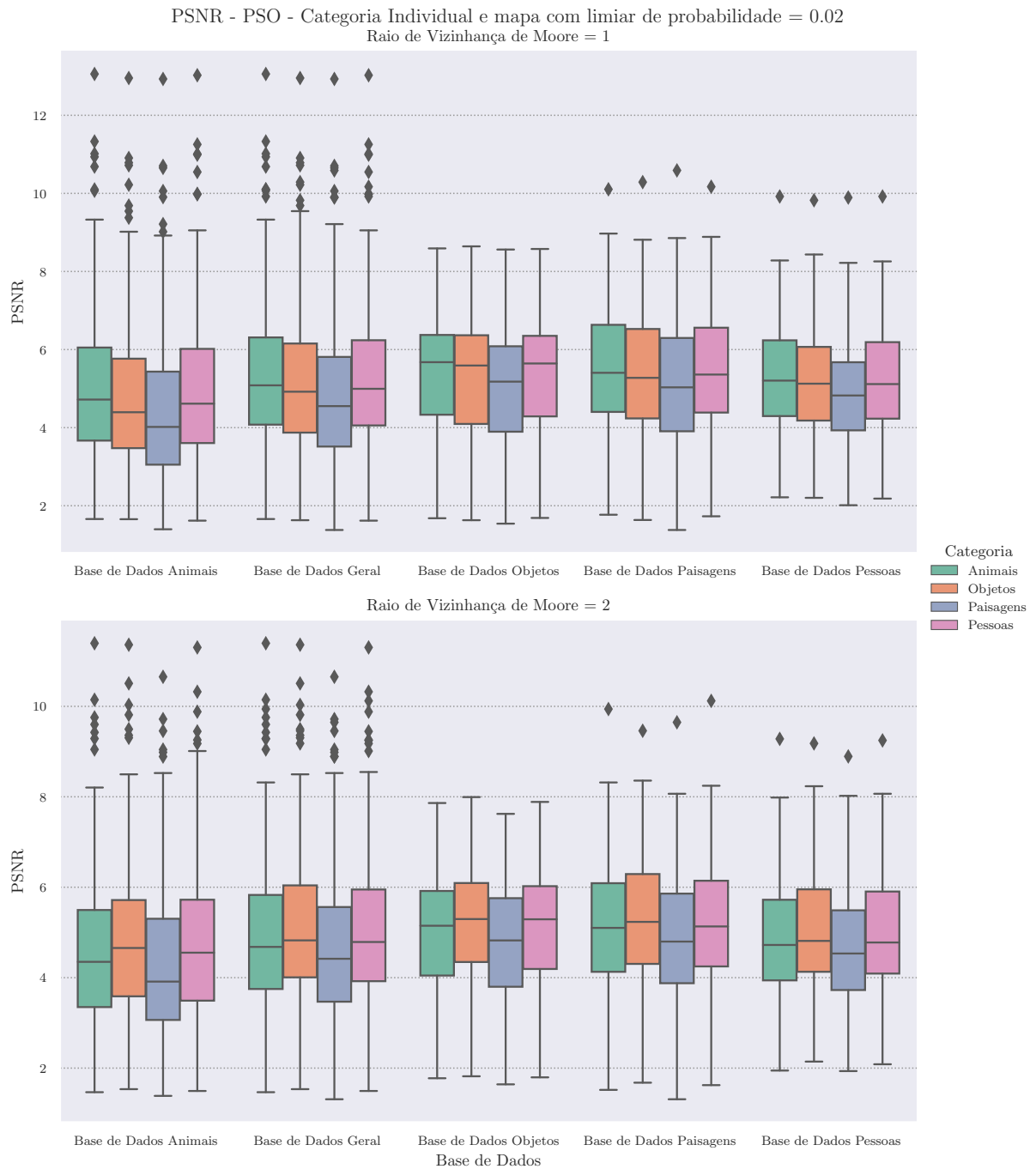
SSIM - PSO-CA Categoria Individual e mapa com limiar de probabilidade = 0.02
Raio de Vizinhança de Moore = 1



Raio de Vizinhança de Moore = 2



(a) SSIM



(a) *PSNR*

Figura 18: Comparação dos valores da métrica *PSNR* para os casos de estudo: 6. Para cada valor de r e cada categoria da base de dados, os modelos treinados exclusivamente nas categorias são comparados com os modelos treinados em outras categoria.

Tabela 8: Resultados da função objetivo da etapa de otimização e os valores dos parâmetros encontrados para o experimento 6 com o modelo PSO-CA e mapa com limiar de probabilidade = 0.02.

Modelo	r	p	Treinamento	Δ	τ	z	DSC
PSO-CA	1	0,02	Animais	75	0,477621	446	0,467
			Objetos	172	0,205238	159	0,543
			Paisagens	50	0,494020	414	0,532
			Pessoas	141	0,305168	351	0,551
	2	0,02	Animais	142	0,519296	585717	0,459
			Objetos	162	0,408242	45775	0,534
			Paisagens	136	0,387359	4199259	0,517
			Pessoas	173	0,497688	8450039	0,542

