



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS E TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**  
**MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA**

**IÁGSON CARLOS LIMA SILVA**

**APES-SOFT: UMA ABORDAGEM EFICIENTE EM CLASSIFICAÇÃO DE OBJETOS  
3D EM NUVENS DE PONTOS COM REDES NEURAIAS CONVOLUCIONAIS**

**FORTALEZA**

**2024**

IÁGSON CARLOS LIMA SILVA

APES-SOFT: UMA ABORDAGEM EFICIENTE EM CLASSIFICAÇÃO DE OBJETOS 3D  
EM NUVENS DE PONTOS COM REDES NEURAIAS CONVOLUCIONAIS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências e Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Sistemas de Energia Elétrica.

Orientador: Prof. Dr. Pedro Pedrosa Rebouças Filho.

Coorientador: Prof. Dr. Pedro Henrique Feijó de Sousa.

FORTALEZA

2024

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- S58a Silva, Iágson Carlos Lima.  
APES-Soft: Uma Abordagem Eficiente em Classificação de Objetos 3D em Nuvens de Pontos com Redes Neurais Convolucionais / Iágson Carlos Lima Silva. – 2024.  
87 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2024.  
Orientação: Prof. Dr. Pedro Pedrosa Rebouças Filho.  
Coorientação: Prof. Dr. Pedro Henrique Feijó de Sousa.
1. Análise de Ablação. 2. Classificação de Objetos 3D. 3. Desempenho Computacional. 4. Redes Neurais Convolucionais. 5. Testes Estatísticos. I. Título.

CDD 621.3

---

IÁGSON CARLOS LIMA SILVA

APES-SOFT: UMA ABORDAGEM EFICIENTE EM CLASSIFICAÇÃO DE OBJETOS 3D  
EM NUVENS DE PONTOS COM REDES NEURAIAS CONVOLUCIONAIS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica do Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências e Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia Elétrica. Área de Concentração: Sistemas de Energia Elétrica.

Aprovada em: 25 de julho de 2024.

BANCA EXAMINADORA

---

Prof. Dr. Pedro Pedrosa Rebouças Filho (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Pedro Henrique Feijó de  
Sousa (Coorientador)  
Instituto Federal de Educação, Ciência e Tecnologia  
do Ceará (IFCE)

---

Prof. Dr. Fabrício Gonzalez Nogueira  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. José Daniel de Alencar Santos  
Instituto Federal de Educação, Ciência e Tecnologia  
do Ceará (IFCE)

A Deus.

A minha família.

Aos meus amigos.

## AGRADECIMENTOS

A Deus, que com Sua providência meticulosa guiou cada detalhe da minha jornada, expresso minha profunda gratidão. Pois Ele foi essencial em todas as minhas conquistas e superações.

À minha família, expresso com profunda gratidão o meu reconhecimento pelo carinho constante, pelo cuidado incansável, pela atenção dedicada e pelo amor incondicional que permearam cada momento da minha existência. Suas demonstrações de afeto foram o alicerce que sustentou minha jornada, proporcionando-me conforto nos momentos difíceis e celebrando comigo nas conquistas. Vocês são minha fonte inesgotável de força e inspiração, e por isso, minha gratidão é eterna.

À minha noiva, expresso minha profunda gratidão pelo amor, paciência e apoio incondicional ao longo de toda esta jornada. Sua presença e compreensão foram fundamentais para manter o equilíbrio e a motivação, oferecendo suporte emocional e inspiração constante.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo financiamento parcial da pesquisa realizada durante o mestrado.

Ao meu orientador, Prof. Dr. Pedro Pedrosa Rebouças Filho, agradeço pela orientação exemplar e pelos ensinamentos valiosos ao longo desta jornada acadêmica. Sua dedicação e conhecimento foram fundamentais para o meu crescimento como estudante e pesquisador, inspirando-me a alcançar novos patamares.

Ao meu coorientador, Prof. Dr. Pedro Henrique Feijó de Sousa, agradeço pelo apoio contínuo, pela orientação perspicaz ao longo do desenvolvimento desta pesquisa. Sua experiência e conselhos foram um diferencial para superar desafios e aprimorar o trabalho apresentado.

Aos professores e colaboradores de pesquisa do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará (PPGEE-UFC).

Aos amigos e pesquisadores do Laboratório de Processamento de Imagens, Sinais e Computação Aplicada (LAPISCO), cujas contribuições diretas foram fundamentais para o meu crescimento acadêmico, pessoal e profissional.

A todos aqueles que, mesmo não mencionados, contribuíram direta ou indiretamente para esta pesquisa, assim como para o meu desenvolvimento acadêmico, pessoal e profissional.

"Nada na vida deve ser temido, somente compreendido. Agora é hora de compreender mais para temer menos."

(Marie Curie)

## RESUMO

Este trabalho propõe um método eficiente para a classificação de objetos 3D em nuvens de pontos, utilizando *Convolutional Neural Networks* (CNNs). O objetivo principal é propor uma rede neural, chamada APES-Soft, para aprimorar o desempenho computacional e a eficiência na classificação de objetos 3D em nuvens de pontos, baseada na análise de ablação da rede *Attention-based Point cloud Edge Sampling* (APES). A abordagem adotada envolve identificar as características mais relevantes para a classificação através da análise de ablação. Para isso, foram realizados três cenários distintos (Cenário I, Cenário II, e Cenário III), cada um ajustando diferentes aspectos da arquitetura da rede, a fim de definir o melhor candidato a ser proposto como APES-Soft. Os cenários foram conduzidos com rigorosa avaliação, incluindo testes estatísticos além das métricas tradicionais de classificação. Os resultados revelaram que o Cenário II se destaca como uma escolha eficaz para a classificação de objetos 3D em nuvens de pontos, obtendo maior acurácia entre os cenários, atingindo 93,8% e apresentando métricas superiores em comparação com os Cenários I e III, APES (Global) e APES (Local), atingindo a precisão 93,7%, sensibilidade 93,8%, F1-Score 93,7%, *Matthews Correlation Coefficient* (MCC) 93,5% e Índice Jaccard 89,2%, além de ser equivalente com os melhores trabalhos relacionados como CurveNet e DeltaConv. Além disso, em termos de desempenho computacional, o Cenário II apresentou uma redução de tempo significativa em relação a quase todos cenários, exigindo apenas 20,35 horas de treinamento, enquanto obteve uma redução do consumo de memória de 21,89% durante o treinamento do modelo. Os testes estatísticos, incluindo *Analysis of Variance* (ANOVA), Tukey's *Honestly Significant Difference* (HSD), Teste de Kruskal-Wallis e Teste de Friedman, foram realizados para validar o modelo e comparar as diferenças entre os métodos propostos. Os resultados desses testes indicaram que as diferenças observadas entre os métodos não foram estatisticamente significativas, sugerindo uma equivalência estatística entre eles. Este estudo não apenas contribui para o avanço da classificação de objetos 3D, mas também oferece percepções valiosas sobre as características essenciais para uma classificação precisa em nuvens de pontos. Além disso, proporciona uma compreensão mais profunda do balanceamento entre eficiência computacional e desempenho do modelo, aspecto crucial para a viabilidade prática em diversas aplicações.

**Palavras-chave:** Análise de Ablação; Classificação de Objetos 3D; Desempenho Computacional; Redes Neurais Convolucionais; Testes Estatísticos.

## ABSTRACT

This work proposes an efficient method for the classification of 3D objects in point clouds using CNNs. The main objective is to propose a neural network called APES-Soft to enhance computational performance and efficiency in classifying 3D objects in point clouds based on the ablation analysis of the APES network. The adopted approach involves identifying the most relevant features for classification through ablation analysis. For this purpose, three distinct scenarios were conducted (Scenario I, Scenario II, and Scenario III), each adjusting different aspects of the network architecture to define the best candidate to be proposed as APES-Soft. The scenarios were conducted with rigorous evaluation, including statistical tests and traditional classification metrics. The results revealed that Scenario II stands out as an effective choice for the classification of 3D objects in point clouds, achieving higher accuracy among the scenarios, reaching 93.8% and presenting superior metrics compared to Scenarios I and III, APES (Global), and APES (Local), achieving a precision of 93.7%, the sensitivity of 93.8%, F1-Score of 93.7%, MCC of 93.5%, and Jaccard Index of 89.2%, besides being equivalent to the best-related works such as CurveNet and DeltaConv. Additionally, regarding computational performance, Scenario II showed a significant reduction in time compared to almost all scenarios, requiring only 20.35 hours of training while obtaining a memory consumption reduction of 21.89% during model training. Statistical tests, including ANOVA, Tukey's HSD, Kruskal-Wallis Test, and Friedman Test, were performed to validate the model and compare the differences between the proposed methods. The results of these tests indicated that the observed differences between the methods were not statistically significant, suggesting a statistical equivalence among them. This study contributes to advancing 3D object classification and offers valuable insights into the essential features for accurate classification in point clouds. Moreover, it provides a deeper understanding of the balance between computational efficiency and model performance, a crucial aspect for practical viability in various applications.

**Keywords:** Ablation Analysis; Classification of 3D Objects; Computational Performance; Convolutional Neural Networks; Statistical Tests.

## LISTA DE FIGURAS

Figura 1 – Ilustração da operação convolucional em quatro iterações. . . . .	25
Figura 2 – Ilustração da operação de <i>pooling</i> . . . . .	27
Figura 3 – Dropout . . . . .	28
Figura 4 – Gráfico da função de ativação Leaky ReLU. . . . .	29
Figura 5 – Gráfico da função de ativação <i>Softmax</i> . . . . .	30
Figura 6 – Fluxo Metodológico para Proposta da Rede APES-Soft. . . . .	41
Figura 7 – Arquitetura da Rede APES. . . . .	44
Figura 8 – Arquitetura do Cenário I. . . . .	53
Figura 9 – Arquitetura do Cenário II. . . . .	55
Figura 10 – Arquitetura do Cenário III. . . . .	58
Figura 11 – Matriz de confusão para a classificação. . . . .	60
Figura 12 – Visualização dos resultados com as nuvens de pontos do conjunto de teste, para os diferentes métodos. . . . .	66
Figura 13 – Tempo de treinamento por modelo. . . . .	75
Figura 14 – Uso de memória durante o treinamento. . . . .	76

## LISTA DE TABELAS

Tabela 1 – Modelos de CNN usados para Classificação de objetos 3D. . . . .	32
Tabela 2 – Configurações de Treinamento do Modelo e Ambiente Computacional. . . .	43
Tabela 3 – Tabela de geral de resultados da classificação com o conjunto de dados ModelNet40. . . . .	67
Tabela 4 – Avaliação métrica dos modelos. . . . .	68
Tabela 5 – Resultados do teste <i>one-way</i> ANOVA para os cenários. . . . .	72
Tabela 6 – Resultados de <i>p-value</i> dos cenários para o teste Tukey's HSD. . . . .	73
Tabela 7 – Estatísticas do uso de memória <i>Video Random Access Memory</i> (VRAM) (MB). . . . .	77

## LISTA DE ALGORITMOS

Algoritmo 1 – Fluxo de dados na extração de características da Rede APES . . . . .	47
Algoritmo 2 – Fluxo de dados na classificação da Rede APES . . . . .	50
Algoritmo 3 – Fluxo de dados na extração de características do Cenário I . . . . .	53
Algoritmo 4 – Fluxo de dados na extração de características do Cenário II . . . . .	55
Algoritmo 5 – Fluxo de dados na classificação do Cenário II . . . . .	57
Algoritmo 6 – Fluxo de dados na extração de características do Cenário III . . . . .	58

## LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i>
APES	<i>Attention-based Point cloud Edge Sampling</i>
MCC	<i>Matthews Correlation Coefficient</i>
ANOVA	<i>Analysis of Variance</i>
HSD	<i>Honestly Significant Difference</i>
VRAM	<i>Video Random Access Memory</i>
DGCNN	<i>Dynamic Graph CNN</i>
KPConv	<i>Kernel Point Convolution</i>
PVCNN	<i>Point-Voxel CNN</i>
$PT^1$	<i>Point Transformer</i>
$PT^2$	<i>Point Transformer</i>
PCT	<i>Point Cloud Transformer</i>
PACnv	<i>Position Adaptive Convolution</i>
PRA-Net	<i>Point Relation-Aware Network</i>
N2P	<i>Neighbor To Point</i>
GPU	<i>Graphics Processing Unit</i>
LiDAR	<i>Light Detection and Ranging</i>
DL	<i>Deep Learning</i>
TPU	<i>Tensor Processing Unit</i>
SDL	<i>Synergic Deep Learning</i>
DCNN	<i>Deep Convolutional Neural Network</i>
VGG	<i>Visual Geometry Group</i>
SCARA	<i>Selective Compliance Assembly Robot Arm</i>
MLP	<i>Multilayer Perceptron</i>
IoT	<i>Internet of Things</i>
VP	Verdadeiro Positivo
FP	Falso Positivo
VN	Verdadeiro Negativo
FN	Falso Negativo

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>Estado da Arte</b>	<b>18</b>
<b>1.2</b>	<b>Objetivo</b>	<b>19</b>
<b>1.2.1</b>	<i>Objetivo Específico</i>	<b>20</b>
<b>1.3</b>	<b>Produção</b>	<b>20</b>
<b>1.3.1</b>	<i>Artigos publicados</i>	<b>20</b>
<b>1.3.2</b>	<i>Artigos aceitos para publicação</i>	<b>20</b>
<b>1.4</b>	<b>Organização do Texto</b>	<b>21</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>22</b>
<b>2.1</b>	<b>Nuvens de Pontos</b>	<b>22</b>
<b>2.2</b>	<i>Multilayer Perceptron</i>	<b>23</b>
<b>2.3</b>	<i>Convolutional Neural Networks</i>	<b>24</b>
<b>2.3.1</b>	<i>Convolução</i>	<b>24</b>
<b>2.3.2</b>	<i>Batch Normalization</i>	<b>26</b>
<b>2.3.3</b>	<i>Pooling</i>	<b>26</b>
<b>2.3.4</b>	<i>Regularização</i>	<b>27</b>
<b>2.3.4.1</b>	<i>Dropout</i>	<b>27</b>
<b>2.3.5</b>	<i>Camada Densa</i>	<b>28</b>
<b>2.3.6</b>	<i>Funções de Ativação</i>	<b>28</b>
<b>2.3.6.1</b>	<i>LeakyReLU</i>	<b>29</b>
<b>2.3.6.2</b>	<i>Softmax</i>	<b>29</b>
<b>2.4</b>	<b>Análise de ablação</b>	<b>30</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>32</b>
<b>3.1</b>	<b>Redes Pioneiras e Evoluções Diretas</b>	<b>33</b>
<b>3.1.1</b>	<i>PointNet</i>	<b>33</b>
<b>3.1.2</b>	<i>PointNet++</i>	<b>33</b>
<b>3.1.3</b>	<i>PointCNN</i>	<b>33</b>
<b>3.2</b>	<b>Convoluções Adaptativas e Redes Geométricas</b>	<b>34</b>
<b>3.2.1</b>	<i>SpiderCNN</i>	<b>34</b>
<b>3.2.2</b>	<i>PointConv</i>	<b>34</b>

3.2.3	<i>Dynamic Graph CNN (DGCNN)</i> . . . . .	35
3.2.4	<i>Kernel Point Convolution (KPConv)</i> . . . . .	35
3.2.5	<i>Point-Voxel CNN (PVCNN)</i> . . . . .	35
3.2.6	<i>CurveNet</i> . . . . .	36
3.2.7	<i>DeltaConv</i> . . . . .	36
3.3	<b>Redes com Abordagens Atentivas e Transformers</b> . . . . .	36
3.3.1	<i>Point Transformer (PT<sup>1</sup>)</i> . . . . .	37
3.3.2	<i>Point Transformer (PT<sup>2</sup>)</i> . . . . .	37
3.3.3	<i>Point Cloud Transformer (PCT)</i> . . . . .	37
3.3.4	<i>Position Adaptive Convolution (PAConv)</i> . . . . .	38
3.4	<b>Redes com Estratégias de Amostragem e Robustez</b> . . . . .	38
3.4.1	<i>PointASNL</i> . . . . .	39
3.4.2	<i>Point Relation-Aware Network (PRA-Net)</i> . . . . .	39
3.4.3	<i>APES</i> . . . . .	39
4	<b>METODOLOGIA</b> . . . . .	41
4.1	<b>Conjunto de dados</b> . . . . .	42
4.2	<b>Configurações</b> . . . . .	42
4.3	<b>Arquitetura da Rede APES</b> . . . . .	43
4.3.1	<b>Rede Neural Convolutacional (CNN)</b> . . . . .	43
4.3.1.1	<i>Camadas Convolucionais</i> . . . . .	43
4.3.1.1.1	Camada de Incorporação ( <i>Embedding Layer</i> ) . . . . .	44
4.3.1.1.2	Camada de Atenção <i>Neighbor To Point</i> (N2P) . . . . .	45
4.3.1.1.3	Camada de Convolução . . . . .	45
4.3.1.2	<i>Camadas de Pooling</i> . . . . .	45
4.3.1.3	<i>Camadas de Downsampling</i> . . . . .	46
4.3.1.3.1	Global . . . . .	46
4.3.1.3.2	Local . . . . .	46
4.3.1.4	<i>Processo de Extração de Características</i> . . . . .	46
4.3.1.4.1	Camada de Incorporação (A) . . . . .	46
4.3.1.4.2	Camada de Atenção ( $B_1, B_2, B_3$ ) . . . . .	47
4.3.1.4.3	Camada de Convolução ( $C_1, C_2, C_3$ ) . . . . .	47
4.3.1.4.4	Operação de Max Pooling e Armazenamento . . . . .	48