DSC - PSO-CA Categorias Individuais e mapa com limiar de probabilidade = 0.02

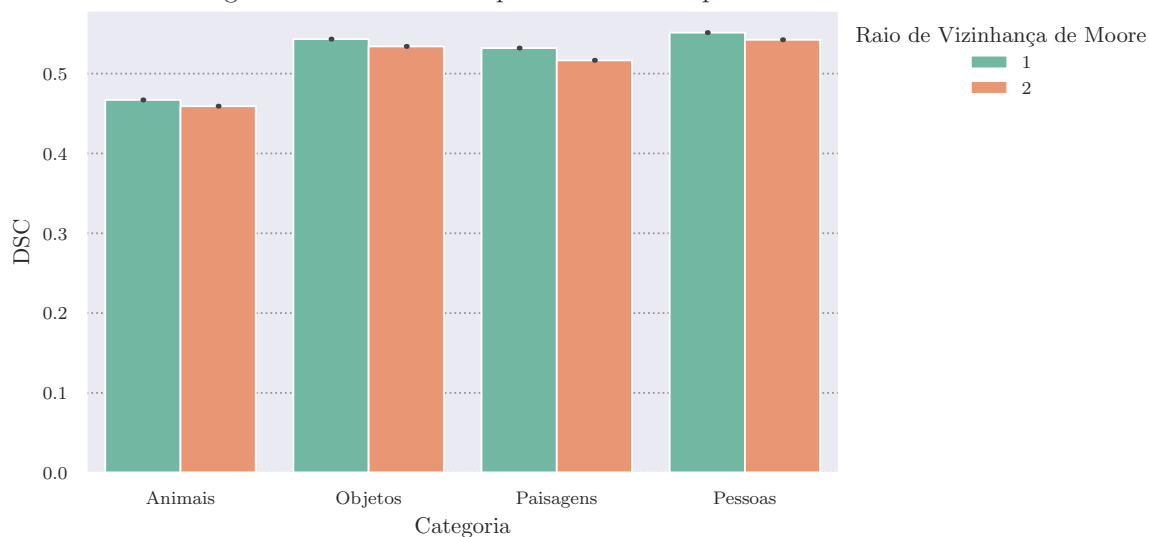
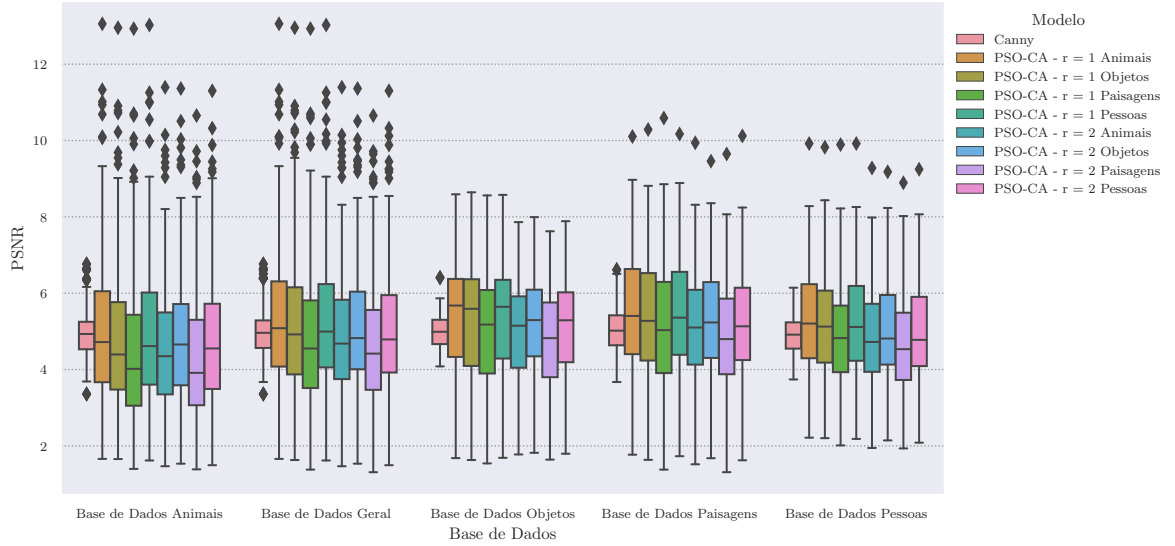


Figura 19: Comparação dos valores da métrica DSC do melhor indivíduo para cada modelo treinado para o caso de estudo 6.

PSNR - Resultados do PSO-CA Categoria Individual comparado com o Canny e mapa com limiar de probabilidade = 0.02



SSIM - Resultados do PSO-CA Categoria Individual comparado com o Canny e mapa com limiar de probabilidade = 0.02

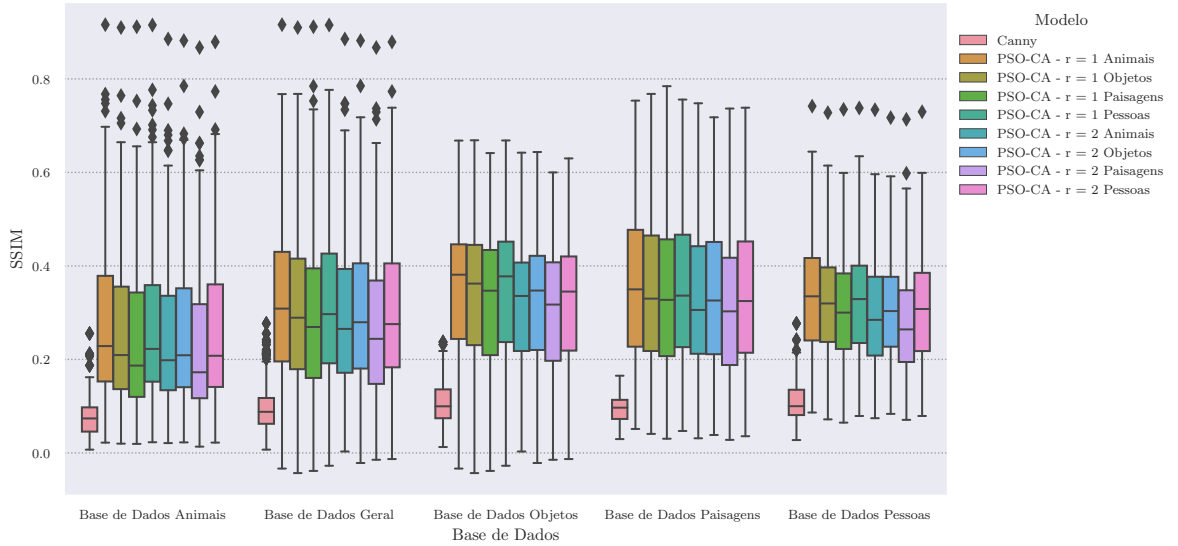


Figura 20: Comparação dos modelos encontrados no caso de estudo 6 com o Canny.

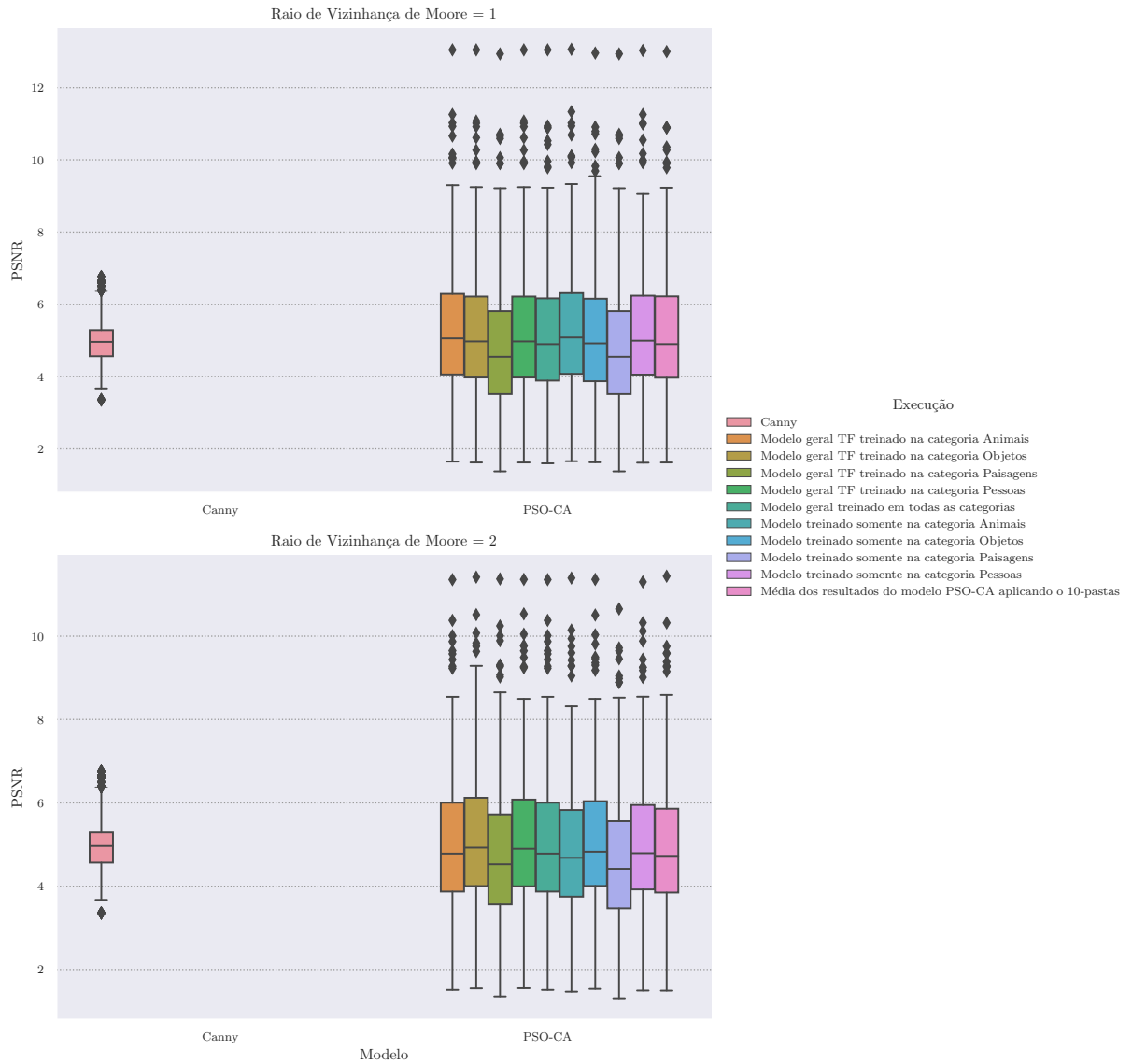
Tabela 9: Comparação entre os resultados médios das métricas de avaliação para cada experimento realizado utilizando o modelo PSO-CA e mapa com limiar de probabilidade = 0.02