4.3.1.4.5	Camada de <i>Downsampling</i> ( $D_1, D_2$ ) . . . . .	48
4.3.1.4.6	Empacotamento dos Resultados . . . . .	48
<b>4.3.2</b>	<b><i>Perceptron Multicamadas (MLP)</i></b> . . . . .	<b>48</b>
4.3.2.1	<i>Camadas Lineares</i> . . . . .	48
4.3.2.1.1	Primeira Camada Linear . . . . .	49
4.3.2.1.2	Segunda Camada Linear . . . . .	49
4.3.2.1.3	Camada Densa . . . . .	49
4.3.2.2	<i>Regularização</i> . . . . .	49
4.3.2.3	<i>Processo de Classificação</i> . . . . .	50
4.3.2.3.1	Primeira Camada Linear ( $E_1$ ) . . . . .	50
4.3.2.3.2	Primeiro <i>Dropout</i> ( $F_1$ ) . . . . .	50
4.3.2.3.3	Segunda Camada Linear ( $E_2$ ) . . . . .	51
4.3.2.3.4	Segundo <i>Dropout</i> ( $F_2$ ) . . . . .	51
4.3.2.3.5	Camada Densa ( $G$ ) . . . . .	51
<b>4.4</b>	<b><i>Análise de ablação</i></b> . . . . .	<b>52</b>
<b>4.4.1</b>	<b><i>Cenário I</i></b> . . . . .	<b>52</b>
4.4.1.1	<i>Camada de Incorporação (A)</i> . . . . .	53
4.4.1.2	<i>Camada de Atenção (<math>B_1, B_2, B_3</math>)</i> . . . . .	54
4.4.1.3	<i>Camada de Convolução (<math>C_1, C_2, C_3</math>)</i> . . . . .	54
4.4.1.4	<i>Operação de Max Pooling e Armazenamento</i> . . . . .	54
4.4.1.5	<i>Empacotamento dos Resultados</i> . . . . .	54
<b>4.4.2</b>	<b><i>Cenário II</i></b> . . . . .	<b>54</b>
4.4.2.1	<i>Camada de Incorporação (A)</i> . . . . .	55
4.4.2.2	<i>Camada de Atenção (<math>B_1</math>)</i> . . . . .	55
4.4.2.3	<i>Camada de Convolução (<math>C_1</math>)</i> . . . . .	56
4.4.2.4	<i>Divisão do Tensor</i> . . . . .	56
4.4.2.5	<i>Empacotamento dos Resultados</i> . . . . .	56
4.4.2.6	<i>Primeira Camada Linear (<math>E_1</math>)</i> . . . . .	56
4.4.2.6.1	Primeiro <i>Dropout</i> ( $F_1$ ) . . . . .	57
4.4.2.7	<i>Camada Densa (G)</i> . . . . .	57
<b>4.4.3</b>	<b><i>Cenário III</i></b> . . . . .	<b>58</b>
4.4.3.1	<i>Camada de Incorporação (A)</i> . . . . .	59

4.4.3.2	<i>Camada de Atenção (<math>B_1</math>)</i>	59
4.4.3.3	<i>Camada de Convolução (<math>C_1</math>)</i>	59
4.4.3.4	<i>Operação de Max Pooling e Armazenamento</i>	59
4.4.3.5	<i>Resultados</i>	59
<b>4.5</b>	<b>Métricas de Avaliação</b>	<b>60</b>
4.5.1	<i>Acurácia</i>	60
4.5.2	<i>Precisão</i>	61
4.5.3	<i>Sensibilidade</i>	61
4.5.4	<i>F1-Score</i>	61
4.5.5	<i>Coefficiente de Correlação de Matthews (MCC)</i>	61
4.5.6	<i>Índice Jaccard</i>	62
<b>4.6</b>	<b>Testes estatísticos</b>	<b>62</b>
4.6.1	<i>Testes Paramétricos</i>	63
4.6.1.1	<i>ANOVA</i>	63
4.6.1.2	<i>Tukey's HSD</i>	63
4.6.2	<i>Testes Não Paramétricos</i>	64
4.6.2.1	<i>Teste de Kruskal-Wallis</i>	64
4.6.2.2	<i>Teste de Friedman</i>	64
<b>5</b>	<b>RESULTADOS</b>	<b>65</b>
5.1	<b>Visão Geral dos Resultados</b>	65
5.2	<b>Cenário I</b>	69
5.3	<b>Cenário II</b>	69
5.4	<b>Cenário III</b>	70
5.5	<b>Testes Estatísticos</b>	71
5.5.1	<i>ANOVA</i>	71
5.5.2	<i>Tukey's HSD</i>	72
5.5.3	<i>Teste de Kruskal-Wallis</i>	73
5.5.4	<i>Teste de Friedman</i>	74
5.6	<b>Desempenho Computacional</b>	75
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>79</b>
	<b>REFERÊNCIAS</b>	<b>81</b>

## 1 INTRODUÇÃO

Atualmente, as pesquisas em visão computacional e técnicas de processamento de imagens bidimensionais atingiram um estágio avançado (ZHANG *et al.*, 2023), o que despertou uma crescente atenção dos pesquisadores para a análise de cenas tridimensionais, que proporcionam uma representação mais fiel dos ambientes do mundo real (MA *et al.*, 2020; MA *et al.*, 2021).

Nos últimos anos, houve um notável avanço na compreensão do mundo real (UY *et al.*, 2019), o avanço dos recursos computacionais, como as *Graphics Processing Units* (GPUs), juntamente com a abundância de dados tridimensionais provenientes de sensores usados para medir distância, como o *Light Detection and Ranging* (LiDAR) (BURTON *et al.*, 2011), em particular, emerge como um sensor fundamental em diversas áreas, incluindo direção não tripulada, navegação autônoma de robôs e controle de voo de drones (XU *et al.*, 2022).

Essa convergência de avanços tecnológicos e disponibilidade de dados tem catalisado a busca por soluções mais eficientes e precisas para análise e interpretação de cenas tridimensionais, como modelos de aprendizagem profunda (DIAB *et al.*, 2022). Nesse contexto, a classificação de nuvens de pontos, uma tarefa fundamental em muitas aplicações de visão computacional, ganha destaque. As nuvens de pontos, compostas por conjuntos massivos de coordenadas tridimensionais, são frequentemente utilizadas para descrever objetos e ambientes tridimensionais de forma detalhada e precisa (ZHANG *et al.*, 2023). No entanto, a classificação precisa dessas nuvens de pontos é um desafio devido à sua alta dimensionalidade e à presença de ruídos e irregularidades nos dados (YAN *et al.*, 2020).

Nesse contexto, o *Deep Learning* (DL) (LECUN *et al.*, 2015) manifestou-se como uma abordagem promissora para lidar com a complexidade dos dados tridimensionais e extrair informações significativas de nuvens de pontos (ZHANG *et al.*, 2019b). As redes neurais convolucionais CNNs (LECUN *et al.*, 1998), em particular, têm demonstrado sucesso em uma variedade de tarefas de visão computacional, incluindo a classificação de imagens 2D, detecção de padrões em imagens (O'SHEA; NASH, 2015). No entanto, a aplicação de CNNs na análise de nuvens de pontos enfrenta desafios únicos devido à sua natureza tridimensional e à alta dimensionalidade dos dados (CHEN *et al.*, 2019). Apesar dos avanços tecnológicos significativos nas últimas décadas, a análise de nuvens de pontos ainda enfrenta desafios significativos, especialmente no que diz respeito aos recursos computacionais necessários para lidar com conjuntos de dados cada vez maiores e mais complexos, que demandam muito poder computacional.

A complexidade das nuvens de pontos, juntamente com a necessidade de considerar a topologia espacial das informações, requer abordagens adaptadas para o processamento eficiente desses dados. As arquiteturas de redes neurais projetadas especificamente para lidar com nuvens de pontos, como PointNet (Qi *et al.*, 2017a) e suas variantes, têm mostrado resultados promissores na classificação e segmentação de nuvens de pontos. No entanto, essas abordagens muitas vezes exigem recursos computacionais consideráveis, tornando-as impraticáveis para implantação em ambientes com restrições de recursos, como dispositivos móveis e sistemas embarcados.

Além disso, mesmo com os avanços recentes em hardware, como o aumento da capacidade computacional das GPUs e o desenvolvimento de *Tensor Processing Units* (TPUs), a demanda por recursos computacionais para treinamento e inferência de modelos de DL continua sendo um desafio significativo. A escalabilidade desses modelos para lidar com conjuntos de dados cada vez maiores e mais complexos requer soluções inovadoras que otimizem não apenas o desempenho dos algoritmos, mas também a eficiência dos recursos computacionais utilizados.

Nesse sentido, abordagens como o estudo de ablação, frequentemente utilizado na neurociência para investigar a função de regiões específicas do cérebro (MURRAY, 1996), têm sido aplicadas de forma inovadora em trabalhos recentes, como os de Wu *et al.* (2023), Thomas *et al.* (2019), Xu *et al.* (2021). Essa técnica, amplamente adotada, visa identificar e eliminar componentes redundantes ou pouco contribuintes de uma rede neural. Essencial para o desenvolvimento de modelos mais eficientes, a abordagem de ablação proporciona uma compreensão mais profunda sobre a relevância de cada elemento na performance geral do modelo. Isso possibilita a otimização da arquitetura da rede, resultando na redução da complexidade computacional sem comprometer a precisão ou a robustez do modelo resultante.

No restante deste trabalho, exploraremos a análise de ablação aplicada à rede APES (WU *et al.*, 2023) para desenvolver a APES-Soft, uma versão otimizada da rede que mantém o desempenho superior da rede original, enquanto melhora a eficiência computacional e reduz a complexidade do modelo. Além disso, discutiremos os resultados obtidos através de cenários rigorosos de avaliação, destacando as vantagens da APES-Soft em termos de eficiência, escalabilidade e desempenho em comparação com métodos existentes e modelos de referência.

## 1.1 Estado da Arte

O campo do DL se estabeleceu como uma área de pesquisa altamente significativa, destacando-se por uma diversidade de algoritmos, cada qual com suas particularidades distintas. Essas técnicas abrangem uma vasta gama de aplicações no processamento de imagens, com ênfase primordial em atividades como classificação e segmentação (CAI *et al.*, 2020).

Zhang *et al.* (2019a) propuseram um modelo de *Synergic Deep Learning* (SDL) para a classificação de imagens médicas, que utiliza várias *Deep Convolutional Neural Networks* (DCNNs) simultaneamente. Cada par de DCNNs contribui com sua representação de imagem aprendida, alimentando uma rede sinérgica que prevê se o par de imagens de entrada pertence à mesma classe.

Hameed *et al.* (2020) enfatiza uma abordagem de DL para classificação automática de imagens histopatológicas de câncer de mama. Essa abordagem utiliza modelos baseados nas arquiteturas *Visual Geometry Group* (VGG)-16 e VGG-19, que são treinados e combinados para aprimorar a precisão da classificação. Ao adotar essa técnica, o estudo visa superar os desafios associados à análise histopatológica manual, fornecendo uma ferramenta auxiliar eficaz para patologistas no diagnóstico precoce e preciso do câncer de mama.

Pritt e Chern (2017) propuseram o uso do DL para análise de imagens de satélite, abordando sua importância em áreas como resposta a desastres e monitoramento ambiental. Diante da complexidade dessas imagens, os algoritmos tradicionais são inadequados, justificando a necessidade do uso de DL. O sistema desenvolvido, baseado em CNNs foi capaz de classificar objetos em 63 classes diferentes, destacando a eficácia do DL nessas aplicações.

No estudo de Sonata *et al.* (2021), é abordado o emprego do algoritmo de DL usando CNN para o reconhecimento do ambiente circundante, visando à criação da navegação automática essencial para os veículos autônomos. O sistema proposto é responsável por criar e aprender com conjuntos de dados previamente coletados, cujos resultados da simulação revelam uma alta precisão no aprendizado da navegação dos veículos autônomos por meio da análise do ambiente ao redor.

Al-Huda *et al.* (2023) desenvolveram um método híbrido de segmentação de rachaduras em pavimentos, que aproveita a relação entre tarefas de classificação e segmentação para melhorar a precisão da segmentação. O método proposto, chamado KTCAM-Net, utiliza mapas de ativação de classe e uma rede encoder-decoder para alcançar resultados de segmentação de última geração.

Moon *et al.* (2023) investigam o uso de tecnologias de veículos autônomos para a previsão de acidentes em ambientes industriais, propondo um sistema de controle de segurança que identifica fatores de risco através da análise de vídeo em tempo real. Essa abordagem permite detectar e mitigar riscos de forma eficaz, prevenindo acidentes antes que ocorram. A integração com sistemas de gestão de segurança existentes proporciona uma visão coordenada da segurança no local de trabalho, melhorando a resposta a incidentes e promovendo um ambiente mais seguro e eficiente. O contínuo desenvolvimento desses algoritmos promete elevar os padrões de segurança e eficiência nas indústrias.

Wang (2019) desenvolvem uma técnica automatizada para verificar a conformidade de andaimes com as normas de segurança, usando dados de nuvem de pontos 3D. A técnica identifica componentes verticais dos andaimes e plataformas de trabalho horizontais, verificando a presença de rodapés e guardas de proteção. Testes em um canteiro de obras mostraram que o método detecta com precisão andaimes e identifica violações de segurança.

O estudo de Wang *et al.* (2020) apresenta o PointNetRGPE, um método de aprendizado profundo para estimar a pose de agarramento de um robô *Selective Compliance Assembly Robot Arm* (SCARA) usando nuvens de pontos. O modelo combina informações de nuvem de pontos e classes em um vetor ponto-classe, utilizando redes inspiradas no PointNet para estimar translatação 3D e rotação 1D, com uma arquitetura adaptada para lidar com simetria rotacional na direção do eixo z. Os testes realizados mostram altas taxas de sucesso na estimativa de pose, indicando sua eficácia para aplicações industriais de carregamento e descarregamento robótico.

E por fim, Wu *et al.* (2023) abordam a amostragem de nuvem de pontos, destacando a falta de exploração nesse campo. Eles propõem o método APES, baseado em atenção, que seleciona pontos salientes na borda da nuvem de pontos. Este método demonstra um desempenho superior em comparação com outras abordagens, tanto qualitativa quanto quantitativamente, em tarefas de referência comuns. Além disso, o estudo também aborda segmentação e classificação, sendo a classificação de objetos 3D utilizando nuvem de pontos um dos principais motivadores para esta dissertação, com o intuito de contribuir para linha de pesquisa e aprimorar a classificação para propor um método igualmente robusto.

## 1.2 Objetivo

O objetivo deste trabalho é propor uma rede neural, chamada APES-Soft, para aprimorar o desempenho computacional e a eficiência na classificação de nuvens de pontos,

baseada na análise de ablação da rede APES.

### **1.2.1 Objetivo Específico**

- a) Identificar os principais componentes da rede APES que contribuem para o desempenho computacional.
- b) Avaliar o impacto da remoção seletiva de camadas da rede APES na classificação de nuvens de pontos.
- c) Analisar o rendimento de cada experimento por meio de diversas métricas de avaliação, como Acurácia, Precisão, Sensibilidade, F1-Score, MCC e Índice Jaccard, além de avaliar o tempo de execução e a utilização de memória.
- d) Realizar uma comparação entre os resultados obtidos através do modelo otimizado e os métodos presentes no estado da arte.
- e) Realizar uma análise estatística comparativa entre o desempenho do modelo APES e os resultados dos cenários, utilizando testes estatísticos adequados para avaliar a significância das diferenças observadas.
- f) Propor a APES-Soft utilizando os resultados obtidos dos cenários realizados na análise de ablação.

## **1.3 Produção**

A produção científica durante o mestrado resultou na submissão e publicação de artigos científicos. A seguir estão listadas as publicações:

### **1.3.1 Artigos publicados**

1. LIMA, Raoni Alves de et al. Impacto da Geração de Plantas Fotovoltaicas Flutuantes sobre a Matriz Energética de Estados do Semiárido Brasileiro. *Revista Brasileira de Energias Renováveis*, v. 29, n. 1, 2023. Disponível em: <<https://doi.org/10.47168/rbe.v29i1.778>>. Acesso em: 1 Jun. 2024.

### **1.3.2 Artigos aceitos para publicação**

1. SANTOS, L. O.; SILVA, I. C. L.; DOS SANTOS, M. A.; MEDEIROS, A. G.; REBOUÇAS FILHO, P. P. (2024). Efficient Classification of Depression using EEG through Spectral

Graph Analysis. In *The 37th IEEE International Symposium on Computer-Based Medical Systems (IEEE CBMS2024)*.

#### **1.4 Organização do Texto**

Esta dissertação está estruturada em seis capítulos para fornecer uma visão abrangente do estudo realizado.

O Capítulo 1 apresenta a introdução ao tema, contextualizando a importância e relevância da pesquisa.

No Capítulo 2, são apresentadas as informações teóricas essenciais sobre as técnicas empregadas neste trabalho, fornecendo uma base sólida para o entendimento do leitor.

O Capítulo 3 aborda os métodos relacionados à pesquisa, contextualizando o estado da arte e discutindo trabalhos relevantes na área de estudo.

No Capítulo 4, são detalhadas a metodologia e as abordagens utilizadas na condução da pesquisa, fornecendo uma compreensão clara dos procedimentos adotados.

O Capítulo 5 apresenta os resultados da pesquisa e as discussões relacionadas, analisando os dados obtidos e destacando suas implicações.

Por fim, o Capítulo 6 apresenta as conclusões derivadas da pesquisa realizada, além de oferecer sugestões para trabalhos futuros, visando contribuir para o avanço contínuo do conhecimento na área.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, exploraremos os conceitos fundamentais relacionados à nuvem de pontos e suas diversas aplicações. Além disso, serão apresentados os principais métodos de aprendizado de máquina utilizados para a classificação de nuvens de pontos, com foco especial nas redes neurais artificiais *Multilayer Perceptron* (MLP) e CNN.

### 2.1 Nuvens de Pontos

As nuvens de pontos são uma forma versátil de representação de dados, encontrando aplicação em diversas áreas, como ciência, indústria e engenharia, incluindo campos como veículos autônomos, reconstrução 3D e até mesmo medicina. Essas nuvens são compostas por conjuntos de pontos definidos em um espaço multidimensional, onde cada ponto é caracterizado por suas coordenadas, refletindo sua posição precisa em um espaço tridimensional (ZHANG *et al.*, 2023).

No âmbito dos veículos autônomos, a construção do vetor de características da nuvem de pontos considera não apenas o contexto semântico do ambiente, mas também as características geométricas 3D (BALADO *et al.*, 2019) e a distribuição espacial dos pontos. Esses avanços têm contribuído significativamente para a automação e a inteligência na classificação dessas nuvens de pontos, melhorando a compreensão do ambiente por parte dos veículos autônomos.

Além disso, na reconstrução de modelos tridimensionais, os dados densos das nuvens de pontos desempenham um papel crucial. Ao contrário das imagens, essas nuvens e malhas oferecem uma representação estruturada e precisa do espaço tridimensional. Por meio de algoritmos especializados, o processo de reconstrução tridimensional extrai informações discretas do espaço e estabelece conexões entre os pontos para criar modelos 3D detalhados e precisos. Essa abordagem, baseada em nuvens de pontos, equilibra eficientemente as demandas do usuário com os custos econômicos associados às tarefas de reconstrução (ZHANG *et al.*, 2023).

Na área da medicina, onde a tecnologia de imagens médicas continua a evoluir, as nuvens de pontos desempenham um papel significativo. Elas oferecem benefícios potenciais que vão desde o planejamento prévio de cirurgias até a orientação durante os procedimentos (WEBER *et al.*, 2024). Assim, as nuvens de pontos podem influenciar diretamente nos resultados e na eficiência dos procedimentos médicos, oferecendo uma representação detalhada e precisa

das estruturas anatômicas.