Modelo	r	p	Execução	$PSNR$		$SSIM$		
PSO-CA	1	0,02	Modelo	5,245	$\pm 1,706$	0,319	$\pm 0,157$	
			TF-Animais					
			Modelo	5,176	$\pm 1,715$	0,315	$\pm 0,157$	
			TF-Objetos					
			Modelo	4,789	$\pm 1,720$	0,289	$\pm 0,156$	
			TF-Paisagens					
			Modelo	5,176	$\pm 1,715$	0,315	$\pm 0,157$	
			TF-Pessoas					
			Modelo geral treinado na base inteira	5,105	$\pm 1,723$	0,311	$\pm 0,157$	
	Modelo treinado somente em Animais	5,278	$\pm 1,702$	0,321	$\pm 0,157$			
	Modelo treinado somente em Objetos	5,094	$\pm 1,697$	0,305	$\pm 0,157$			
	Modelo treinado somente em Paisagens	4,789	$\pm 1,720$	0,289	$\pm 0,156$			
	Modelo treinado somente em Pessoas	5,217	$\pm 1,688$	0,315	$\pm 0,156$			
	Média dos resultados do modelo com 10-pastas	5,145	$\pm 1,707$	0,312	$\pm 0,157$			
		2	0,02	Modelo	4,968	$\pm 1,569$	0,297	$\pm 0,152$
				TF-Animais				
				Modelo	5,101	$\pm 1,572$	0,306	$\pm 0,153$
				TF-Objetos				
Modelo				4,710	$\pm 1,596$	0,281	$\pm 0,152$	
TF-Paisagens								
Modelo				5,081	$\pm 1,555$	0,303	$\pm 0,152$	
TF-Pessoas								
Modelo geral treinado na base inteira				4,968	$\pm 1,569$	0,297	$\pm 0,152$	
Modelo treinado somente em Animais		4,838	$\pm 1,560$	0,288	$\pm 0,151$			
Modelo treinado somente em Objetos		5,049	$\pm 1,530$	0,297	$\pm 0,150$			
Modelo treinado somente em Paisagens		4,566	$\pm 1,523$	0,267	$\pm 0,148$			
Modelo treinado somente em Pessoas		4,975	$\pm 1,541$	0,297	$\pm 0,151$			
Média dos resultados do modelo com 10-pastas		4,897	$\pm 1,554$	0,292	$\pm 0,150$			

Tabela 10: Comparação entre os resultados finais da métrica utilizada durante a fase de otimização para cada experimento realizado utilizando o modelo PSO-CA e mapa com limiar de probabilidade = 0.02

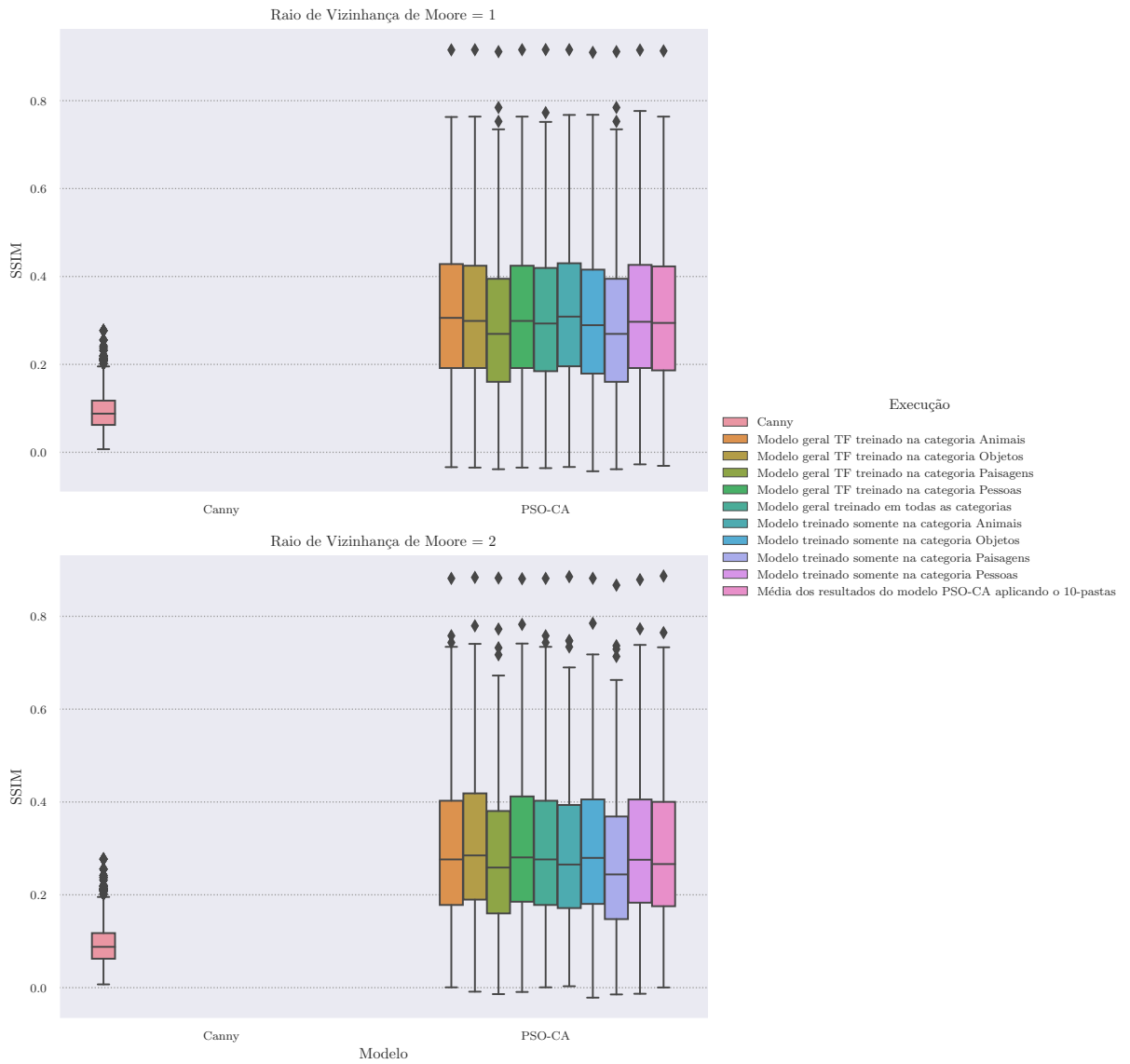
Modelo	r	p	Execução	Δ	τ	z	DSC
PSO-CA	1	0,02	Modelo geral TF treinado na categoria Animais	89	0,431699	446	0,467 \pm 0,000
			Modelo geral TF treinado na categoria Objetos	82	0,445122	447	0,543 \pm 0,000
			Modelo geral TF treinado na categoria Paisagens	74	0,398369	415	0,532 \pm 0,000
			Modelo geral TF treinado na categoria Pessoas	91	0,419048	446	0,552 \pm 0,000
			Modelo geral treinado em todas as categorias	102	0,382719	447	0,513 \pm 0,000
			Modelo treinado somente na categoria Animais	75	0,477621	446	0,467 \pm 0,000
			Modelo treinado somente na categoria Objetos	172	0,205238	159	0,543 \pm 0,000
			Modelo treinado somente na categoria Paisagens	50	0,494020	414	0,532 \pm 0,000
			Modelo treinado somente na categoria Pessoas	141	0,305168	351	0,551 \pm 0,000
			Média dos resultados do modelo PSO-CA aplicando o 10-pastas	-	-	-	-
2	0,02	0,02	Modelo geral TF treinado na categoria Animais	146	0,490741	61307	0,461 0,000
			Modelo geral TF treinado na categoria Objetos	146	0,491368	44990	0,537 \pm 0,000
			Modelo geral TF treinado na categoria Paisagens	108	0,488158	44508	0,527 \pm 0,000
			Modelo geral TF treinado na categoria Pessoas	147	0,498451	61367	0,544 \pm 0,000
			Modelo geral treinado em todas as categorias	146	0,490741	61307	0,506 \pm 0,000
			Modelo treinado somente na categoria Animais	142	0,519296	585717	0,459 \pm 0,000
			Modelo treinado somente na categoria Objetos	162	0,408242	45775	0,534 \pm 0,000
			Modelo treinado somente na categoria Paisagens	136	0,387359	4199259	0,517 \pm 0,000
			Modelo treinado somente na categoria Pessoas	173	0,497688	8450039	0,542 \pm 0,000
			Média dos resultados do modelo PSO-CA aplicando o 10-pastas	-	-	-	-

PSNR - PSO-CA Comparando modelos com Canny e mapa com limiar de probabilidade = 0.02



(a) PSNR

SSIM - PSO-CA Comparando modelos com Canny e mapa com limiar de probabilidade = 0.02



(b) *SSIM*

Figura 21: Comparação dos valores da métricas *SSIM* e *PSNR* entre todos os modelos dos casos de estudo.

Tabela 11: Resultados das métricas de avaliação e função objetivo de cada categoria da base dados para os experimentos 2, 3, 4, 5 e 6 para o modelo PSO-CA e mapa com limiar de probabilidade = 0.02