Em resumo, as nuvens de pontos desempenham um papel central no avanço de tecnologias como a condução autônoma, reconstrução tridimensional e medicina, proporcionando uma representação rica e detalhada do ambiente físico que pode ser explorada de diversas maneiras para aprimorar a eficiência e a precisão em uma variedade de aplicações.

## 2.2 Multilayer Perceptron

O modelo conhecido como MLP é amplamente utilizado em aprendizado supervisionado para tarefas de classificação e regressão. Ele consiste em uma rede neural artificial formada por múltiplos Perceptrons simples, que juntos lidam com a resolução de problemas complexos e não lineares (HAYKIN, 2001).

O Perceptron simples, proposto por Rosenblatt (1958), opera combinando linearmente suas entradas ponderadas por pesos específicos. Originalmente destinado a resolver problemas lineares, sua saída é definida pela Equação 2.1,

$$\hat{y} = \phi \left( \sum_{i=1}^p x_i w_i + b \right) \quad (2.1)$$

onde  $\phi$  é a função de ativação,  $x_i$  representa o vetor de entrada,  $w_i$  é o vetor de pesos associados às entradas (ambos de dimensão  $p$ ), e  $b$  é o viés. Durante o treinamento, os parâmetros  $w_i$  e  $b$  são ajustados iterativamente conforme a Equação 2.2,

$$w_i(t+1) = w_i(t) + \eta (y - \hat{y}(t)) x_i, \quad \forall i = 1, \dots, p \quad (2.2)$$

onde  $t$  denota a iteração do treinamento,  $y$  é a classe da amostra e  $\hat{y}(t)$  é a predição naquela iteração. A taxa de aprendizagem  $\eta$  regula a magnitude da atualização dos pesos em relação ao erro de predição.

Como o Perceptron simples não é capaz de resolver problemas não lineares, o MLP foi desenvolvido como uma rede de Perceptrons interligados em múltiplas camadas, conhecidas como neurônios. Cada neurônio utiliza uma função de ativação não linear, como a sigmoide, introduzindo assim a não linearidade nos cálculos. A saída de cada neurônio é definida pela Equação 2.3,

$$\hat{y}_{l_k} = \begin{cases} \phi_{0_k} \left( \sum_{i=1}^p x_i w_{0_k} + b_{0_k} \right) & \text{se } l = 0 \\ \phi_{l_k} \left( \sum_{i=1}^D \hat{y}_{l-1} w_{l_k} + b_{l_k} \right) & \text{se } l \neq 0 \end{cases} \quad (2.3)$$

onde  $l$  representa o número da camada e  $k$  é a posição do neurônio dentro dessa camada. O vetor  $x$  é o conjunto de atributos de dimensão  $p$ , enquanto  $D$  é o número de neurônios nas camadas ocultas. A função de ativação  $\phi$  introduz a não linearidade nos cálculos.

Durante o treinamento, o MLP utiliza o algoritmo de *Backpropagation* (RUMELHART *et al.*, 1986) para ajustar os pesos da rede em cada iteração e reduzir o erro de predição. Os pesos são atualizados conforme a Equação 2.4,

$$\hat{w}_{l_{k_i}}(t+1) = \begin{cases} w_{0_{k_i}}(t) + \eta \cdot \delta_{l_i}(t) \cdot x_k(t) & \text{se } l = 0 \\ w_{l_{k_i}}(t) + \eta \cdot \delta_{(l+1)_i}(t) \cdot \hat{y}_{l_k}(t) & \text{se } l \neq 0 \end{cases} \quad (2.4)$$

onde  $\delta_{l_k}(t)$  representa o gradiente local na camada  $l$ -ésima do neurônio  $k$ , e pode ser calculado como na Equação 2.5:

$$\delta_{l_k} = \begin{cases} \phi'_{l_k}[u_{l_k}(t)] \frac{1}{2M} \sum_{m=1}^M [y_{l_m} - \hat{y}_{l_m}(t)]^2 & \text{se } l = z \\ \phi'_{l_k}[u_{l_k}(t)] \sum_{j=1}^J w_{l_{k_i}}(t) - \delta_{(l+1)_j}(t) & \text{se } l \neq 0 \end{cases} \quad (2.5)$$

onde  $z$  representa a camada de saída da rede, composta por  $M$  neurônios, e  $l$  denota qualquer outra camada com  $J$  neurônios. A função  $\phi'$  é a derivada da função de ativação utilizada.

Considerando as restrições enfrentadas pelo MLP ao lidar com conjuntos de dados de alta dimensionalidade, como imagens e, no caso deste estudo, nuvens de pontos, surgiram novas arquiteturas de redes neurais, como as CNNs. Na seção seguinte, serão abordadas algumas definições fundamentais sobre as CNNs.

## 2.3 Convolutional Neural Networks

Vamos explorar, nesta seção, os elementos fundamentais que compõem a implementação, treinamento e validação das CNNs. Nosso objetivo é fornecer uma compreensão completa sobre o funcionamento dessas redes, destacando os principais métodos, operações e funções de ativação que as impulsionam.

### 2.3.1 Convolução

A operação de convolução desempenha um papel decisivo nas CNNs, permitindo a extração de características importantes das imagens de entrada. Essa operação consiste na aplicação de filtros sobre a imagem, onde cada filtro é projetado para detectar padrões específicos, como bordas, texturas e gradientes, ou seja, informações relevantes das imagens (ALBAWI *et al.*, 2017). Durante a convolução, o filtro é deslizado sobre a imagem, calculando-se a soma

ponderada dos valores dos *pixels* em cada posição, o que resulta em um mapa de características que destaca as características relevantes da imagem.

Matematicamente, a convolução entre um filtro  $K$  e uma imagem  $I$  é representada pela Equação 2.6, onde  $S(i, j)$  é o valor resultante na posição  $(i, j)$  do mapa de características:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \tag{2.6}$$

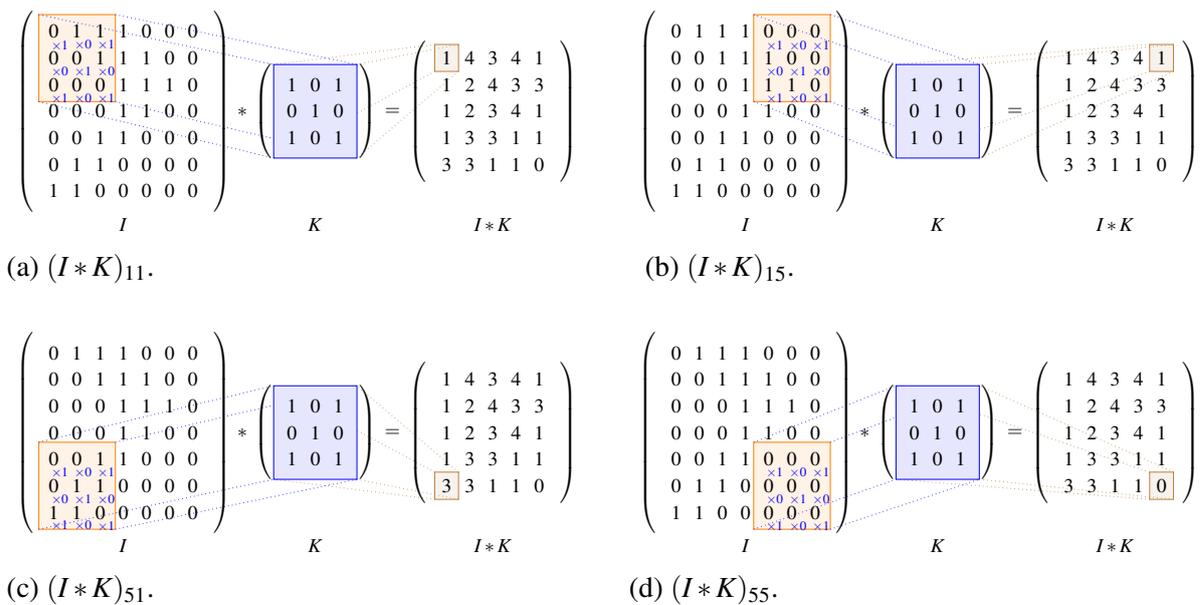
Essa operação é exemplificada em quatro iterações na Figura 1, onde um filtro  $3 \times 3$  é aplicado em uma matriz  $7 \times 7$ . O deslocamento do filtro, conhecido como passo (*stride*), influencia na ordem da matriz de saída, conforme expresso na Equação 2.7.

$$\gamma = \frac{\lambda - \phi}{\sigma} + 1 \tag{2.7}$$

Nesta equação,  $\gamma$  representa a ordem da matriz resultante,  $\lambda$  é a ordem da matriz de entrada,  $\phi$  é a ordem do filtro da convolução, e  $\sigma$  é o tamanho do passo (*stride*).

Na Figura 1a, é evidenciada a iteração que calcula o elemento  $(1, 1)$  da matriz resultante, realizando a soma dos produtos elemento por elemento do filtro com a matriz de entrada. Por outro lado, a Figura 1b demonstra a iteração necessária para obter o elemento  $(1, 5)$  da matriz de saída  $I * K$ . As Figuras 1c e 1d esboçam a mesma operação para os elementos  $(5, 1)$  e  $(5, 5)$  de  $I * K$ , respectivamente.

Figura 1 – Ilustração da operação convolucional em quatro iterações.



Fonte: Elaborada pelo autor.

Nas próximas seções, exploraremos os métodos utilizados para aprimorar a operação de convolução e extrair características significativas dos dados de entrada nas CNNs.

### 2.3.2 *Batch Normalization*

A *Batch Normalization*, também conhecida como Normalização em Lote, proposta por Ioffe e Szegedy (2015), apresenta uma abordagem para otimizar e estabilizar o treinamento de redes neurais profundas, como discutido por Luo *et al.* (2018). Essa técnica consiste na normalização das entradas de cada camada por lotes pré-definidos, utilizando média ( $\mu$ ) e desvio padrão ( $\sigma$ ), conforme a equação,

$$H' = \frac{H - \mu}{\sigma}, \quad (2.8)$$

onde  $H$  representa as ativações originais de uma camada específica. Ao estabilizar o processo de aprendizagem da rede neural, a *Batch Normalization* reduz o número de épocas necessárias para o treinamento. Em vez de atualizar os pesos somente após a passagem completa pela rede, como é feito tradicionalmente, os pesos são atualizados após o processamento de cada lote. Isso contribui para uma convergência mais rápida e eficiente durante o treinamento.

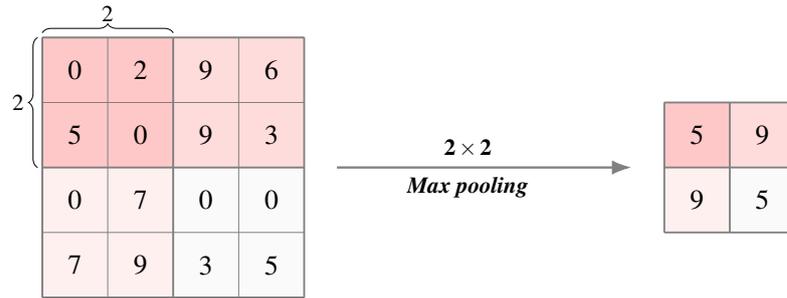
### 2.3.3 *Pooling*

Uma estratégia comum para reduzir a dimensionalidade dos dados de entrada de uma CNN é empregar a operação de *pooling* na saída da camada convolucional. Esse método utiliza campos receptivos sobre a matriz de entrada para diminuir a quantidade de informação, resultando na redução da dimensão dos dados (SCHERER *et al.*, 2010). Assim como um filtro na convolução, esses campos receptivos também percorrem os dados de entrada, porém com um *stride* igual ao tamanho do campo receptivo.

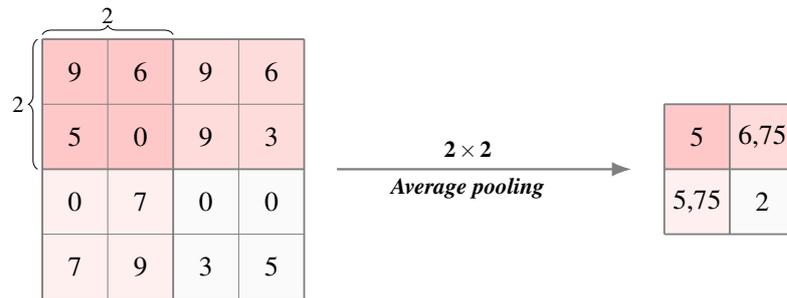
Dois tipos de *pooling* são comumente utilizados, ambos ilustrados na Figura 2, onde os filtros receptivos têm tamanho 2x2 e percorrem uma matriz 4x4. Ao final dessas operações, a matriz de saída terá dimensão 2x2, evidenciando a redução de dimensionalidade que é o objetivo da operação.

O primeiro é o *max pooling* visto na Figura 2a, é a operação em que o maior valor do campo receptivo é escolhido e atribuído à matriz de saída. Já o segundo *pooling* é o *average pooling* que calcula a média dos valores campo receptivo e concede à matriz de saída (GOODFELLOW *et al.*, 2016), como observado na Figura 2b.

Figura 2 – Ilustração da operação de *pooling*.



(a) Ilustração do *max pooling*.



(b) Ilustração do *average pooling*.

Fonte: Elaborada pelo autor.

### 2.3.4 Regularização

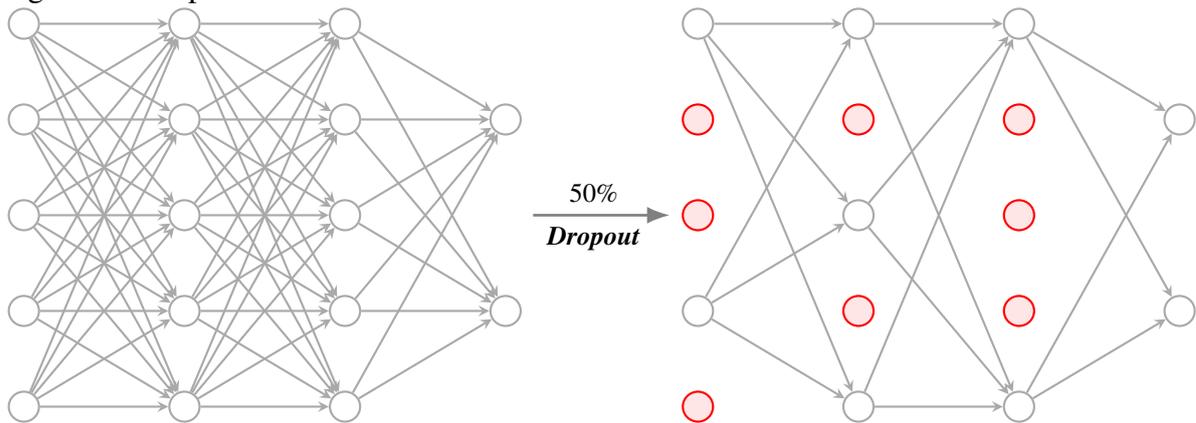
A regularização é uma técnica fundamental no campo do aprendizado de máquina, essencial para evitar o *overfitting* e melhorar a capacidade de generalização dos modelos. Ela consiste em adicionar termos de penalidade à função de custo durante o treinamento do modelo (DEY *et al.*, 2017), o que ajuda a controlar a complexidade do modelo. O *dropout* é uma técnica popular em redes neurais, onde unidades aleatórias são temporariamente removidas durante o treinamento para evitar a co-adaptação dos neurônios.

#### 2.3.4.1 Dropout

A técnica *Dropout*, proposta por Srivastava *et al.* (2014), é uma estratégia de regularização utilizada em redes neurais para mitigar o risco de *overfitting*. Consiste na aleatória desativação de uma parcela dos neurônios durante o treinamento da rede, como ilustrado na Figura 3.

Neste exemplo, foi aplicado um *dropout* de 50%, o que significa que metade dos neurônios em cada camada foram desativados aleatoriamente durante o treinamento, representados em vermelho. Esses neurônios desativados não contribuem diretamente para a propagação

Figura 3 – Dropout



Fonte: Elaborada pelo autor.

dos sinais na rede e seus pesos não são atualizados durante o processo de retropropagação (*backpropagation*).

### 2.3.5 Camada Densa

A camada densa, também conhecida como totalmente conectada, desempenha um papel muito importante em redes neurais, consistindo em neurônios que processam e distribuem as informações ponderadas de suas entradas. Quando posicionada na camada de saída de uma rede neural, é responsável por classificar o vetor de características resultante das operações de convolução e *pooling* (LECUN *et al.*, 1989). Ao longo da rede, os dados de entrada são submetidos a filtragens por convolução e redução de dimensão por *pooling*. No topo da rede, antes da camada densa, resta apenas um vetor de características, a partir do qual ocorre a predição das classes presentes no conjunto de treinamento.

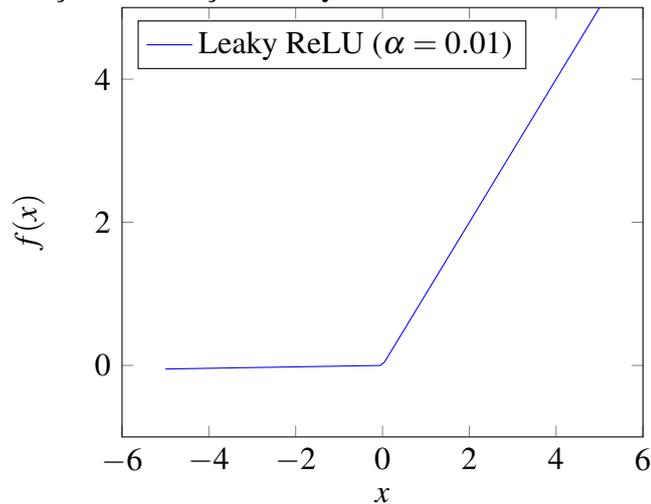
### 2.3.6 Funções de Ativação

Nas redes neurais, as funções de ativação desempenham um papel fundamental ao permitir que os modelos aprendam relações complexas nos dados, introduzindo não linearidades (RASAMOELINA *et al.*, 2020). Elas são aplicadas em cada camada da rede neural para garantir que as saídas dos neurônios possam representar de forma adequada as características dos dados, capturando sua complexidade.

### 2.3.6.1 LeakyReLU

A função de ativação *LeakyReLU* (*Rectified Linear Unit*) é uma alternativa à função *ReLU* tradicional, enquanto a *ReLU* zera valores negativos, a *LeakyReLU* permite que gradientes pequenos passem para trás durante o treinamento, mitigando o problema de "neurônios mortos" que ocorre quando os gradientes se tornam zero (XU *et al.*, 2015).

Figura 4 – Gráfico da função de ativação Leaky ReLU.



Fonte: Elaborada pelo autor.

Na Figura 4, é perceptível a transição em que a saída deixa de ser zero, assemelhando-se à função *ReLU*, e passa a ser proporcional à entrada da função de ativação. Como consequência, os neurônios permanecem sempre ativos; contudo, quando a saída é negativa, a informação é propagada com uma proporção menor em comparação à informação positiva.

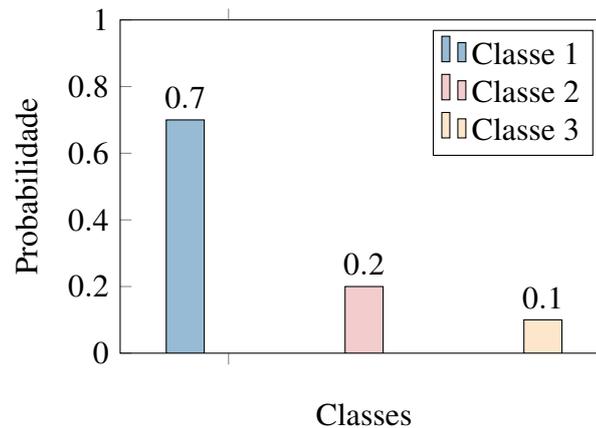
### 2.3.6.2 Softmax

A função *Softmax*, frequentemente empregada nas camadas de classificação de redes neurais (BRIDLE, 1989), desempenha a função de transformar o espaço do vetor de saída da rede em um espaço probabilístico. Nesse espaço, os valores estão normalizados entre 0 e 1, e a soma de todos os valores resulta em 1. A função *Softmax* é matematicamente expressa pela Equação 2.9:

$$\sigma(\mathbf{z}) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.9)$$

onde  $i = 1, \dots, K$ ,  $z_i = (z_1, \dots, z_k) \in \mathbb{R}^K$ , representando a saída do neurônio  $i$  da última camada da rede, e  $K$  denota o número de classes no problema.

Figura 5 – Gráfico da função de ativação *Softmax*.



Fonte: Elaborada pelo autor.

Na Figura 5, temos um exemplo de aplicação da função *Softmax* em um problema de classificação com três classes (Classe 1, Classe 2 e Classe 3). Cada barra no gráfico representa a probabilidade de um dado exemplo pertencer a uma das classes. As alturas das barras indicam essas probabilidades, e como a função *Softmax* normaliza esses valores para que a soma de todas as probabilidades seja igual a 1, as alturas das barras representam as probabilidades relativas de pertencimento a cada classe.

Por exemplo, para o primeiro exemplo representado, a barra mais alta indica que há uma probabilidade de aproximadamente 70% de pertencer à Classe 1, uma probabilidade de aproximadamente 20% de pertencer à Classe 2 e uma probabilidade de aproximadamente 10% de pertencer à Classe 3.

Essa visualização demonstra como a função *Softmax* é útil em problemas de classificação, ao fornecer uma distribuição de probabilidade sobre as possíveis classes para cada exemplo de entrada. Isso permite que o modelo de rede neural faça previsões não apenas sobre a classe mais provável, mas também sobre a incerteza associada a essa previsão.

## 2.4 Análise de ablação

Estudos de ablação são comumente empregados na área da neurociência para investigar sistemas biológicos complexos, como o sistema nervoso central de organismos como a *Drosophila*<sup>1</sup>, que tem sido amplamente estudado, assim como o cérebro de vertebrados e, de maneira mais complexa e fascinante, o cérebro humano. Historicamente, esses estudos têm sido

<sup>1</sup> *Drosophila*, também conhecida como drosófila, é um gênero que inclui um grande número de espécies de pequenas moscas pertencentes à família *Drosophilidae*. Elas são também chamadas de moscas-das-frutas e, menos frequentemente, de moscas-do-vinagre.

utilizados para compreender a estrutura e a organização do cérebro, mapeando as respostas a estímulos externos em diferentes regiões do neocórtex (MEYES *et al.*, 2019).

Nos estudos de ablação na neurociência, onde o tecido neural é danificado de maneira controlada, investigamos como esse dano afeta a capacidade do cérebro de executar determinadas tarefas (MEYES *et al.*, 2019). Por meio dessa metodologia, o autor buscou compreender o papel funcional das regiões cerebrais afetadas, além de explorar a estrutura e a organização do processamento de estímulos externos no cérebro humano, provando assim ser um método eficaz investigar sistemas neurais grandes e complexos.

No contexto de CNNs, a análise de ablação tem uma aplicação semelhante. Em vez de tecido neural, elementos específicos das redes neurais, como camadas, filtros ou unidades individuais, são removidos ou desativados para entender seu impacto no desempenho geral da rede (MONTAHA *et al.*, 2021). Essa metodologia permite identificar quais componentes da CNN são essenciais para a sua capacidade de classificação ou detecção de padrões.

A análise de ablação em CNNs é fundamental para melhorar o desempenho computacional ao simplificar a estrutura da rede, reduzindo o tempo de computação e o uso de memória sem comprometer a precisão. Além disso, permite a identificação e ajuste de componentes que podem estar causando gargalos de desempenho, tornando o processamento de dados mais eficiente.

Ao eliminar componentes que provocam *overfitting* nos dados de treinamento, a rede se torna mais capaz de generalizar para novos dados, melhorando o desempenho em testes e aplicações reais. Redes neurais menores e simplificadas consomem menos energia, o que é crucial para dispositivos com recursos limitados, como *smartphones* e dispositivos *Internet of Things* (IoT). Além disso, redes simplificadas são mais fáceis de interpretar e analisar, oferecendo uma compreensão mais profunda sobre o funcionamento interno da rede.

Assim, a análise de ablação não só aprimora a compreensão teórica das CNNs, mas também promove eficiência prática e maior aplicabilidade em diversos contextos tecnológicos.

### 3 TRABALHOS RELACIONADOS

Este capítulo oferece uma visão ampla e detalhada das contribuições mais significativas no campo do processamento de nuvens de pontos, destacando avanços recentes e identificando oportunidades para pesquisas futuras. A Tabela a seguir oferece uma visão objetiva e comparativa dos diferentes modelos de CNNs utilizados para a classificação de objetos 3D.

Tabela 1 – Modelos de CNN usados para Classificação de objetos 3D.

Métodos	Modelos	Descrição
Redes Pioneiras e Evoluções Diretas	PointNet (QI <i>et al.</i> , 2017a)	Lida diretamente com nuvens de pontos irregulares, mantendo a invariância de permutação para classificação de objetos e análise semântica de cenas.
	PointNet++ (QI <i>et al.</i> , 2017b)	Supera a PointNet ao reconhecer padrões locais refinados e generalizar para cenas complexas com uma rede neural hierárquica.
	PointCNN (LI <i>et al.</i> , 2018)	Pondera e reordena características de nuvens de pontos através da transformação X, adaptando as CNNs para dados irregulares e desordenados.
Convoluções Adaptativas e Redes Geométricas	SpiderCNN (XU <i>et al.</i> , 2018)	Extraí características de nuvens de pontos 3D usando convoluções adaptadas para conjuntos de pontos irregulares, capturando informações locais e geodésicas.
	PointConv (WU <i>et al.</i> , 2019)	Aplica filtros dinâmicos em nuvens de pontos 3D, obtendo resultados de ponta em segmentação semântica.
	DGCNN (WANG <i>et al.</i> , 2019)	Utiliza o módulo EdgeConv para incorporar informações de vizinhança local em tarefas de classificação e segmentação de nuvens de pontos.
	KPConv (THOMAS <i>et al.</i> , 2019)	Elimina a necessidade de representações intermediárias, usando pontos de <i>kernel</i> para convoluções adaptativas em nuvens de pontos.
	PVCNN (LIU <i>et al.</i> , 2019)	Combina representações em pontos e convoluções em <i>voxels</i> para melhorar a eficiência e precisão no processamento de nuvens de pontos.
	CurveNet (MUZAHID <i>et al.</i> , 2020)	Utiliza direções de curvatura principal de objetos 3D para melhorar o reconhecimento de objetos em uma rede neural convolucional 3D.
Redes com Abordagens Atentivas e <i>Transformers</i>	DeltaConv (WIERSMA <i>et al.</i> , 2022)	Constrói filtros anisotrópicos em nuvens de pontos para representar e processar informações direcionais explicitamente, melhorando o desempenho em diversas tarefas de classificação.
	PT <sup>1</sup> (ENGEL <i>et al.</i> , 2021)	Utiliza um mecanismo de atenção local-global para capturar relações espaciais e informações de forma em conjuntos de pontos desordenados.
	PT <sup>2</sup> (ZHAO <i>et al.</i> , 2021)	Adapta redes autoatentivas para nuvens de pontos 3D, melhorando a segmentação semântica de cenas e classificação de objetos.
	PCT (GUO <i>et al.</i> , 2021)	Adapta o <i>Transformer</i> para lidar com a complexidade e irregularidade das nuvens de pontos, capturando contexto local e global.
Redes com Estratégias de Amostragem e Robustez	PACConv (XU <i>et al.</i> , 2021)	Utiliza <i>kernels</i> de convolução adaptativos para processar dados irregulares e não ordenados em nuvens de pontos, mantendo eficiência computacional e precisão.
	PointASNL (YAN <i>et al.</i> , 2020)	Reajusta pesos dos vizinhos de pontos amostrados para lidar com ruído e <i>outliers</i> , capturando dependências locais e não locais.
	PRA-Net (CHENG <i>et al.</i> , 2021)	Integra aprendizado de contextos intra-região e relações inter-região para análise robusta de nuvens de pontos em diversas tarefas.
	APES (WU <i>et al.</i> , 2023)	Seleciona pontos em nuvens de pontos usando um mecanismo de atenção inspirado na detecção de bordas de Canny, visando melhorar a amostragem para reconhecimento e segmentação.

Fonte: Elaborada pelo autor.

A Tabela 1 oferece uma visão objetiva e comparativa das diferentes redes utilizadas para a classificação de objetos 3D. A tabela está organizada em quatro categorias principais:

Redes Pioneiras e Evoluções Diretas, Convoluções Adaptativas e Redes Geométricas, Redes com Abordagens Atentivas e *Transformers*, e Redes com Estratégias de Amostragem e Robustez. Cada categoria destaca métodos específicos, seus modelos correspondentes e uma breve descrição de suas características e inovações, detalhadas abaixo.

### **3.1 Redes Pioneiras e Evoluções Diretas**

Esta seção específica detalha os primeiros desenvolvimentos em redes neurais para nuvens de pontos e suas evoluções diretas. Cada subseção aqui descreve um modelo específico, destacando suas inovações e melhorias.

#### **3.1.1 *PointNet***

A *PointNet* (Qi *et al.*, 2017a) oferece uma abordagem para lidar com nuvens de pontos, operando diretamente com esses dados irregulares em vez de convertê-los em formas regulares. Com foco na invariância de permutação dos pontos de entrada, esta rede neural é aplicável em várias áreas, como classificação de objetos e análise semântica de cenas. Demonstrando eficiência comparável ou superior ao estado da arte, a *PointNet* alcança uma acurácia de 89,2% em testes empíricos. Além disso, análises teóricas aprofundam a compreensão do aprendizado da rede e sua robustez em relação a perturbações nos dados de entrada, tornando-a uma ferramenta valiosa para diversas aplicações em análise de nuvens de pontos.

#### **3.1.2 *PointNet++***

A *PointNet++* (Qi *et al.*, 2017b) supera a *PointNet* ao abordar sua limitação na identificação de estruturas locais nos pontos, essencial para reconhecer padrões refinados e generalizar para cenas complexas. Introduz uma rede neural hierárquica que aprende características locais em diferentes escalas e propõe camadas de aprendizado de conjunto inovadoras para lidar com variações na densidade dos pontos. Experimentos mostram sua eficácia superior em *benchmarks* desafiadores de nuvens de pontos 3D, alcançando uma acurácia de 91,9%.

#### **3.1.3 *PointCNN***

A *PointCNN* (Li *et al.*, 2018) apresenta um *framework* simples e eficaz para o aprendizado de características em nuvens de pontos, superando desafios de ordenação e irregularidade.

Ao introduzir a transformação  $X$ , que pondera e reordena as características associadas aos pontos, o PointCNN generaliza as CNNs para esse contexto. Experimentos demonstram que o PointCNN alcança ou supera o desempenho dos métodos mais avançados em diversos conjuntos de dados e tarefas desafiadoras, com uma acurácia de 92,2%.

## 3.2 Convoluções Adaptativas e Redes Geométricas

Nesta seção, são abordadas as redes que empregam convoluções adaptativas e abordagens geométricas. Essas técnicas avançadas são essenciais para lidar com a natureza irregular e complexa das nuvens de pontos, proporcionando uma análise mais precisa e eficiente.

### 3.2.1 *SpiderCNN*

A SpiderCNN (XU *et al.*, 2018) oferece uma arquitetura de convolução otimizada para extrair características de nuvens de pontos 3D, desafiando as limitações das CNNs tradicionais que dependem de estruturas regulares. Ela emprega unidades chamadas SpiderConv, adaptadas para operar em conjuntos de pontos irregulares, capturando informações locais e geodésicas de forma eficaz. Ao combinar funções simples com polinômios de Taylor, o SpiderCNN consegue extrair características profundas e semânticas em várias escalas. Experimentos revelam que o SpiderCNN alcança uma acurácia notável de 92,4% em *benchmarks* padrão, além de apresentar um desempenho competitivo na segmentação de objetos.

### 3.2.2 *PointConv*

A PointConv (WU *et al.*, 2019) é uma técnica inovadora para realizar operações de convolução em nuvens de pontos 3D, superando os desafios específicos apresentados por sua natureza irregular. Ao estender o conceito de filtro dinâmico, o PointConv oferece uma abordagem eficaz para construir redes convolucionais profundas em ambientes tridimensionais. Experimentos em diversos conjuntos de dados, como ModelNet40, ShapeNet e ScanNet, demonstram que as redes construídas com o PointConv alcançam resultados de ponta em tarefas desafiadoras de segmentação semântica, com uma acurácia de 92,5%. Além disso, ao converter o conjunto de dados CIFAR-10 em uma nuvem de pontos, observa-se que redes baseadas no PointConv conseguem rivalizar o desempenho de redes convolucionais em imagens 2D de estrutura semelhante.

### 3.2.3 DGCNN

A DGCNN (WANG *et al.*, 2019) propõe o módulo EdgeConv para tarefas de alto nível em nuvens de pontos, como classificação e segmentação, adaptando conceitos de CNNs para esse contexto. Ao contrário de abordagens anteriores, o EdgeConv incorpora informações de vizinhança local e pode ser empilhado para aprender propriedades globais. Sua aplicação em sistemas de múltiplas camadas captura características semânticas em distâncias longas na incorporação original. Experimentos em *benchmarks* padrão, como ModelNet40 e ShapeNetPart, demonstram a eficácia do DGCNN, alcançando uma acurácia de 92,9%.

### 3.2.4 KPConv

A KPConv (THOMAS *et al.*, 2019) revoluciona a convolução de pontos em nuvens de pontos, eliminando a necessidade de representações intermediárias. Com pesos de convolução localizados no espaço euclidiano através de pontos de kernel, sua flexibilidade permite a utilização de qualquer quantidade de pontos de kernel, superando limitações das convoluções fixas em grade. Além disso, os pontos de kernel podem ser ajustados pela rede, permitindo convoluções deformáveis que se adaptam à geometria local. Com uma estratégia de subamostragem regular, o KPConv demonstra eficiência e robustez a variações de densidade. Experimentos revelam que as redes KPConv superam abordagens de última geração em classificação e segmentação, alcançando uma acurácia de 92,9% no conjunto de dados ModelNet40. Adicionalmente, estudos de ablação e visualizações oferecem informações sobre o aprendizado do KPConv e validam sua eficácia descritiva.

### 3.2.5 PVCNN

A PVCNN (LIU *et al.*, 2019), foi desenvolvida como uma solução eficiente e rápida para aprendizado profundo em 3D. Enquanto abordagens anteriores utilizavam modelos de redes neurais baseadas em voxel ou pontos, a PVCNN supera essas limitações, resolvendo de forma eficaz os desafios computacionais associados a essas técnicas. Ao combinar representações em pontos e convoluções em voxels, a PVCNN reduz significativamente o consumo de memória, ao mesmo tempo em que melhora a localidade dos dados. Experimentos demonstraram que a PVCNN alcança uma precisão superior e maior velocidade em comparação com as abordagens anteriores, destacando-se com uma precisão de 92,4%, duas vezes maior que o modelo PointNet

em *benchmarks* de segmentação de partes e cenas. Além disso, a PVCNN mostrou-se eficaz na detecção de objetos 3D, superando modelos concorrentes em velocidade e eficiência de memória na GPU.

### 3.2.6 *CurveNet*

CurveNet (MUZAHID *et al.*, 2020) é uma abordagem inovadora que utiliza direções de curvatura principal de objetos 3D, obtidas de modelos CAD, como entrada para uma rede neural convolucional 3D (CNN 3D). Essas direções de curvatura proporcionam uma representação robusta das características geométricas, permitindo ao CurveNet extrair características discriminativas para o reconhecimento de objetos. Além disso, incorporamos o aprendizado multitarefa, utilizando a classificação de pose como uma tarefa auxiliar para melhorar a generalização do CurveNet. Resultados experimentais destacam a eficácia do CurveNet em comparação com abordagens convencionais, tanto com entrada de vetores de curvatura quanto com combinação de vetores de curvatura e voxels. Esses resultados promissores são validados em conjuntos de dados públicos, demonstrando a capacidade competitiva do CurveNet no reconhecimento de objetos 3D, com uma acurácia de classificação de 93,8%.

### 3.2.7 *DeltaConv*

A DeltaConv (WIERSMA *et al.*, 2022) é uma inovação que visa a construção de filtros anisotrópicos em nuvens de pontos, utilizando operadores geométricos do cálculo vetorial. Essa abordagem se destaca por sua capacidade de representar e processar informações direcionais de forma explícita, separando a rede em fluxos escalar e vetorial, conectados pelos operadores. Além disso, as convoluções resultantes são robustas, simples de implementar e demonstram desempenho competitivo ou superior em diversos *benchmarks*, contribuindo para a aceleração do treinamento e da inferência. A DeltaConv alcançou uma acurácia de 93,8% em tarefas de classificação.

## 3.3 **Redes com Abordagens Atentivas e *Transformers***

Esta seção descreve as redes que utilizam mecanismos de atenção e arquiteturas baseadas em *Transformers*, revolucionando o processamento e a análise de nuvens de pontos 3D. Tais abordagens permitem um foco seletivo em partes específicas dos dados, aprimorando a

capacidade de interpretação e inferência.