Modelo	r	p	Base	Treinamento	PSNR	SSIM					
PSO-CA	1	0,02	Animais	Animais	5,116	$\pm 2,028$	0,282	$\pm 0,175$			
				Geral	4,903	$\pm 2,038$	0,270	$\pm 0,173$			
				Objetos	4,865	$\pm 1,996$	0,260	$\pm 0,170$			
				Paisagens	4,514	$\pm 1,998$	0,241	$\pm 0,166$			
				Pessoas	5,050	$\pm 2,007$	0,274	$\pm 0,173$			
				TF-Animais	5,075	$\pm 2,031$	0,279	$\pm 0,175$			
				TF-Objetos	4,990	$\pm 2,035$	0,274	$\pm 0,174$			
				TF-Paisagens	4,514	$\pm 1,998$	0,241	$\pm 0,166$			
				TF-Pessoas	4,990	$\pm 2,035$	0,274	$\pm 0,174$			
				Geral	Animais	5,278	$\pm 1,702$	0,321	$\pm 0,157$		
					Geral	5,105	$\pm 1,723$	0,311	$\pm 0,157$		
					Objetos	5,094	$\pm 1,697$	0,305	$\pm 0,157$		
					Paisagens	4,789	$\pm 1,720$	0,289	$\pm 0,156$		
					Pessoas	5,217	$\pm 1,688$	0,315	$\pm 0,156$		
					TF-Animais	5,245	$\pm 1,706$	0,319	$\pm 0,157$		
					TF-Objetos	5,176	$\pm 1,715$	0,315	$\pm 0,157$		
					TF-Paisagens	4,789	$\pm 1,720$	0,289	$\pm 0,156$		
					TF-Pessoas	5,176	$\pm 1,715$	0,315	$\pm 0,157$		
			Objetos		Animais	5,463	$\pm 1,564$	0,355	$\pm 0,148$		
					Geral	5,316	$\pm 1,600$	0,348	$\pm 0,151$		
					Objetos	5,330	$\pm 1,586$	0,345	$\pm 0,150$		
					Paisagens	5,044	$\pm 1,609$	0,328	$\pm 0,152$		
					Pessoas	5,406	$\pm 1,554$	0,350	$\pm 0,149$		
					TF-Animais	5,434	$\pm 1,571$	0,353	$\pm 0,149$		
					TF-Objetos	5,376	$\pm 1,587$	0,351	$\pm 0,150$		
					TF-Paisagens	5,044	$\pm 1,609$	0,328	$\pm 0,152$		
					TF-Pessoas	5,376	$\pm 1,587$	0,351	$\pm 0,150$		
				Paisagens	Animais	5,489	$\pm 1,610$	0,354	$\pm 0,158$		
					Geral	5,350	$\pm 1,660$	0,348	$\pm 0,161$		
					Objetos	5,356	$\pm 1,641$	0,345	$\pm 0,160$		
					Paisagens	5,106	$\pm 1,713$	0,333	$\pm 0,163$		
					Pessoas	5,443	$\pm 1,603$	0,349	$\pm 0,158$		
					TF-Animais	5,463	$\pm 1,620$	0,353	$\pm 0,159$		
					TF-Objetos	5,407	$\pm 1,638$	0,350	$\pm 0,160$		
					TF-Paisagens	5,106	$\pm 1,713$	0,333	$\pm 0,163$		
					TF-Pessoas	5,407	$\pm 1,638$	0,350	$\pm 0,160$		
			Pessoas		Animais	5,283	$\pm 1,292$	0,338	$\pm 0,122$		
					Geral	5,127	$\pm 1,293$	0,328	$\pm 0,121$		
					Objetos	5,130	$\pm 1,275$	0,324	$\pm 0,122$		
					Paisagens	4,844	$\pm 1,276$	0,307	$\pm 0,121$		
					Pessoas	5,215	$\pm 1,279$	0,331	$\pm 0,121$		
					TF-Animais	5,254	$\pm 1,291$	0,336	$\pm 0,121$		
					TF-Objetos	5,193	$\pm 1,294$	0,333	$\pm 0,121$		
					TF-Paisagens	4,844	$\pm 1,276$	0,307	$\pm 0,121$		
					TF-Pessoas	5,193	$\pm 1,294$	0,333	$\pm 0,121$		
				2	0,02	Animais	Animais	4,685	$\pm 1,818$	0,254	$\pm 0,168$
							Geral	4,818	$\pm 1,830$	0,261	$\pm 0,168$
							Objetos	4,916	$\pm 1,789$	0,262	$\pm 0,167$
							Paisagens	4,374	$\pm 1,775$	0,232	$\pm 0,163$
							Pessoas	4,843	$\pm 1,802$	0,264	$\pm 0,168$
							TF-Animais	4,818	$\pm 1,830$	0,261	$\pm 0,168$
							TF-Objetos	4,963	$\pm 1,841$	0,270	$\pm 0,170$
							TF-Paisagens	4,494	$\pm 1,842$	0,241	$\pm 0,164$
							TF-Pessoas	4,942	$\pm 1,819$	0,267	$\pm 0,169$
Geral	Animais	4,838	$\pm 1,560$				0,288	$\pm 0,151$			
	Geral	4,968	$\pm 1,569$				0,297	$\pm 0,152$			
	Objetos	5,049	$\pm 1,530$				0,297	$\pm 0,150$			
	Paisagens	4,566	$\pm 1,523$				0,267	$\pm 0,148$			
	Pessoas	4,975	$\pm 1,541$				0,297	$\pm 0,151$			
	TF-Animais	4,968	$\pm 1,569$				0,297	$\pm 0,152$			
	TF-Objetos	5,101	$\pm 1,572$				0,306	$\pm 0,153$			
	TF-Paisagens	4,710	$\pm 1,596$				0,281	$\pm 0,152$			
	TF-Pessoas	5,081	$\pm 1,555$				0,303	$\pm 0,152$			
	Objetos	Animais	4,994			$\pm 1,419$	0,318	$\pm 0,139$			
		Geral	5,130			$\pm 1,439$	0,328	$\pm 0,141$			
		Objetos	5,214			$\pm 1,399$	0,329	$\pm 0,140$			
		Paisagens	4,740			$\pm 1,384$	0,299	$\pm 0,138$			
		Pessoas	5,105			$\pm 1,387$	0,325	$\pm 0,138$			
		TF-Animais	5,130			$\pm 1,439$	0,328	$\pm 0,141$			
		TF-Objetos	5,260			$\pm 1,447$	0,338	$\pm 0,143$			
		TF-Paisagens	4,913			$\pm 1,480$	0,315	$\pm 0,144$			
		TF-Pessoas	5,232			$\pm 1,420$	0,334	$\pm 0,140$			
Paisagens		Animais	5,101			$\pm 1,529$	0,327	$\pm 0,156$			
		Geral	5,216			$\pm 1,532$	0,334	$\pm 0,156$			
		Objetos	5,280			$\pm 1,482$	0,333	$\pm 0,154$			
		Paisagens	4,828			$\pm 1,500$	0,305	$\pm 0,154$			
		Pessoas	5,198			$\pm 1,501$	0,331	$\pm 0,155$			
		TF-Animais	5,216			$\pm 1,532$	0,334	$\pm 0,156$			
		TF-Objetos	5,332			$\pm 1,518$	0,342	$\pm 0,156$			
		TF-Paisagens	5,014			$\pm 1,600$	0,324	$\pm 0,160$			
		TF-Pessoas	5,313			$\pm 1,505$	0,339	$\pm 0,156$			
	Pessoas	Animais	4,807			$\pm 1,217$	0,296	$\pm 0,118$			
		Geral	4,941			$\pm 1,226$	0,307	$\pm 0,119$			
		Objetos	5,010			$\pm 1,199$	0,307	$\pm 0,116$			
		Paisagens	4,577			$\pm 1,175$	0,276	$\pm 0,114$			
		Pessoas	4,951			$\pm 1,213$	0,307	$\pm 0,118$			
		TF-Animais	4,941			$\pm 1,226$	0,307	$\pm 0,119$			
		TF-Objetos	5,069			$\pm 1,227$	0,317	$\pm 0,119$			
		TF-Paisagens	4,715			$\pm 1,219$	0,292	$\pm 0,118$			
		TF-Pessoas	5,052			$\pm 1,218$	0,314	$\pm 0,119$			