### 3.3.1 $PT^1$

A  $PT^1$  (ENGEL *et al.*, 2021) é uma abordagem inovadora para processamento de conjuntos de pontos desordenados e sem estrutura. Projetada para extrair características locais e globais, ela introduz um mecanismo de atenção local-global, visando capturar relações espaciais entre os pontos e informações sobre a forma. O SortNet é um componente chave que garante invariância à permutação dos dados de entrada, selecionando pontos com base em uma pontuação aprendida. A saída da  $PT^1$  é uma lista de características classificadas e invariáveis à permutação, adequada para integração em diversas aplicações de visão computacional. Experimentos demonstram resultados promissores, com uma acurácia de 92,8%, em comparação com abordagens anteriores em tarefas de classificação e segmentação de partes em conjuntos de dados convencionais.

### 3.3.2 $PT^2$

A  $PT^2$  (ZHAO *et al.*, 2021) é uma arquitetura de redes neurais autoatentivas desenvolvida para processar nuvens de pontos 3D. Inspirada pelo sucesso das redes autoatentivas em tarefas de processamento de linguagem natural e análise de imagens, nossa pesquisa explora sua aplicação específica em nuvens de pontos. Projetamos camadas de autoatenção adaptadas para nuvens de pontos e as empregamos na construção de redes autoatentivas para tarefas como segmentação semântica de cenas, segmentação de partes de objetos e classificação de objetos. A abordagem Point Transformer ( $PT^2$ ) aprimora significativamente o desempenho em comparação com métodos anteriores em várias áreas e tarefas. Por exemplo, no conjunto de dados desafiador S3DIS para segmentação semântica de cenas em grande escala, o ( $PT^2$ ) alcança um mIoU de 70,4% na Área 5, superando o modelo anterior mais forte em 3,3 pontos percentuais absolutos, e atingindo pela primeira vez a marca de 70% de mIoU, na classificação obteve uma acurácia total de 93,7% para o conjunto de dados ModelNet40.

### 3.3.3 $PCT$

A  $PCT$  (GUO *et al.*, 2021) é uma solução inovadora para aprendizado em nuvens de pontos. Inspirado no sucesso do *Transformer* em processamento de linguagem natural e suas

promissoras aplicações em processamento de imagens, a PCT foi desenvolvido para lidar com a complexidade das nuvens de pontos, que são irregulares e não ordenadas. Sua capacidade de lidar com permutações o torna adequado para trabalhar com sequências de pontos. Para capturar o contexto local nas nuvens de pontos, aprimoramos a incorporação de entrada com a ajuda de técnicas como amostragem dos pontos mais distantes e pesquisa dos vizinhos mais próximos. Experimentos detalhados mostram que a PCT alcança resultados de ponta em tarefas como classificação de formas, segmentação de partes, segmentação semântica e estimação de normais, obtendo na classificação uma acurácia de 93,2%.

### **3.3.4 PAConv**

A PAConv (XU *et al.*, 2021) representa uma abordagem inovadora para o processamento de nuvens de pontos 3D. Introduzindo uma operação genérica de convolução, a PAConv constrói dinamicamente o kernel de convolução utilizando matrizes de peso adaptativas aprendidas a partir das posições dos pontos. Essa técnica confere à PAConv uma flexibilidade excepcional para lidar com dados de nuvem de pontos irregulares e não ordenados, sem comprometer a eficiência computacional. Uma característica distintiva da PAConv é sua integração eficiente em pipelines de processamento baseados em MLP, sem exigir modificações nas configurações da rede. Experimentos demonstraram que a PAConv alcança resultados competitivos em tarefas de classificação e segmentação, superando modelos de referência enquanto mantém uma eficiência computacional notável. Além disso, obteve uma acurácia de 93,6% em tarefas de classificação, destacando sua eficácia em reconhecer padrões e estruturas complexas nos dados de entrada. Estudos aprofundados de ablação e visualizações complementam a compreensão da metodologia da PAConv.

## **3.4 Redes com Estratégias de Amostragem e Robustez**

Esta seção discute redes que implementam estratégias de amostragem e técnicas de robustez. Essas metodologias são fundamentais para melhorar a análise de nuvens de pontos em ambientes desafiadores, garantindo maior resistência a ruídos e variabilidades nos dados.

### 3.4.1 PointASNL

PointASNL (YAN *et al.*, 2020) é uma rede inovadora para processar nuvens de pontos robustamente, especialmente em cenários com ruído, alcançando uma acurácia de 93,2%. Seu destaque está no módulo de amostragem adaptativa (AS), que reajusta os pesos dos vizinhos dos pontos amostrados inicialmente pelo método de amostragem do ponto mais distante (FPS). Esse ajuste dinâmico permite lidar eficazmente com *outliers* e ruídos nos dados de entrada. Além disso, a PointASNL utiliza um módulo local-não local (L-NL) para capturar as dependências locais e não locais dos pontos amostrados, garantindo robustez mesmo em ambientes com ruído. Experimentos mostram que a PointASNL supera métodos anteriores em tarefas de classificação e segmentação em vários conjuntos de dados, incluindo o desafiador conjunto de dados real SemanticKITTI, com notável presença de ruído.

### 3.4.2 PRA-Net

A PRA-Net (CHENG *et al.*, 2021) é uma estrutura inovadora para análise de nuvens de pontos, destacando-se pela integração de estratégias eficazes para fortalecer a representação de características. Composta por um módulo de Aprendizado de Estrutura Intra-região (ISL) e um módulo de Aprendizado de Relações Inter-região (IRL), a PRA-Net combina aprendizado de contextos intra-região e relações inter-região de forma sinérgica. O módulo ISL integra dinamicamente informações estruturais locais às características dos pontos, enquanto o módulo IRL captura relações inter-região de maneira adaptativa e eficiente por meio de um esquema de partição de região diferenciável e uma estratégia baseada em pontos representativos. Experimentos extensivos em diversos conjuntos de dados 3D, abrangendo classificação de formas, estimativa de pontos-chave e segmentação de partes, validaram a eficácia e a capacidade de generalização da PRA-Net, alcançando uma acurácia de 93,2%.

### 3.4.3 APES

A APES (WU *et al.*, 2023) é uma abordagem inovadora de amostragem para nuvens de pontos. Enquanto métodos convencionais ainda recorrem a técnicas de amostragem aleatória ou amostragem do ponto mais distante, a APES adota uma abordagem orientada à tarefa, inspirada no algoritmo de detecção de bordas de Canny para imagens. Por meio do mecanismo de atenção, a APES seleciona pontos salientes no contorno da nuvem de pontos, oferecendo

uma alternativa mais eficaz para a amostragem em tarefas de reconhecimento de objetos ou segmentação. Experimentos qualitativos e quantitativos evidenciam o desempenho superior da APES em relação aos métodos convencionais, solidificando seu papel como uma contribuição significativa para o campo de processamento de nuvens de pontos. Além disso, a APES alcançou uma acurácia geral nas tarefas de classificação de 93,8%.

## 4 METODOLOGIA

Nessa seção é descrita a metodologia utilizada na pesquisa, seguindo uma abordagem sistemática para avaliar e aprimorar redes neurais convolucionais (CNNs) para a tarefa de classificação de objetos a partir de nuvem de pontos. A Figura 6 serve como guia para o processo, que é dividido em cinco etapas distintas.

Figura 6 – Fluxo Metodológico para Proposta da Rede APES-Soft.



Fonte: Elaborada pelo autor.

1. Execução da Rede APES: A implementação da Rede APES representa o ponto de partida fundamental deste estudo, sendo especialmente treinada com o conjunto de dados ModelNet40 (WU *et al.*, 2015) (detalhado na seção 4.1), com o propósito de classificar objetos 3D em nuvens de pontos.
2. Análise de Ablação: A análise de ablação é conduzida para investigar o impacto de diferentes componentes da rede APES. Cenários são realizados, nos quais camadas convolucionais, camadas lineares, são removidas e modificadas. Esta etapa visa entender a contribuição relativa de cada componente para o desempenho geral da rede.
3. Avaliação dos Resultados: Na terceira etapa, os resultados obtidos são avaliados utilizando métricas de desempenho padrão para problemas de classificação. Métricas como Acurácia, Precisão, Sensibilidade, F1-Score, Coeficiente de Correlação de Matthews (MCC), Índice Jaccard, e Similaridade por Cosseno, para medir o desempenho da rede APES e suas variações resultantes da análise de ablação.
4. Testes Estatísticos: Testes estatísticos são realizados para validar as diferenças de desempenho observadas entre as diferentes configurações da rede. Utilizando testes de hipóteses apropriados, como ANOVA, Tukey's HSD, Teste de Kruskal-Wallis, e Teste de Friedman, verifica-se se as melhorias de desempenho são estatisticamente significativas.
5. Proposta da Rede APES-Soft: Por fim, com base nas percepções obtidas da análise de

ablação e avaliação dos resultados, propõe-se uma versão aprimorada da rede APES, denominada APES-Soft. Contendo as melhores modificações introduzidas na arquitetura, com o objetivo de otimizar seu desempenho para classificação de objetos em nuvem de pontos.

#### 4.1 Conjunto de dados

O conjunto de dados ModelNet40 (WU *et al.*, 2015) é uma compilação abrangente de 12.311 modelos 3D CAD distribuídos em 40 categorias de objetos comuns. Para assegurar uma análise imparcial, adotamos a divisão oficial entre conjuntos de treinamento e teste, onde 9.843 modelos são dedicados ao treinamento e 2.468 para teste.

Cada modelo é processado transformando sua superfície de malha em uma nuvem de pontos, os quais são selecionados uniformemente e normalizados para se encontrarem dentro de uma esfera unitária. Para simplificar o processamento, utilizamos apenas as coordenadas 3D como entrada. Além disso, para enriquecer a diversidade dos dados, aplicamos transformações aleatórias, como escalonamento, rotação e deslocamento, a cada nuvem de pontos no espaço 3D.

É importante ressaltar que durante os testes, assim como Wu *et al.* (2023) não foi utilizada nenhuma técnicas de aumento de dados, que são habitualmente utilizadas quando o conjunto de dados é limitado (ZHU *et al.*, 2024), ou métodos de votação, que servem para estimar a transformação rígida (rotação e tradução) em nuvens de pontos 3D (YUAN *et al.*, 2024).

#### 4.2 Configurações

Para garantir consistência, todas as configurações e o ambiente de desenvolvimento foram mantidos constantes. A rede APES foi reexecutada sem alterações, utilizando o mesmo conjunto de dados e as duas modalidades de *downsampling*<sup>1</sup> propostas: global e local.

O modelo foi treinado com base nas configurações detalhadas na Tabela 2. Utilizamos o otimizador AdamW com uma taxa de aprendizado inicial de  $1 \times 10^{-4}$ , a qual é reduzida progressivamente até  $1 \times 10^{-8}$  utilizando um agendamento de decaimento cossenoidal. O hiperparâmetro de decaimento de peso foi ajustado para 1. Adicionalmente, aplicamos *dropout* com uma probabilidade de 0,5. O treinamento foi conduzido com um tamanho de lote de 8 exemplos por iteração, ao longo de 200 épocas.

<sup>1</sup> *Downsampling* em nuvem de pontos é o processo de reduzir o número de pontos enquanto mantém a estrutura principal. Isso facilita o processamento e mantém a integridade da nuvem original.

Tabela 2 – Configurações de Treinamento do Modelo e Ambiente Computacional.

Configurações de Treinamento do Modelo	Especificações do Ambiente Computacional
Otimizador AdamW	Sistema Operacional: Linux
Taxa de Aprendizado Inicial: $1 \times 10^{-4}$	Python: 3.9.19, compilado em março de 2024
Agendamento de Decaimento Cossenoidal	Compilador: GCC 11.2.0
Decaimento de Peso: 1	<i>Hardware:</i>
<i>Dropout:</i> 0.5	- CPU: Intel Core i7-7700 (7ª Geração)
Tamanho do Lote ( <i>Batch Size</i> ): 8	- GPU: NVIDIA GeForce GTX 1070
Número de Épocas: 200	- CUDA: Versão 10.0.13, Runtime CUDA: 11.7
	<i>Bibliotecas:</i>
	- OpenCV: Versão 4.9.0
	- MMEngine: Versão 0.7.2

Fonte: Elaborada pelo autor.

O ambiente computacional utilizado para o treinamento foi configurado da seguinte forma: o sistema operacional base foi Linux, com Python na versão 3.9.19, compilado em março de 2024. Utilizamos o GCC 11.2.0 como compilador. O hardware incluiu uma CPU Intel Core i7-7700 de 7ª geração e uma GPU NVIDIA GeForce GTX 1070, aproveitando as capacidades do CUDA na versão 10.0.13 com Runtime CUDA 11.7. Além disso, foram utilizadas as bibliotecas OpenCV na versão 4.9.0 e MMEngine na versão 0.7.2 para suporte adicional durante o treinamento do modelo.

### 4.3 Arquitetura da Rede APES

A arquitetura da rede Attention-Based Point Cloud Edge Sampling (APES) é dividida em duas partes fundamentais: a Rede Neural Convolutacional (CNN) e a Perceptron Multicamadas (MLP), como é possível visualizar na Figura 7.

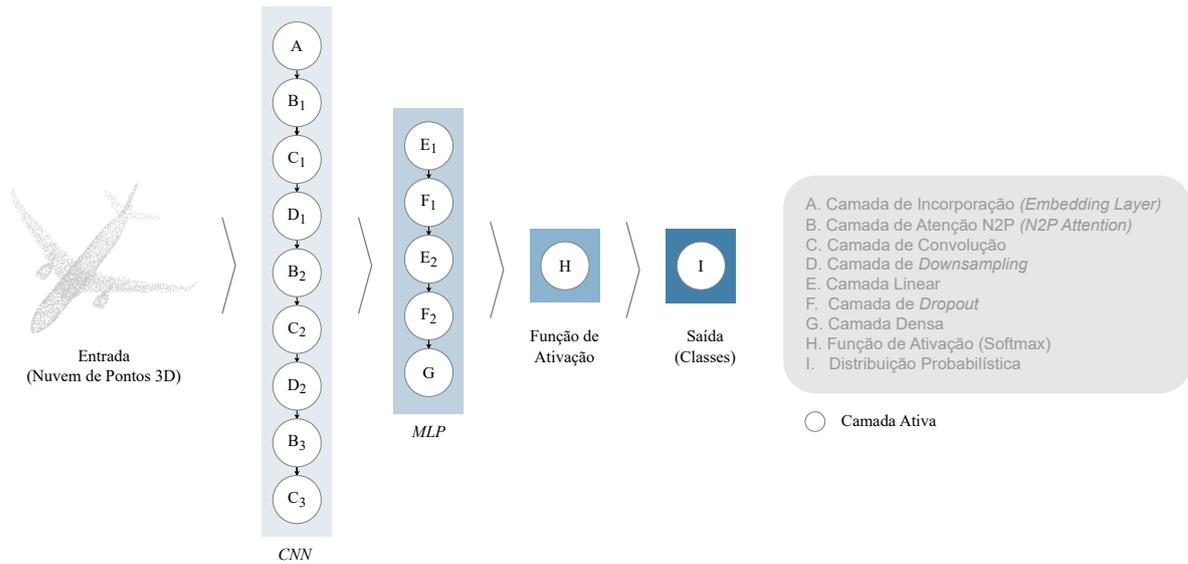
#### 4.3.1 Rede Neural Convolutacional (CNN)

A CNN é encarregada de processar as coordenadas tridimensionais de entrada e de extrair características relevantes da nuvem de pontos. Este componente consiste nas seguintes camadas:

##### 4.3.1.1 Camadas Convolutacionais

As camadas convolutacionais desempenham um papel fundamental na extração de características da nuvem de pontos, sendo capazes de identificar padrões locais e globais nos dados. Isso permite que a rede neural aprenda representações hierárquicas complexas. A seguir,

Figura 7 – Arquitetura da Rede APES.



Fonte: Elaborada pelo autor.

teremos uma melhor compreensão sobre as camadas presentes nesta arquitetura.

#### 4.3.1.1.1 Camada de Incorporação (*Embedding Layer*)

Esta camada transforma as coordenadas 3D de entrada em um espaço de características de dimensão superior. Inicialmente, os pontos são agrupados utilizando um esquema específico, resultando em uma expansão do número de canais de 3 para 6. Em seguida, várias operações de convolução são aplicadas para extrair e combinar características significativas.

A camada inclui múltiplas convoluções sequenciais, cada uma seguida por uma normalização por lotes e ativação de *LeakyReLU*. Essas operações aumentam progressivamente a complexidade e a representatividade das características. Posteriormente, os pontos são novamente agrupados e o processo de convolução é repetido para capturar informações em diferentes escalas e níveis de detalhe.

Por fim, os resultados das operações de convolução são combinados e retornados como um único tensor de características de dimensões  $(B, 128, N)$ , onde  $B$  representa o tamanho do lote, 128 é a dimensão do espaço de características, isso significa que cada ponto de entrada é representado por um vetor de 128 elementos, capturando uma representação densa e rica dos dados, e  $N$  é o número de pontos, no caso 2048. Essas características representam uma representação dos pontos de entrada, pronta para serem utilizadas em camadas subsequentes, conforme discutido por Wu *et al.* (2023).

#### 4.3.1.1.2 Camada de Atenção N2P

Esta camada de atenção opera sobre as características extraídas pela camada de incorporação, destacando padrões cruciais na nuvem de pontos, que calcula mapas de correlação entre cada ponto e seus vizinhos. Para isso, ela divide os dados em quatro partes. Cada parte executa sua própria operação de atenção em paralelo, o que permite uma exploração mais ampla dos padrões presentes nos dados.

Internamente, a camada realiza uma série de operações, incluindo convoluções separadas para as consultas ( $q$ ), chaves ( $k$ ) e valores ( $v$ ), seguidas pelo cálculo da atenção entre os pontos. A atenção resultante é então aplicada aos valores, gerando uma representação ponderada dos dados que destaca as características mais importantes. Além disso, a camada inclui um mecanismo de *feedforward*, composto por duas camadas convolucionais seguidas de normalizações por lotes, que ajudam a aprimorar ainda mais a representação dos dados.

Por fim, os resultados da atenção e do *feedforward* são combinados com os dados de entrada original, através de conexões residuais seguidas de normalizações por lotes, permitindo que o modelo integre informações relevantes das diferentes etapas do processamento.

#### 4.3.1.1.3 Camada de Convolução

Após cada etapa de *downsampling*, ocorre uma convolução 1D para transformar as características dimensionais. São usadas três camadas de convolução, com tamanhos de entrada e saída de 128 e 1024, respectivamente. Já o do *kernel* é 1, o que significa que cada operação de convolução é aplicada a apenas uma característica de entrada em cada passo. Essencialmente, isso significa que a convolução é aplicada independentemente em cada uma das 128 características de entrada, produzindo 1024 características de saída.

#### 4.3.1.2 Camadas de Pooling

As operações de *Max Pooling* são usadas em redes neurais convolucionais para reduzir a dimensionalidade dos dados. Consiste em subdividir a entrada em regiões e selecionar o valor máximo de cada região para formar a saída. Isso ajuda a preservar as características mais proeminentes da entrada, enquanto reduz sua dimensionalidade. É especialmente eficaz na preservação de características importantes, como bordas ou padrões.

### 4.3.1.3 *Camadas de Downsampling*

Existem duas opções de camadas de *downsampling* disponíveis: global e local. Essas camadas reduzem progressivamente o número de pontos na nuvem de pontos, primeiro pela metade e depois novamente pela metade.

#### 4.3.1.3.1 Global

Utiliza convoluções 1D para realizar o *downsampling*. Ela opera sobre a representação compacta da nuvem de pontos, transformando-a de uma dimensão de entrada para uma dimensão reduzida. Isso é feito por meio de convoluções separadas para as consultas ( $q$ ), chaves ( $k$ ) e valores ( $v$ ), seguidas por uma operação de atenção e seleção dos pontos mais importantes. Os pontos selecionados são então reorganizados para formar a saída da camada de *downsampling* global.

#### 4.3.1.3.2 Local

Realiza o *downsampling* agrupando os pontos próximos uns dos outros para formar grupos, e em seguida, aplicando convoluções 2D para processar esses grupos. A atenção é então calculada entre os pontos dentro de cada grupo, e os pontos mais importantes são selecionados. A saída é obtida combinando os valores correspondentes aos pontos selecionados, resultando em uma redução significativa no número de pontos na nuvem de pontos.

### 4.3.1.4 *Processo de Extração de Características*

O Algoritmo 1 representa um fluxo de dados na extração de características, onde os dados de entrada passam por várias camadas da arquitetura, para extrair características relevantes antes de serem utilizados na tarefa de classificação.

Vamos entender cada parte deste algoritmo utilizado para extrair as características das nuvem de pontos:

#### 4.3.1.4.1 Camada de Incorporação ( $A$ )

Esta camada inicial transforma o tensor de entrada  $X$  em um novo tensor  $X'$  por meio de uma camada de incorporação. Aqui, cada ponto de dado, representado por um vetor de

---

**Algoritmo 1:** Fluxo de dados na extração de características da Rede APES
 

---

- 1: **Entrada:** Tensores  $X \in \mathbb{R}^{B \times 3 \times N}$ , onde  $B$  é o tamanho do lote e  $N$  é o número de pontos.
  - 2: **Saída:** Tensores  $X_{\text{cnn}} \in \mathbb{R}^{B \times K}$ , onde  $K$  é o tamanho do vetor de características.
  - 3:
  - 4: **Início**
  - 5:   **Camada de Incorporação:**  $X' = A(X)$ , onde  $A : \mathbb{R}^{B \times 3 \times N} \rightarrow \mathbb{R}^{B \times 128 \times N}$
  - 6:   **Camada de Atenção:**  $X'' = B_1(X')$ , onde  $B_1 : \mathbb{R}^{B \times 128 \times N} \rightarrow \mathbb{R}^{B \times 128 \times N}$
  - 7:   **Camada de Convolução:**  $X''' = C_1(X'')$ , onde  $C_1 : \mathbb{R}^{B \times 128 \times N} \rightarrow \mathbb{R}^{B \times 1024 \times N}$
  - 8:   **Operação de Max Pooling e Armazenamento:**  $R_1 = \max(X''', \text{dim} = -1)$
  - 9:   **Camada de Downsampling:**  $X'''' = D_1(X''', \text{tipo} = \text{global ou local}, 1024 \text{ pontos})$
  - 10:   **Camada de Atenção:**  $X_2'''' = B_2(X''''')$
  - 11:   **Camada de Convolução:**  $X_3'''' = C_2(X_2''''')$
  - 12:   **Operação de Max Pooling e Armazenamento:**  $R_2 = \max(X_3''''', \text{dim} = -1)$
  - 13:   **Camada de Downsampling:**  $X_4'''' = D_2(X_3''''', \text{tipo} = \text{global ou local}, 512 \text{ pontos})$
  - 14:   **Camada de Atenção:**  $X_5'''' = B_3(X_4''''')$
  - 15:   **Camada de Convolução:**  $X_6'''' = C_3(X_5''''')$
  - 16:   **Operação de Max Pooling e Armazenamento:**  $R_3 = \max(X_6''''', \text{dim} = -1)$
  - 17:   **Empacotamento dos Resultados:**  $X_{\text{cnn}} = \text{pack}([R_1, R_2, R_3], \text{dim} = 1)$
  - 18: **Fim**
- 

tamanho 3, é transformado em um vetor de características de tamanho 128. Matematicamente, podemos expressar isso como:

$$X' = A(X) \tag{4.1}$$

#### 4.3.1.4.2 Camada de Atenção ( $B_1, B_2, B_3$ )

As camadas de atenção aplicam mecanismos de atenção ao tensor de entrada. No contexto deste algoritmo, as camadas de atenção são aplicadas após a camada de incorporação e após cada camada de *downsampling*. Elas podem modificar o tensor de entrada, realçando partes importantes ou suprimindo informações irrelevantes. Se representarmos essas camadas de atenção como  $B_i$ , onde  $i$  varia de 1 a 3, podemos descrever sua operação como:

$$X'' = B_i(X') \tag{4.2}$$

#### 4.3.1.4.3 Camada de Convolução ( $C_1, C_2, C_3$ )

As camadas de convolução aplicam operações de convolução ao tensor de entrada. Elas podem extrair características importantes dos dados, como bordas, padrões ou texturas. Cada camada de convolução neste algoritmo aumenta a dimensionalidade dos dados. Representando essas camadas de convolução como  $C_i$ , onde  $i$  varia de 1 a 3, temos:

$$X''' = C_i(X'') \quad (4.3)$$

#### 4.3.1.4.4 Operação de Max Pooling e Armazenamento

Após cada camada de convolução, uma operação de pooling é aplicada, seguida por uma operação de maximização. O max pooling é uma técnica de *downsampling* que reduz a dimensionalidade dos dados, mantendo apenas as características mais importantes. A dimensão especificada (*dim*) indica a dimensão ao longo da qual a operação de pooling é aplicada.

#### 4.3.1.4.5 Camada de *Downsampling* ( $D_1, D_2$ )

Estas camadas são cruciais para reduzir a dimensionalidade dos dados. No algoritmo em questão, duas camadas de *downsampling* são empregadas, cada uma com uma estratégia diferente:  $t$  pode ser global ou local, e um número específico de pontos resultantes  $n$  pode ser selecionado (512 ou 1024 pontos). Se representarmos essas camadas como  $D_i$ , onde  $i$  varia de 1 a 2, a operação pode ser expressa de forma geral como:

$$X'''' = D_i(X''', t, n) \quad (4.4)$$

#### 4.3.1.4.6 Empacotamento dos Resultados

Finalmente, os resultados das operações anteriores são empacotados em um único tensor  $X_{\text{cnn}}$ , concatenando os tensores  $R_1, R_2, R_3$  ao longo da dimensão especificada (*dim*).

### 4.3.2 *Perceptron Multicamadas (MLP)*

A MLP, por sua vez, recebe a representação compacta da nuvem de pontos da CNN e realiza a etapa de classificação, composta pelas seguintes camadas:

#### 4.3.2.1 *Camadas Lineares*

A representação compacta da nuvem de pontos é processada por meio de uma sequência de camadas lineares, intercaladas com operações de normalização por lotes e ativações de *LeakyReLU*. Esse processo é essencial para transformar a informação espacialmente distribuída da entrada em representações mais compactas e significativas para a tarefa de classificação.

#### 4.3.2.1.1 Primeira Camada Linear

A primeira camada recebe uma entrada de dimensão 3072 e produz uma saída de dimensão 1024. Em seguida, é aplicada uma camada de normalização por lotes, que normaliza os valores ao longo do lote de entrada, seguida de uma ativação *LeakyReLU*. A normalização por lotes ajuda a estabilizar e acelerar o treinamento, enquanto a ativação *LeakyReLU* introduz não-linearidade na rede, permitindo que ela aprenda relações mais complexas nos dados.

#### 4.3.2.1.2 Segunda Camada Linear

A segunda camada linear recebe a saída da primeira camada, que tem dimensão 1024, e produz uma saída de dimensão 256. Novamente, essa saída é seguida por uma camada de normalização por lotes e uma ativação *LeakyReLU*. Essa etapa de redução progressiva da dimensionalidade visa extrair características cada vez mais abstratas e informativas da representação compacta, preparando-a para a tarefa final de classificação.

#### 4.3.2.1.3 Camada Densa

É a camada linear de saída, recebe uma entrada de dimensão 256 e produz a saída da rede, um vetor de tamanho 40 que representa as classes de destino para a classificação.

Essas camadas lineares e de normalização por lotes formam o núcleo do processo de transformação de dados no cabeçalho da rede neural, capacitando-a a aprender representações discriminativas que podem ser utilizadas para classificar as nuvens de pontos em suas respectivas categorias.

#### 4.3.2.2 Regularização

Operações de *Dropout* é uma técnica de regularização usada para evitar o sobreajuste, do inglês (*overfitting*), durante o treinamento de redes neurais. Ela funciona desativando aleatoriamente um percentual de unidades (neurônios) em cada passo de treinamento. No caso deste modelo, duas camadas de *dropout* são aplicadas com uma taxa de 0.5 cada. Especificamente, um dropout de 0.5 é aplicado após a primeira camada linear, e outro após a segunda camada linear. Isso significa que, durante o treinamento, metade das unidades em cada camada é desativada aleatoriamente, ajudando a rede a aprender características mais robustas e evitando que os neurônios se tornem muito dependentes uns dos outros. Essa abordagem promove uma melhor

generalização do modelo, tornando-o mais capaz de lidar com dados de teste ou novos dados que não foram vistos durante o treinamento.

#### 4.3.2.3 Processo de Classificação

O Algoritmo 2 representa um fluxo de dados na classificação, onde os dados de entrada são processados por várias camadas da arquitetura, para realizar a tarefa de classificação.

---

#### Algoritmo 2: Fluxo de dados na classificação da Rede APES

---

- 1: **Entrada:** Tensores  $X_{\text{cnn}} \in \mathbb{R}^{B \times K}$ , onde  $K$  é o tamanho do vetor de características.
  - 2: **Saída:** Classe  $Y \in \mathbb{R}^{B \times C}$ , onde  $C$  é o número de classes.
  - 3:
  - 4: **Início**
  - 5: **Primeira Camada Linear:**  $X' = E_1(X_{\text{cnn}})$
  - 6: **Primeiro Dropout:**  $X'' = F_1(X', p = 0.5)$
  - 7: **Segunda Camada Linear:**  $X''' = E_2(X'')$
  - 8: **Segundo Dropout:**  $X'''' = F_2(X''', p = 0.5)$
  - 9: **Camada Densa:**  $Y = G(X'''' )$
  - 10: **Fim**
- 

##### 4.3.2.3.1 Primeira Camada Linear ( $E_1$ )

Esta operação transforma o tensor de entrada  $X_{\text{cnn}}$  em um novo tensor  $X'$  através de uma transformação linear. Isso é feito multiplicando  $X_{\text{cnn}}$  por uma matriz de pesos  $W_1$  e adicionando um vetor de vies  $b_1$ :

$$X' = E_1(X_{\text{cnn}}) = X_{\text{cnn}} \times W_1 + b_1 \quad (4.5)$$

##### 4.3.2.3.2 Primeiro Dropout ( $F_1$ )

O *dropout* é uma técnica de regularização amplamente utilizada em redes neurais para evitar o *overfitting*. Durante o treinamento, o *dropout* desliga aleatoriamente uma fração das unidades na camada de forma que essas unidades não participam do cálculo da saída da rede. Esse processo é controlado por uma taxa de *dropout*, denotada por  $p$ , que representa a probabilidade de uma unidade ser desligada.

Matematicamente, o *dropout* pode ser descrito como a multiplicação do tensor de saída  $X'$  por uma máscara binária  $M_1$ . Cada elemento da máscara  $M_1$  é 1 com probabilidade

$1 - p$  e 0 com probabilidade  $p$ , onde  $p$  é a taxa de *dropout*. Assim, para uma taxa de *dropout* de  $p = 0.5$ , a fórmula matemática que descreve essa operação é:

$$X'' = F_1(X', p = 0.5) = X' \odot M_1 \quad (4.6)$$

Aqui,  $X''$  é o tensor de saída após a aplicação do *dropout*,  $X'$  é o tensor de saída antes do *dropout*, e  $\odot$  representa a multiplicação elemento a elemento entre  $X'$  e  $M_1$ . A máscara  $M_1$  é gerada de forma que a cada elemento de  $X'$  é multiplicado por 1 com probabilidade  $1 - p$  ou por 0 com probabilidade  $p$ , efetivamente desligando aleatoriamente uma fração de unidades na camada durante o treinamento.

#### 4.3.2.3.3 Segunda Camada Linear ( $E_2$ )

Esta operação é semelhante à primeira camada linear, transformando o tensor de entrada  $X''$  em um novo tensor  $X'''$  usando uma matriz de pesos  $W_2$  e um vetor de viés  $b_2$ :

$$X''' = E_2(X'') = X'' \times W_2 + b_2 \quad (4.7)$$

#### 4.3.2.3.4 Segundo Dropout ( $F_2$ )

O *dropout* é novamente aplicado ao tensor de saída  $X'''$ , gerando  $X''''$ :

$$X'''' = F_2(X''', p = 0.5) = X''' \odot M_2 \quad (4.8)$$

O operador  $\odot$  denota a multiplicação elemento a elemento entre  $X'''$  e a máscara binária  $M_2$ , onde  $M_2$  contém zeros e uns, desativando aleatoriamente 50% dos elementos de  $X'''$  para prevenir *overfitting*.

#### 4.3.2.3.5 Camada Densa ( $G$ )

Esta é a camada final do modelo, que transforma o tensor resultante  $X''''$  em um tensor de saída  $Y$  representando as classes preditas. Isso é feito usando uma última matriz de pesos  $W_3$  e um vetor de viés  $b_3$ :

$$Y = G(X''') = X'''' \times W_3 + b_3 \quad (4.9)$$

Essas operações formam o fluxo de dados na classificação, onde cada camada linear é seguida por um *dropout* para regularização, e a última camada produz as predições finais.

## 4.4 Análise de ablação

Nesta análise de ablação, realizamos três cenários com o propósito de avaliar o impacto de diferentes elementos da arquitetura. No Cenário I, examinamos o efeito da camada de *downsampling*. Nos Cenários II e III, progressivamente removemos as camadas de convolução, bem como uma camada linear e, conseqüentemente, uma aplicação de *dropout*. Em seguida, ajustamos a primeira camada linear para que seja compatível com cada redução do vetor de características.

Na configuração original da arquitetura, é extraído um vetor de características final com dimensão 3072. Inspirados pelo estudo de ablação conduzido por Wu *et al.* (2023), onde o número total de pontos é reduzido incrementalmente, adotamos uma abordagem similar. Se denotarmos o número total de pontos como  $N = 3072$ , no Cenário II, obtivemos um vetor de características com dimensão  $N/2 = 1536$ . No Cenário III, utilizamos  $N/3$ , resultando em um vetor de características de dimensão 1024. Esta metodologia nos permite avaliar como a diminuição gradual das dimensões afeta o desempenho da rede neural em tarefas específicas de classificação.

A seguir, apresentaremos uma descrição detalhada de cada cenário, destacando como essas modificações afetam a arquitetura da rede neural e as implicações para tarefas específicas de classificação.

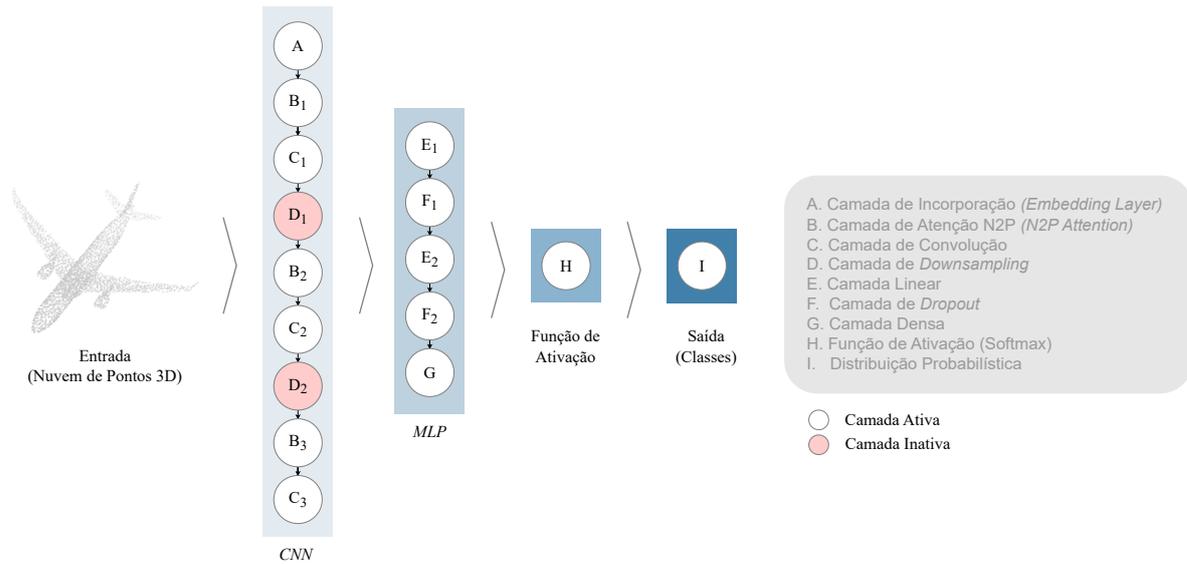
### 4.4.1 Cenário I

O Cenário I segue a mesma estrutura descrita no Processo de Extração de Características 4.3.1.4 da rede APES, com a diferença de que não utiliza as camadas de *downsampling*, e exatamente o mesmo Processo de Classificação 4.3.2.3, como pode ser visto na Figura 8. Abaixo, apresentamos as modificações realizadas no cenário:

O Algoritmo 3 (Fluxo de dados na extração de características do Cenário I) representa um fluxo de dados na extração de características, onde os dados de entrada passam por várias camadas da arquitetura, para extrair características relevantes antes de serem utilizados na tarefa de classificação.

Vamos entender cada parte deste algoritmo utilizado para a extração de características:

Figura 8 – Arquitetura do Cenário I.



Fonte: Elaborada pelo autor.