SSIM - Comparação entre os experimentos de PSO-CA para cada categoria 0.02

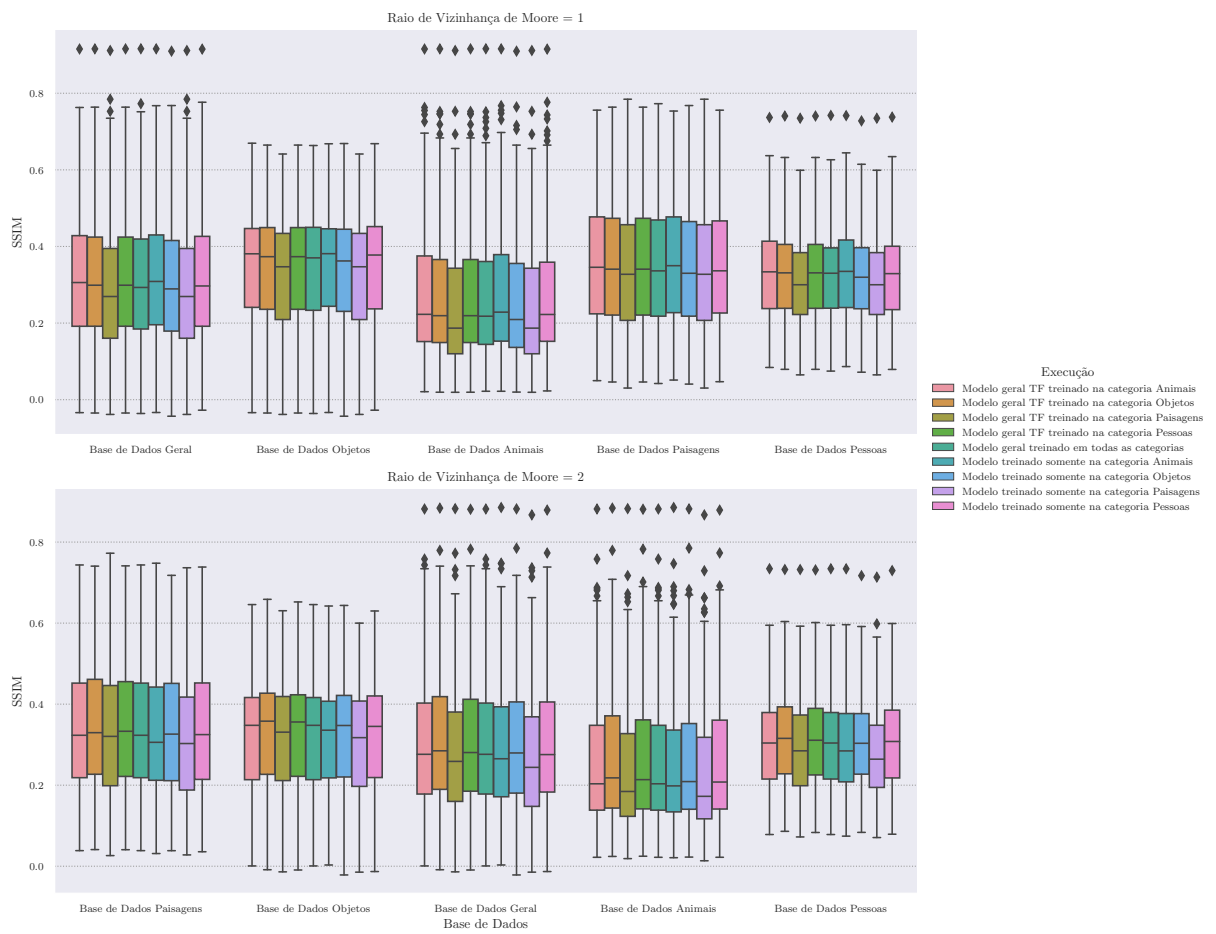


Figura 22: Comparação dos valores da métrica *SSIM* entre os itens de estudo 2, 3 e 6.

PSNR - Comparação entre os experimentos de PSO-CA para cada categoria 0.02

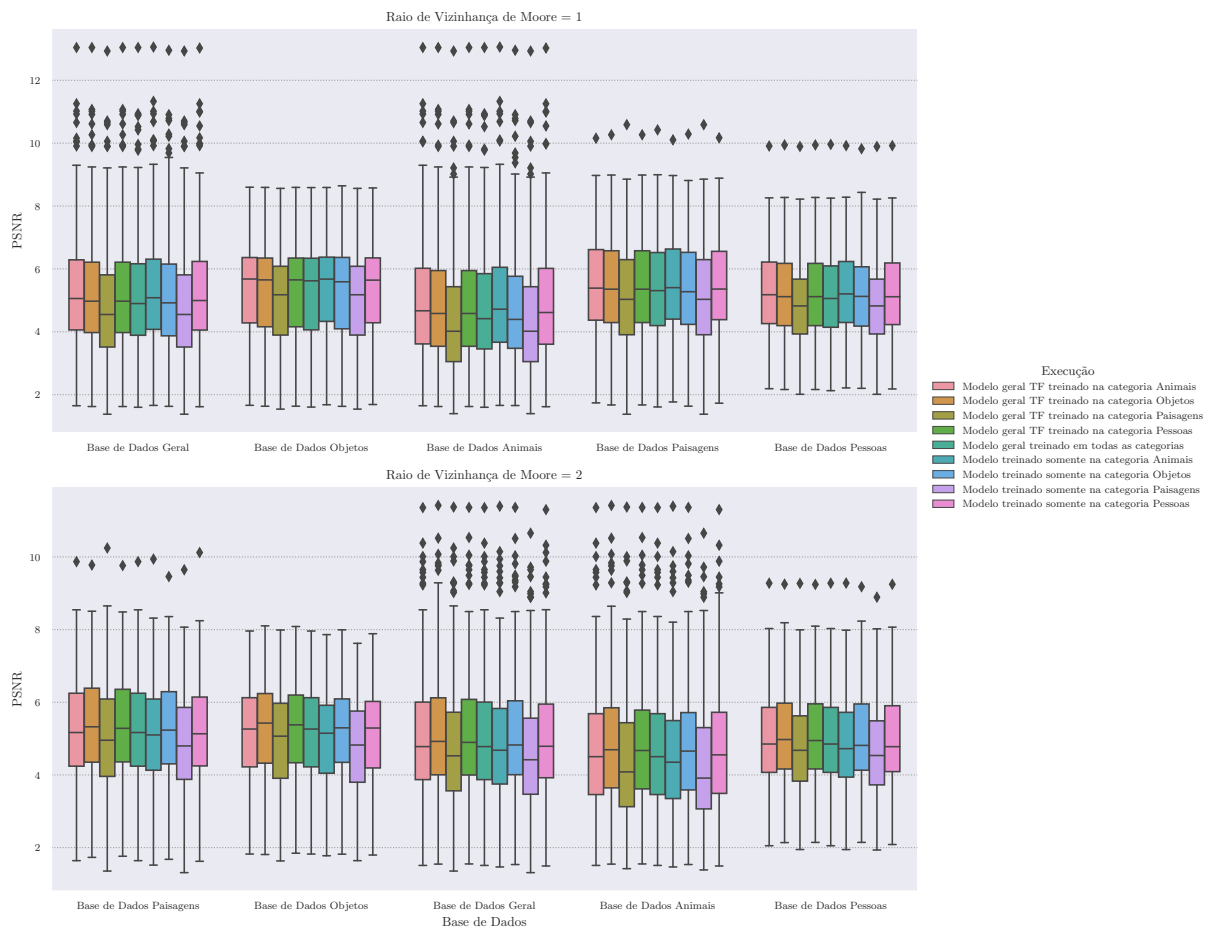


Figura 23: Comparação dos valores da métrica $PSNR$ entre os itens de estudo 2, 3 e 6.

5 CONCLUSÕES

A partir dos resultados, é possível concluir que para o cenário com $p = 0.02$, ampliar o espaço de busca das regras do autômato não traz benefícios no resultado médio uma vez que o modelo com $r = 1$ apresentou resultados melhores ou bem próximos em ambas as métricas para grande parte das execuções, indicando que o espaço de busca do modelo do trabalho de referência já foi o suficiente para estes experimentos.