---

### Algoritmo 3: Fluxo de dados na extração de características do Cenário I

---

- 1: **Entrada:** Tensores  $X \in \mathbb{R}^{B \times 3 \times N}$ , onde  $B$  é o tamanho do lote e  $N$  é o número de pontos.
  - 2: **Saída:** Tensores  $X_{\text{cnn}} \in \mathbb{R}^{B \times K}$ , onde  $K$  é o tamanho do vetor de características.
  - 3:
  - 4: **Início**
  - 5: **Camada de Incorporação:**  $X' = A(X)$ , onde  $A : \mathbb{R}^{B \times 3 \times N} \rightarrow \mathbb{R}^{B \times 128 \times N}$
  - 6: **Camada de Atenção:**  $X'' = B_1(X')$ , onde  $B_1 : \mathbb{R}^{B \times 128 \times N} \rightarrow \mathbb{R}^{B \times 128 \times N}$
  - 7: **Camada de Convolução:**  $X''' = C_1(X'')$ , onde  $C_1 : \mathbb{R}^{B \times 128 \times N} \rightarrow \mathbb{R}^{B \times 1024 \times N}$
  - 8: **Operação de Max Pooling e Armazenamento:**  $R_1 = \max(X''', \text{dim} = -1)$
  - 9: **Camada de Atenção:**  $X_2'''' = B_2(X''')$
  - 10: **Camada de Convolução:**  $X_3'''' = C_2(X_2''')$
  - 11: **Operação de Max Pooling e Armazenamento:**  $R_2 = \max(X_3''', \text{dim} = -1)$
  - 12: **Camada de Atenção:**  $X_5'''' = B_3(X_4''')$
  - 13: **Camada de Convolução:**  $X_6'''' = C_3(X_5''')$
  - 14: **Operação de Max Pooling e Armazenamento:**  $R_3 = \max(X_6''', \text{dim} = -1)$
  - 15: **Empacotamento dos Resultados:**  $X_{\text{cnn}} = \text{pack}([R_1, R_2, R_3], \text{dim} = 1)$
  - 16: **Fim**
- 

#### 4.4.1.1 Camada de Incorporação (A)

Esta camada transforma o tensor de entrada  $X$  em um novo tensor  $X'$  através de uma camada de incorporação. Neste caso, a camada de incorporação aumenta a dimensionalidade dos dados, transformando cada ponto de dados (representado por um vetor de tamanho 3) em um vetor de características de tamanho 128.

#### 4.4.1.2 Camada de Atenção ( $B_1, B_2, B_3$ )

As camadas de atenção são responsáveis por aplicar mecanismos de atenção ao tensor de entrada. No caso deste algoritmo, as camadas de atenção são aplicadas após a camada de incorporação. Elas podem modificar o tensor de entrada, destacando partes importantes ou suprimindo informações irrelevantes.

#### 4.4.1.3 Camada de Convolução ( $C_1, C_2, C_3$ )

As camadas de convolução aplicam operações de convolução ao tensor de entrada. Elas podem extrair características importantes dos dados, como bordas, padrões ou texturas. Cada camada de convolução neste algoritmo aumenta a dimensionalidade dos dados.

#### 4.4.1.4 Operação de Max Pooling e Armazenamento

Esta operação de *pooling*, seguida por uma operação de maximização, é aplicada após cada camada de convolução. O *max pooling* é uma técnica de *downsampling* que reduz a dimensionalidade dos dados, mantendo apenas as características mais importantes. A dimensão especificada (*dim*) indica a dimensão ao longo da qual a operação de *pooling* é aplicada.

#### 4.4.1.5 Empacotamento dos Resultados

Finalmente, os resultados das operações anteriores são empacotados em um único tensor  $X_{\text{cnn}}$ , concatenando os tensores  $R_1, R_2, R_3$  ao longo da dimensão especificada (*dim*).

### 4.4.2 Cenário II

Na Figura 9, é possível visualizar as modificações realizadas na arquitetura, onde partimos do Cenário I 4.4.1, ou seja, permanecendo sem as camadas de *downsampling*, e realizando modificações nas camadas convolucionais e nas camadas lineares. Abaixo é possível verificar as modificações:

O Algoritmo 4 (Fluxo de dados na extração de características do Cenário II) representa um fluxo de dados na extração de características, onde os dados de entrada passam por várias camadas da arquitetura, para extrair características relevantes antes de serem utilizados na tarefa de classificação.



tensor de entrada, destacando partes importantes ou suprimindo informações irrelevantes.

#### 4.4.2.3 Camada de Convolução ( $C_1$ )

As camadas de convolução aplicam operações de convolução ao tensor de entrada. Elas podem extrair características importantes dos dados, como bordas, padrões ou texturas. Aqui, a camada de convolução aumenta a dimensionalidade dos dados.

#### 4.4.2.4 Divisão do Tensor

Após a camada de convolução, o tensor resultante  $X'''$  é dividido em três partes iguais ao longo da dimensão especificada (-1, que é a dimensão de características). Isso é feito pela operação de divisão, resultando em três tensores  $R_1$ ,  $R_2$  e  $R_3$ , que representam as características extraídas de diferentes partes dos dados de entrada, como diferentes regiões espaciais ou partes do objeto representado. Matematicamente, podemos expressar isso como:

$$R_1, R_2, R_3 = \text{split}(X''', \text{número de seções} = 3, \text{dim} = -1) \quad (4.10)$$

#### 4.4.2.5 Empacotamento dos Resultados

Após a divisão, os resultados das operações anteriores são empacotados em um único tensor  $X_{\text{cnn}}$ , concatenando os tensores  $R_1$ ,  $R_2$  e  $R_3$  ao longo da dimensão especificada (dim). Matematicamente, essa operação de empacotamento pode ser representada como:

$$X_{\text{cnn}} = \text{pack}([R_1, R_2, R_3], \text{dim} = 1) \quad (4.11)$$

Essas operações permitem extrair e combinar características relevantes de diferentes regiões dos dados de entrada, facilitando a representação eficaz das informações para a tarefa de classificação.

Já o Algoritmo 5 (Fluxo de dados na classificação do Cenário II) representa um fluxo de dados na classificação, onde os dados de entrada são processados por camadas da arquitetura, para realizar a tarefa de classificação.

#### 4.4.2.6 Primeira Camada Linear ( $E_1$ )

Esta operação transforma o tensor de entrada  $X_{\text{cnn}}$  em um novo tensor  $X'$  através de uma transformação linear. Isso é feito multiplicando  $X_{\text{cnn}}$  por uma matriz de pesos  $W_1$  e

---

**Algoritmo 5:** Fluxo de dados na classificação do Cenário II
 

---

- 1: **Entrada:** Tensores  $X_{\text{cnn}} \in \mathbb{R}^{B \times K}$ , onde  $K$  é o tamanho do vetor de características.
  - 2: **Saída:** Classe  $Y \in \mathbb{R}^{B \times C}$ , onde  $C$  é o número de classes.
  - 3:
  - 4: **Início**
  - 5:   **Primeira Camada Linear:**  $X' = E_1(X_{\text{cnn}})$
  - 6:   **Primeiro Dropout:**  $X'' = F_1(X', p = 0.5)$
  - 7:   **Camada Densa:**  $Y = G(X'')$
  - 8: **Fim**
- 

adicionando um vetor de viés  $b_1$ :

$$X' = E_1(X_{\text{cnn}}) = X_{\text{cnn}} \times W_1 + b_1 \quad (4.12)$$

#### 4.4.2.6.1 Primeiro Dropout ( $F_1$ )

O *dropout* é uma técnica de regularização que desliga aleatoriamente uma fração de unidades na camada durante o treinamento. Matematicamente, isso significa multiplicar cada elemento do tensor de saída  $X'$  por um valor de máscara binária  $M_1$  onde cada elemento de  $M_1$  é 1 com probabilidade  $1 - p$  e 0 com probabilidade  $p$ , onde  $p$  é a taxa de *dropout* (neste caso,  $p = 0.5$ ):

$$X'' = F_1(X', p = 0.5) = X' \odot M_1 \quad (4.13)$$

O operador  $\odot$  indica que cada elemento de  $X'$  é multiplicado pelo elemento correspondente de  $M_1$ , resultando em um novo tensor  $X''$  onde algumas unidades foram desativadas com base na taxa de *dropout*  $p$ .

#### 4.4.2.7 Camada Densa ( $G$ )

Esta é a camada final do modelo, que transforma o tensor resultante  $X''$  em um tensor de saída  $Y$  representando as classes preditas. Isso é feito usando uma última matriz de pesos  $W_2$  e um vetor de viés  $b_2$ :

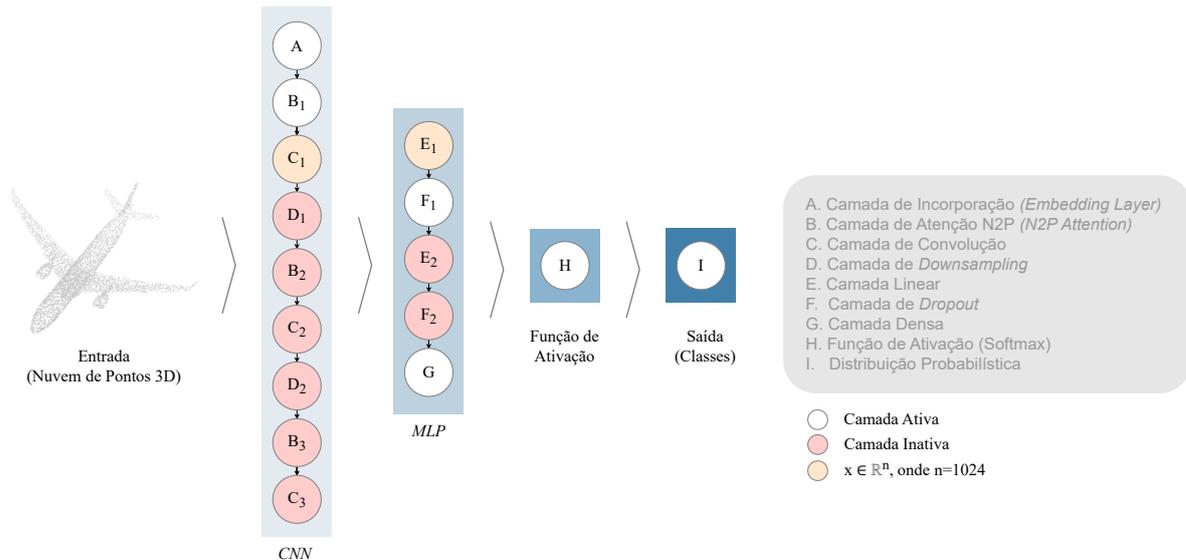
$$Y = G(X'') = X'' \times W_2 + b_2 \quad (4.14)$$

Essas operações formam o fluxo de dados na classificação, onde cada camada linear é seguida por um *dropout* para regularização, e a última camada produz as previsões finais.

### 4.4.3 Cenário III

Na Figura 10, é possível visualizar as modificações realizadas na arquitetura, vamos detalhar as operações realizadas no Cenário III, usando fórmulas matriciais para descrever cada etapa do processamento dos dados de entrada  $x$  até a produção do vetor de recursos final:

Figura 10 – Arquitetura do Cenário III.



Fonte: elaborada pelo autor.

O Algoritmo 6 representa um fluxo de dados na extração de características, onde os dados de entrada passam por várias camadas da arquitetura, para extrair características relevantes antes de serem utilizados na tarefa de classificação.

---

#### Algoritmo 6: Fluxo de dados na extração de características do Cenário III

---

- 1: **Entrada:** Tensores  $X \in \mathbb{R}^{B \times 3 \times N}$ , onde  $B$  é o tamanho do lote e  $N$  é o número de pontos.
  - 2: **Saída:** Tensores  $X_{\text{cnn}} \in \mathbb{R}^{B \times K}$ , onde  $K$  é o tamanho do vetor de características.
  - 3:
  - 4: **Início**
  - 5: **Camada de Incorporação:**  $X' = A(X)$ , onde  $A : \mathbb{R}^{B \times 3 \times N} \rightarrow \mathbb{R}^{B \times 128 \times N}$
  - 6: **Camada de Atenção:**  $X'' = B_1(X')$ , onde  $B_1 : \mathbb{R}^{B \times 128 \times N} \rightarrow \mathbb{R}^{B \times 128 \times N}$
  - 7: **Camada de Convolução:**  $X''' = C_1(X'')$ , onde  $C_1 : \mathbb{R}^{B \times 128 \times N} \rightarrow \mathbb{R}^{B \times 1024 \times N}$
  - 8: **Operação de Max Pooling e Armazenamento:**  $R = \max(X''', \text{dim} = -1)$
  - 9: **Resultados:**  $X_{\text{cnn}} = R$
  - 10: **Fim**
- 

Vamos entender cada parte deste algoritmo de extração de características:

#### 4.4.3.1 *Camada de Incorporação (A)*

Esta camada transforma o tensor de entrada  $X$  em um novo tensor  $X'$  por meio de uma camada de incorporação. Aqui, a camada de incorporação aumenta a dimensionalidade dos dados, transformando cada ponto de dados (representado por um vetor de tamanho 3) em um vetor de características de tamanho 128.

#### 4.4.3.2 *Camada de Atenção ( $B_1$ )*

As camadas de atenção aplicam mecanismos de atenção ao tensor de entrada. Neste caso, a camada de atenção é aplicada após a camada de incorporação. Ela pode modificar o tensor de entrada, destacando partes importantes ou suprimindo informações irrelevantes.

#### 4.4.3.3 *Camada de Convolução ( $C_1$ )*

As camadas de convolução aplicam operações de convolução ao tensor de entrada. Elas podem extrair características importantes dos dados, como bordas, padrões ou texturas. Aqui, a camada de convolução aumenta a dimensionalidade dos dados.

#### 4.4.3.4 *Operação de Max Pooling e Armazenamento*

Após a camada de convolução, é realizada uma operação de *max pooling* para reduzir a dimensionalidade dos dados, mantendo apenas as características mais importantes. O *max pooling* é aplicado ao tensor  $X'''$  ao longo da dimensão de características (-1).

#### 4.4.3.5 *Resultados*

O tensor resultante da operação de *max pooling* é atribuído a  $X_{\text{cnn}}$ , representando as características extraídas dos dados de entrada.

Para a classificação, segue a mesma implementação do Algoritmo 5 (Fluxo de dados na classificação do Cenário II), descrita no Cenário II 4.4.2, apenas a mudança na dimensão do vetor de características correspondente a este cenário.

## 4.5 Métricas de Avaliação

Para avaliar o desempenho das redes utilizadas neste estudo na classificação de objetos em nuvens de pontos, recorreremos a seis métricas principais: Acurácia, Precisão, Sensibilidade, *F1-Score*, Coeficiente de Correlação de Matthews (MCC) e Índice Jaccard. Essas métricas são fundamentais para entender como as redes estão classificando as nuvens de pontos em diferentes categorias.

A matriz de confusão, como pode ser vista na Figura 11, é uma ferramenta crucial na avaliação de modelos de classificação, comparando as classes reais com as previstas. Ela permite identificar facilmente os acertos e erros do modelo. Verdadeiro Positivo (VP) são as previsões corretas de classe positiva, Falso Positivo (FP) são previsões incorretas de classe positiva, Verdadeiro Negativo (VN) são previsões corretas de classe negativa, e Falso Negativo (FN) são previsões incorretas de classe negativa (GRANDINI *et al.*, 2020).

Figura 11 – Matriz de confusão para a classificação.

		Classes Previstas	
		P	N
Classes Reais	P	VP	FN
	N	FP	VN

Fonte: Elaborada pelo autor.

### 4.5.1 Acurácia

Na análise de nuvens de pontos, a Acurácia (GRANDINI *et al.*, 2020) é a medida que avalia a proporção de pontos corretamente classificados em relação ao total de pontos na nuvem. Ela nos dá uma visão geral de quão bem a rede está realizando as classificações, e é calculada a partir da Equação 4.15.

$$Acurácia = \frac{VP + VN}{VP + FP + FN + VN}. \quad (4.15)$$

#### 4.5.2 Precisão

A Precisão (GRANDINI *et al.*, 2020), por outro lado, representa a proporção de pontos corretamente classificados como pertencentes a uma determinada categoria em relação ao total de pontos que a rede classificou como pertencentes a essa categoria, assim como expressado na Equação 4.16. Isso nos ajuda a entender quão confiáveis são as classificações positivas da rede.

$$Precisão = \frac{VP}{VP + FP} \quad (4.16)$$

#### 4.5.3 Sensibilidade

A Sensibilidade (GRANDINI *et al.*, 2020), também conhecida como taxa de verdadeiros positivos, é a proporção de pontos pertencentes a uma determinada categoria que foram corretamente identificados pela rede em relação ao total de pontos reais pertencentes a essa categoria. Isso é crucial para avaliar a capacidade da rede de identificar corretamente objetos de interesse, sendo calculada a partir da Equação 4.17.

$$Sensibilidade = \frac{VP}{VP + FN} \quad (4.17)$$

#### 4.5.4 F1-Score

O F1-Score (GRANDINI *et al.*, 2020) é uma medida que combina a Precisão e a Sensibilidade em uma única pontuação, proporcionando uma avaliação mais equilibrada do desempenho da rede. Ele nos ajuda a entender o equilíbrio entre a precisão das classificações positivas e a capacidade de capturar todos os exemplos positivos, e é calculada a partir da Equação 4.18.

$$F1 - Score = 2 \cdot \frac{Precisão \cdot Sensibilidade}{Precisão + Sensibilidade} \quad (4.18)$$

#### 4.5.5 Coeficiente de Correlação de Matthews (MCC)

O Coeficiente de Correlação de Matthews (MCC) (GRANDINI *et al.*, 2020) é uma métrica para avaliar a precisão dos classificadores em nuvens de pontos. Ele fornece uma medida

abrangente da qualidade das previsões, variando de -1 a 1, onde valores mais próximos de 1 indicam uma previsão mais precisa. O MCC é essencial para compreender o desempenho do classificador diante da complexidade dos dados, permitindo uma avaliação clara da capacidade da rede em identificar corretamente as diferentes classes nas nuvens de pontos, e é calculado a partir da Equação 4.19.

$$\text{MCC} = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FN)(VP + FP)(VN + FN)(VN + FP)}} \quad (4.19)$$

#### 4.5.6 Índice Jaccard

O Índice de Jaccard (FLETCHER *et al.*, 2018) em nuvens de pontos oferece uma medida de quão eficazmente a rede identifica os pontos de um objeto em comparação com os pontos reais desse objeto. Ele é calculado como a proporção entre os pontos corretamente classificados como pertencentes ao objeto e o número total de pontos classificados ou verdadeiros. Uma pontuação elevada no Índice de Jaccard sugere uma significativa sobreposição entre as nuvens de pontos classificadas pela rede e as nuvens de pontos reais, o que indica um desempenho robusto na identificação de objetos em nuvens de pontos. O Índice de Jaccard pode ser expresso pela seguinte Equação 4.20, onde  $|A \cap B|$  representa o tamanho da interseção dos conjuntos  $A$  e  $B$ , ou seja, o número de elementos que estão presentes em ambos os conjuntos, e  $|A \cup B|$  denota o tamanho da união dos conjuntos  $A$  e  $B$ , ou seja, o número total de elementos distintos em ambos os conjuntos.