A aplicação da técnica de *Transfer Learning* não apresentou a adaptabilidade esperada. Os resultados dos experimentos apresentam que o *Transfer Learning* não teve um impacto significativo para os cenários de especialização, encontrando regras com resultados próximos ao modelo geral e aos modelos individuais (21a e 21b). A fase de otimização proposta não foi o suficiente para a fase de treinamento quando combinado com *Transfer Learning* para explorar os espaços de solução dos autômatos, encontrando novas regras com pouca variação do valor de da função objetivo *DSC* (tabelas 9, 11, 10, 8 e 6).

O modelo de paisagem apresentou constantemente os piores resultados para todas as variações de experimentos. Esse resultado pode ser justificado pela falta de características intrínsecas à mesma e a variação dos cenários de paisagens que diferentemente de humanos, animais e objetos, podem não possuir muitos pontos em comum como feições de rostos e anatomias parecidas.

Os resultados do modelo geral demonstram que o mesmo foi capaz de se adaptar à base como um todo, apresentando resultados comparativos aos modelos especializados e individuais para grande parte das subcategorias de imagens e resultados melhores para a base de paisagens quando comparado com as versões especializada e individual (figuras 13 e 14).

A validação cruzada com 10 pastas também foi capaz de reforçar a adaptabilidade do modelos, pois, apesar das diferenças nas técnicas de treinamento e separação dos conjuntos, os resultados médios finais foram bem próximos para todos os experimentos e os valores da métricas encontradas nas execuções podem ser descritos por um intervalo em comum para o modelo (figura 21).

Quando comparados com o detector Canny, todas as variações do modelo proposto foram capazes de apresentar melhores resultados nas métricas de avaliação, confirmando

a adaptabilidade do modelo às características de uma base.

Ao se analisar o modelo com $r = 2$, é possível que a granularidade do domínio do *PSO* proposto, valores no intervalo de 0 e 1, não sejam o suficiente para representar a expansão do espaço de busca das regras de forma precisa. Qualquer valor pequeno do domínio do intervalo quando multiplicado pelo número máximo da regra de $r = 2$, apresentará um número regra com pouca granularidade, tornando-se difícil encontrar uma regra de números pequenos ou de explorar um espaço de maneira precisa.

Outro ponto de atenção seria na fase de otimização, ao aplicar técnicas de *Transfer Learning*, o modelo não apresentou uma aderência melhor às características de cada categoria. Este ponto pode ser justificado pelos valores de *DSC* e regras encontradas na fase de otimização. Valores próximos indicam que a segunda etapa de otimização para os modelos especializados estão ficando com partículas presas e não estão conseguindo explorar soluções com valores globais. Uma proposição seria uma variação dos parâmetros para a segunda etapa de otimização para instigar a exploração de novas soluções e de uma alteração do processamento em lote para ambas as etapas de otimização ou somente para segunda etapa. A alteração do processamento em lote ajudaria a diminuir a diluição do valor do *DSC* que para a pesquisa é representado pela média de todas as imagens passadas ao modelo.

Para futuros estudos, além das considerações já mencionadas, também seria interessante a aplicação do modelo proposto para bases com categorias já predefinidas ou um processo de pré-processamento e divisão de categorias mais robustos como a utilização de métricas de contraste ou de outros classificadores para ajudar na imparcialidade e em uma divisão mais robusta utilizando técnicas de aumento de dados para equilibrar a proporção das subcategorias.

Referências

- ADDAE, B.; DRAGIĆEVIĆ, S. Integrating multi-criteria analysis and spherical cellular automata approach for modelling global urban land-use change. *Geocarto International*, Taylor & Francis, p. 2152498, 2022.
- AL-AMRI, S. S.; KALYANKAR, N.; KHAMITKAR, S. Image segmentation by using edge detection. *International journal on computer science and engineering*, Citeseer, v. 2, n. 3, p. 804–807, 2010.
- ARBELAEZ, P.; MAIRE, M.; FOWLKES, C.; MALIK, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 33, n. 5, p. 898–916, maio 2011. ISSN 0162-8828.
- ARORA, M.; WIESMANN, L.; CHEN, X.; STACHNISS, C. Mapping the static parts of dynamic scenes from 3d lidar point clouds exploiting ground segmentation. In: *2021 European Conference on Mobile Robots (ECMR)*. [S.l.: s.n.], 2021. p. 1–6.
- BASIRI, A.; LINES, T.; PEREIRA, M. F. Scalable 3d mapping of cities using computer vision and signals of opportunity. *International Journal of Geographical Information Science*, Taylor & Francis, v. 37, n. 7, p. 1470–1495, 2023.
- BERRAR, D. Cross-validation. In: RANGANATHAN, S.; GRIBSKOV, M.; NAKAI, K.; SCHÖNBACH, C. (Ed.). *Encyclopedia of Bioinformatics and Computational Biology*. Oxford: Academic Press, 2019. p. 542–545. ISBN 978-0-12-811432-2. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B978012809633820349X>>.
- BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. *Information Sciences*, Elsevier, v. 237, p. 82–117, 2013.
- BROADHURST, D. I.; KELL, D. B. Statistical strategies for avoiding false discoveries in metabolomics and related experiments. *Metabolomics*, Springer, v. 2, p. 171–196, 2006.
- BROWNE, M. W. Cross-validation methods. *Journal of Mathematical Psychology*, v. 44, n. 1, p. 108–132, 2000. ISSN 0022-2496. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0022249699912798>>.

- CANNY, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, n. 6, p. 679–698, 1986.
- CHO, H.; SWARTZLANDER, E. E. Adder designs and analyses for quantum-dot cellular automata. *IEEE Transactions on Nanotechnology*, v. 6, n. 3, p. 374–383, 2007.
- DICE, L. R. Measures of the amount of ecologic association between species. *Ecology*, JSTOR, v. 26, n. 3, p. 297–302, 1945.
- DUMITRU, D.; ANDREICA, A.; DIOȘAN, L.; BÁLINT, Z. Evolutionary curriculum learning approach for transferable cellular automata rule optimization. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*. [S.l.: s.n.], 2020. p. 63–64.
- DUMITRU, D.; ANDREICA, A.; DIOȘAN, L.; BÁLINT, Z. Robustness analysis of transferable cellular automata rules optimized for edge detection. *Procedia Computer Science*, Elsevier, v. 176, p. 713–722, 2020.
- DUMITRU, D.; DIOȘAN, L.; ANDREICA, A.; BÁLINT, Z. A transfer learning approach on the optimization of edge detectors for medical images using particle swarm optimization. *Entropy*, MDPI, v. 23, n. 4, p. 414, 2021.
- HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COURNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. Fernández del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. Array programming with NumPy. *Nature*, v. 585, p. 357–362, 2020.
- HE, J.; ZHANG, S.; YANG, M.; SHAN, Y.; HUANG, T. Bi-directional cascade network for perceptual edge detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 3828–3837.
- HORÉ, A.; ZIOU, D. Is there a relationship between peak-signal-to-noise ratio and structural similarity index measure? *IET Image Processing*, Wiley Online Library, v. 7, n. 1, p. 12–24, 2013.

JAIN, R.; KASTURI, R.; SCHUNCK, B. *Machine Vision*. [S.l.: s.n.], 1995. ISBN 978-0-07-032018-5.

JI, Y.; WANG, W.; ZHENG, M.; CHEN, S. Real time building evacuation modeling with an improved cellular automata method and corresponding iot system implementation. *Buildings*, v. 12, n. 6, 2022. ISSN 2075-5309. Disponível em: <<https://www.mdpi.com/2075-5309/12/6/718>>.

KARI, J. Theory of cellular automata: A survey. *Theoretical Computer Science*, v. 334, n. 1, p. 3–33, 2005. ISSN 0304-3975. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S030439750500054X>>.

Karl Kosaen. *Karl Kosaen: Learning - Log*. 2016. <http://karlrosaen.com/ml/learning-log/2016-06-20/>.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *Proceedings of ICNN'95-international conference on neural networks*. [S.l.], 1995. v. 4, p. 1942–1948.

KHAN, A. R.; CHOUDHURY, P. P.; DIHIDAR, K.; MITRA, S.; SARKAR, P. Vlsi architecture of a cellular automata machine. *Computers & Mathematics with Applications*, Elsevier, v. 33, n. 5, p. 79–94, 1997.

LIU, Y.; CHENG, M.-M.; HU, X.; WANG, K.; BAI, X. Richer convolutional features for edge detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 3000–3009.

MIRANDA, L. J. V. PySwarms, a research-toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software*, v. 3, 2018. Disponível em: <<https://doi.org/10.21105/joss.00433>>.

MOHAMMED, J.; NAYAK, D. R. An efficient edge detection technique by two dimensional rectangular cellular automata. In: IEEE. *International Conference on Information Communication and Embedded Systems (ICICES2014)*. [S.l.], 2014. p. 1–4.

PADMAVATHI, G.; SUBASHINI, P.; LAVANYA, P. Performance evaluation of the various edge detectors and filters for the noisy ir images. In: *2nd International Conference on Sensors, Signals, Visualization, Imaging, Simulation and Material, Baltimore, USA*. [S.l.: s.n.], 2009. p. 199–203.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; BUC, F. d'Alché; FOX, E.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems 32*. [S.l.]: Curran Associates, Inc., 2019. p. 8024–8035.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, É. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PERES, F.; CASTELLI, M. Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development. *Applied Sciences*, v. 11, n. 14, 2021. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/11/14/6449>>.

POOBATHY, D.; CHEZIAN, R. M. Edge detection operators: Peak signal to noise ratio based comparison. *IJ Image, Graphics and Signal Processing*, MECS, v. 10, p. 55–61, 2014.

PU, M.; HUANG, Y.; LIU, Y.; GUAN, Q.; LING, H. Edter: Edge detection with transformer. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2022. p. 1402–1412.

RANI, S.; GHAI, D.; KUMAR, S. Object detection and recognition using contour based edge detection and fast r-cnn. *Multimedia Tools and Applications*, Springer, v. 81, n. 29, p. 42183–42207, 2022.

SAMMUT, C.; WEBB, G. I. *Encyclopedia of machine learning*. [S.l.]: Springer Science & Business Media, 2011.

SANTÉ, I.; GARCÍA, A. M.; MIRANDA, D.; CRECENTE, R. Cellular automata models for the simulation of real-world urban processes: A review and analysis. *Landscape and urban planning*, Elsevier, v. 96, n. 2, p. 108–122, 2010.

SAYAMA, H. *Introduction to the modeling and analysis of complex systems*. [S.l.]: Open SUNY Textbooks, 2015.

TORRES, F. S.; WILLE, R.; NIEMANN, P.; DRECHSLER, R. An energy-aware model for the logic synthesis of quantum-dot cellular automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 37, n. 12, p. 3031–3041, 2018.

UGUZ, S.; SAHIN, U.; SAHIN, F. Edge detection with fuzzy cellular automata transition function optimized by pso. *Computers & Electrical Engineering*, Elsevier, v. 43, p. 180–192, 2015.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J.; van der Walt, S. J.; BRETT, M.; WILSON, J.; MILLMAN, K. J.; MAYOROV, N.; NELSON, A. R. J.; JONES, E.; KERN, R.; LARSON, E.; CAREY, C. J.; POLAT, İ.; FENG, Y.; MOORE, E. W.; VanderPlas, J.; LAXALDE, D.; PERKTOLD, J.; CIMRMAN, R.; HENRIKSEN, I.; QUINTERO, E. A.; HARRIS, C. R.; ARCHIBALD, A. M.; RIBEIRO, A. H.; PEDREGOSA, F.; van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020. Disponível em: <<https://doi.org/10.1038/s41592-019-0686-2>>.

WALT, S. van der; SCHÖNBERGER, J. L.; NUNEZ-IGLESIAS, J.; BOULOGNE, F.; WARNER, J. D.; YAGER, N.; GOUILLART, E.; YU, T. *scikit-image: Image processing in Python*. [S.l.]: PeerJ, 2014. <https://doi.org/10.5281/zenodo.11813>. Dataset.

WALT, S. van der; SCHÖNBERGER, J. L.; NUNEZ-IGLESIAS, J.; BOULOGNE, F.; WARNER, J. D.; YAGER, N.; GOUILLART, E.; YU, T.; CONTRIBUTORS the scikit-image. scikit-image: image processing in python. *PeerJ*, v. 2, p. e453, jun 2014. ISSN 2167-8359. Disponível em: <<https://doi.org/10.7717/peerj.453>>.

WANG, G.-N.; CHEN, T.; CHEN, J.-W.; DENG, K.; WANG, R.-D. Simulation of crowd dynamics in pedestrian evacuation concerning panic contagion: A cellular automaton approach. *Chinese Physics B*, IOP Publishing, v. 31, n. 6, p. 060402, 2022.

- WANG, R. Edge detection using convolutional neural network. In: *Advances in Neural Networks–ISNN 2016: 13th International Symposium on Neural Networks, ISNN 2016, St. Petersburg, Russia, July 6-8, 2016, Proceedings 13*. [S.l.]: Springer, 2016. p. 12–20.
- WANG, Z.; BOVIK, A. C.; SHEIKH, H. R.; SIMONCELLI, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, IEEE, v. 13, n. 4, p. 600–612, 2004.
- WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. *Journal of Big data*, SpringerOpen, v. 3, n. 1, p. 1–40, 2016.
- XESS, M.; AGNES, S. A. Analysis of image segmentation methods based on performance evaluation parameters. *Int. J. Comput. Eng. Res*, v. 4, n. 3, p. 68–75, 2014.
- YASIR, M.; HOSSAIN, M. S.; NAZIR, S.; KHAN, S.; THAPA, R. Object identification using manipulated edge detection techniques. *Science*, v. 3, n. 1, p. 1–6, 2022.
- ZHOU, W.; DU, X.; WANG, S. Techniques for image segmentation based on edge detection. In: *2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*. [S.l.: s.n.], 2021. p. 400–403.
- ZIOU, D.; TABBONE, S. Edge detection techniques: An overview. *International Journal of Pattern Recognition and Image Analysis*, v. 4, p. 537–559, 01 1998.