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4.20)$$

#### 4.6 Testes estatísticos

Os testes estatísticos são fundamentais para analisar e interpretar resultados experimentais, evitando conclusões precipitadas. Eles ajudam a determinar se as diferenças observadas entre o desempenho das redes são estatisticamente significativas ou ocorreram ao acaso, especialmente em conjuntos de dados complexos como nuvens de pontos. Para validar os resultados experimentais, serão realizados testes paramétricos e não paramétricos.

#### 4.6.1 Testes Paramétricos

A análise paramétrica concentra-se na amostra proveniente de uma distribuição normal (HARRIS *et al.*, 2008), onde buscamos entender se as diferenças observadas entre as redes são estatisticamente significativas, fornecendo confiança na comparação entre elas.

##### 4.6.1.1 ANOVA

Primeiramente, realizou-se o Teste ANOVA (Análise de Variância) (ST»HLE; WOLD, 1989) para determinar se as redes utilizadas apresentam resultados significativamente diferentes entre si. Para este teste, um valor de  $p < 0,05$  indica a rejeição da hipótese nula  $H_0$ . Assim, a hipótese alternativa  $H_1$  é aceita como verdadeira. A confiança utilizada foi de 0,95, e as hipóteses estão enumeradas abaixo:

**Hipótese 0 ( $H_0$ ):** *Todas as redes são estatisticamente equivalentes, o que significa que não há diferença entre utilizar uma ou outra rede para a tarefa de classificação de objetos 3D.*

**Hipótese 1 ( $H_1$ ):** *Pelo menos uma rede é diferente de outra, o que indica que existe uma rede com resultados melhores ou piores que as outras.*

No entanto, o teste ANOVA não revela quais redes são distintas entre si, portanto não é possível concluir que uma rede é superior ou inferior às outras.

##### 4.6.1.2 Tukey's HSD

Para uma análise mais detalhada, aplicou-se o Teste de Tukey's HSD (Diferença Honestamente Significativa de Tukey) (NANDA *et al.*, 2021). Este teste compara as redes uma a uma e segue o mesmo critério para aceitação ou rejeição das seguintes hipóteses:

**Hipótese 0 ( $H_0$ ):** *A rede em questão é estatisticamente equivalente a outra rede em comparação, indicando que não há diferença significativa entre usar uma ou outra rede para a tarefa de Classificação de objetos 3D.*

**Hipótese 1 ( $H_1$ ):** *As redes são estatisticamente diferentes entre si, evidenciando que há uma diferença de significância estática ao usar uma ou outra rede para a tarefa de Classificação de objetos 3D.*

#### 4.6.2 Testes Não Paramétricos

Os testes não paramétricos são úteis quando os pressupostos dos testes paramétricos não são atendidos, como quando os dados não seguem uma distribuição normal (HARRIS *et al.*, 2008). Eles oferecem uma abordagem robusta para a análise estatística, mesmo em condições adversas, permitindo uma avaliação mais abrangente do desempenho das redes sob diferentes condições experimentais.

##### 4.6.2.1 Teste de Kruskal-Wallis

Teste de Kruskal-Wallis (KRUSKAL; WALLIS, 1952) é uma alternativa não paramétrica à ANOVA, adequada quando os dados não seguem uma distribuição normal ou quando a homogeneidade de variâncias não é atendida. Este teste foi realizado para investigar se existem diferenças significativas entre as redes utilizadas com relação ao desempenho na tarefa de Classificação de objetos 3D. O nível de significância adotado foi de  $\alpha = 0,05$ .

**Hipótese 0 (  $H_0$  ):** *Não há diferenças significativas no desempenho entre as redes de classificação de nuvem de pontos.*

**Hipótese 1 (  $H_1$  ):** *Pelo menos uma das redes de classificação de nuvem de pontos têm desempenhos significativamente diferentes.*

##### 4.6.2.2 Teste de Friedman

O Teste de Friedman (FRIEDMAN, 1937), o qual é adequado para comparar múltiplos algoritmos ou modelos em várias condições de cenário. Ele é uma extensão do Teste de Kruskal-Wallis. Este teste foi conduzido com o mesmo nível de significância ( $\alpha = 0,05$ ) e visa determinar se há diferenças estatisticamente significativas entre os desempenhos das redes utilizadas.

**Hipótese 0 (  $H_0$  ):** *As redes em avaliação são estatisticamente equivalentes as outras redes comparada, sugerindo que não há uma diferença significativa entre utilizar uma ou outra rede para a tarefa de classificação de objetos 3D.*

**Hipótese 1 (  $H_1$  ):** *As redes são estatisticamente diferentes entre si, indicando que há uma diferença estatisticamente significativa ao utilizar uma ou outra rede para a tarefa de classificação de objetos 3D.*

## 5 RESULTADOS

Neste capítulo, apresentamos uma análise detalhada dos resultados obtidos nos cenários realizados. Inicialmente, apresentamos uma visão geral dos resultados, destacando as métricas de desempenho para os diferentes métodos avaliados. Em seguida, exploramos os resultados específicos de cada cenário, comparando os modelos em termos de acurácia, precisão, sensibilidade, *F1-Score*, MCC e Índice de Jaccard.

Além dessa avaliação direta, aplicamos uma rigorosa análise estatística para avaliar a significância das diferenças observadas entre os métodos. Utilizamos uma variedade de testes estatísticos, incluindo ANOVA, Tukey's HSD, o teste de Kruskal-Wallis e o teste de Friedman. Esses testes nos permitiram uma compreensão mais profunda da variação entre os métodos e suas implicações.

Por fim, oferecemos informações detalhadas sobre o tempo de treinamento e o consumo de memória para cada modelo, proporcionando uma visão completa do desempenho e eficiência de cada abordagem.

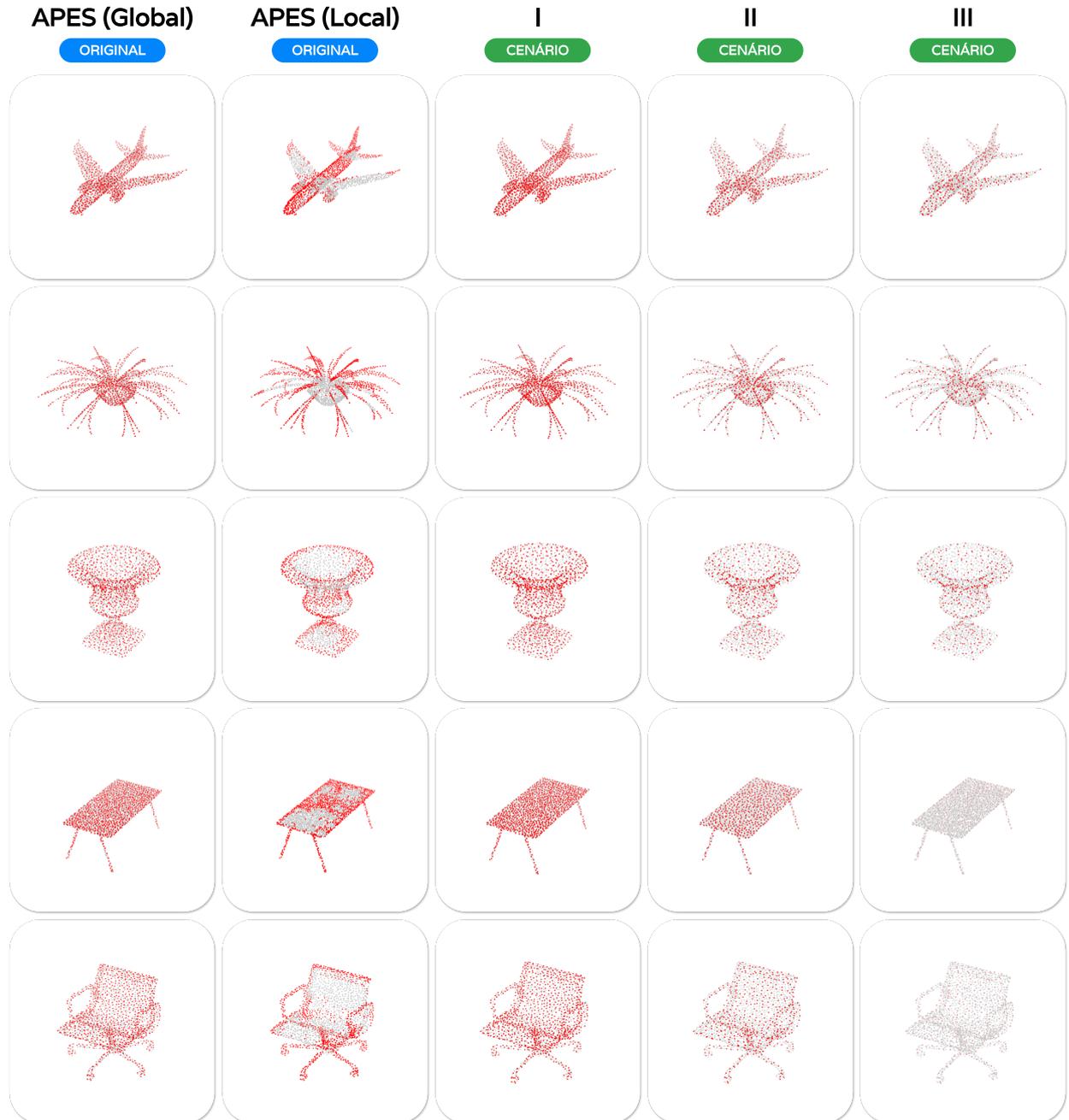
### 5.1 Visão Geral dos Resultados

A Figura 12 proporciona uma representação visual das características que serão consideradas durante o treinamento da rede, apresentadas sob a forma de nuvens de pontos originais do conjunto de teste para os diferentes métodos analisados a fundo. Essa visualização permite uma avaliação imediata do desempenho relativo de cada método de classificação. Ao examinar as nuvens de pontos, é possível discernir quais características específicas dos dados originais estão sendo enfatizadas para distinguir e classificar os objetos. Isso fornece percepções valiosas sobre as informações que as redes neurais irão utilizar durante o processo de aprendizagem, ajudando a compreender quais características são mais relevantes para cada método de classificação.

Além disso, ao observar as distribuições de pontos para cada método, é possível inferir quais características estão sendo enfatizadas ou desconsideradas durante o processo de classificação. Essas percepções são essenciais para ajustar e otimizar os métodos de classificação, garantindo um desempenho ideal em diferentes cenários e tipos de dados.

A Tabela 3 apresenta uma comparação entre os modelos propostos - APES (Global), APES (Local) e os Cenários (I, II e III) - com uma seleção de trabalhos relacionados na tarefa de classificação utilizando o conjunto de dados ModelNet40. A métrica de avaliação considerada é

Figura 12 – Visualização dos resultados com as nuvens de pontos do conjunto de teste, para os diferentes métodos.



Fonte: Elaborada pelo autor.

a acurácia geral, que representa a proporção de amostras classificadas corretamente em relação ao total de amostras.

Dentro da lista de modelos apresentados na Tabela 3, todos foram considerados para análise comparativa com os Cenários I, II e III em relação à eficácia na classificação de dados 3D usando o conjunto de dados ModelNet40. Os modelos PointNet, PointNet++, SpiderCNN, DGCNN, PointCNN, PointConv, PVCNN, KPConv, PointASNL,  $PT^1$ ,  $PT^2$ , PCT, PRA-Net, PAConv, CurveNet, DeltaConv, APES (Local) e APES (Global), alcançaram a acurácia geral

Tabela 3 – Tabela de geral de resultados da classificação com o conjunto de dados ModelNet40.

Modelos	Acurácia Geral
PointNet (QI <i>et al.</i> , 2017a)	89,2%
PointNet++ (QI <i>et al.</i> , 2017b)	91,9%
SpiderCNN (XU <i>et al.</i> , 2018)	92,4%
DGCNN (WANG <i>et al.</i> , 2019)	92,9%
PointCNN (LI <i>et al.</i> , 2018)	92,2%
PointConv (WU <i>et al.</i> , 2019)	92,5%
PVCNN (LIU <i>et al.</i> , 2019)	92,4%
KPCConv (THOMAS <i>et al.</i> , 2019)	92,9%
PointASNL (YAN <i>et al.</i> , 2020)	93,2%
$PT^1$ (ENGEL <i>et al.</i> , 2021)	92,8%
$PT^2$ (ZHAO <i>et al.</i> , 2021)	93,7%
PCT (GUO <i>et al.</i> , 2021)	93,2%
PRA-Net (CHENG <i>et al.</i> , 2021)	93,7%
PACConv (XU <i>et al.</i> , 2021)	93,6%
CurveNet (MUZAHID <i>et al.</i> , 2020)	93,8%
DeltaConv (WIERSMA <i>et al.</i> , 2022)	93,8%
APES (Local) (WU <i>et al.</i> , 2023)	93,5%
APES (Global) (WU <i>et al.</i> , 2023)	93,8%
Cenário I	93,6%
Cenário II	93,8%
Cenário III	93,4%

Fonte: Elaborada pelo autor.

variando de 89,2% a 93,8%.

Os modelos que atingiram as maiores acurácias na tabela são CurveNet, DeltaConv e APES (Global), todos com 93,8%. Em seguida, temos  $PT^2$  e PRA-Net com 93,7%. Notavelmente, Cenário (II) também alcançou uma acurácia de 93,8%, posicionando-o ao lado de CurveNet, DeltaConv, e APES (Global), os modelos de referência com as maiores pontuações.

Enquanto isso, Cenário (I) registrou uma acurácia de 93,6%, ficando próximo dos modelos de ponta, mas ligeiramente abaixo do desempenho máximo, superando até mesmo a APES (Local). Já Cenário (III) apresentou uma acurácia de 93,4%, o que o coloca um pouco abaixo do ponto de referência estabelecido pelos modelos mais bem classificados.

Essa comparação destaca a eficácia dos Cenários (I) e (II) em se equiparar aos modelos líderes, sugerindo que essas abordagens realizadas pela Análise de Ablação 4.4 são capazes de oferecer desempenho competitivo em relação aos modelos de ponta. Cenário (III), embora não tenha atingido a mesma acurácia, ainda demonstra um desempenho sólido, o que sugere que mesmo as variações menos otimizadas podem fornecer resultados promissores.

Em resumo, os Cenários (I) e (II) se destacam como candidatos viáveis para o modelo proposto APES-Soft, mostrando-se capazes de competir com os melhores modelos da tabela em termos de acurácia geral. Esses resultados validam a eficácia das técnicas propostas e sua

relevância para a pesquisa atual em classificação de dados 3D.

Tabela 4 – Avaliação métrica dos modelos.

<b>Modelos</b>	<b>Acurácia</b>	<b>Precisão</b>	<b>Sensibilidade</b>	<b>F1-Score</b>	<b>MCC</b>	<b>Jaccard</b>
APES (Global)	0,93709	0,93673	0,93709	0,93648	0,93486	0.89082
APES (Local)	0,93263	0,93029	0,93263	0,93063	0,93024	0.88219
Cenário I	0,93588	0,93506	0,93588	0,93488	0,93359	0.88871
Cenário II	0,93750	0,93698	0,93750	0,93680	0,93527	0.89204
Cenário III	0,93425	0,93487	0,93425	0,93385	0,93192	0.88670

Fonte: Elaborada pelo autor.

A Tabela 4 apresenta uma avaliação métrica detalhada dos modelos APES (Global), APES (Local) e os Cenários I, II e III. Dentre os modelos apresentados, os Cenários (I), (II) e (III) estão sendo considerados como candidatos ao modelo proposto APES-Soft, conforme sugerido pela abordagem de Análise de Ablação 4.4. Vamos comparar esses cenários com os melhores modelos da Tabela 3 para entender melhor sua eficácia relativa.

Considerando que a Análise de Ablação 4.4 é focada na rede APES, os resultados obtidos na configuração atual dos cenários são fundamentais para entender o desempenho dessa rede. Além da acurácia, é importante analisar outras métricas para entender completamente o desempenho dos modelos. A precisão, sensibilidade, F1-Score, MCC e Jaccard são métricas cruciais que fornecem um entendimento sobre o desempenho em diferentes aspectos da classificação.

O Cenário II demonstra consistência em todas essas métricas, com valores altos e próximos, indicando um bom equilíbrio entre precisão e sensibilidade. Essa uniformidade sugere uma capacidade robusta de classificação em várias classes.

Por outro lado, o APES (Global), APES (Local) e os Cenários I e III apresentam resultados ligeiramente inferiores em comparação com o Cenário II, mas ainda mantêm desempenhos sólidos. A diferença de desempenho entre o APES (Global) e o APES (Local) são atribuída à maneira como cada modelo captura e utiliza informações espaciais globais e locais dos dados.

O Cenário II se destaca com a maior acurácia entre os cenários, alcançando 93,8%, o que o coloca em uma posição favorável para ser considerado como a versão APES-Soft proposta neste estudo, onde a APES (Global) atingiu a acurácia de 93,7% nos cenários utilizando as configurações descritas em Análise de Ablação 4.4. Suas métricas de precisão (93,7%), sensibilidade (93,8%), F1-Score (93,7%), MCC (93,5%) e Jaccard (89,2%) são competitivas e próximas às do APES (Global) (93,7%, 93,7%, 93,6%, 93,5%, e 89,1%, respectivamente), sugerindo que ele pode herdar as vantagens do modelo original.

## 5.2 Cenário I

O Cenário I 4.4.1 registrou uma acurácia de 93,6%, o que o coloca em uma posição competitiva em relação aos modelos de ponta na tabela de resultados. Sua precisão (93,5%), sensibilidade (93,6%), *F1-Score* (93,5%), MCC (93,4%) e Jaccard (88,9%) são próximas às do APES (Global), sugerindo que este cenário tem um desempenho consistente em várias métricas de avaliação.

Comparado ao Cenário II, o Cenário I apresenta resultados ligeiramente inferiores em todas as métricas, indicando uma performance ligeiramente menor. No entanto, essa diferença é mínima e não compromete a eficácia geral do Cenário I como um candidato viável para o modelo proposto APES-Soft. A diferença está relacionada à ausência de *downsampling*, o que afeta sutilmente a capacidade do modelo de capturar características detalhadas dos dados.

No contexto da Análise de Ablação 4.4, o Cenário I fornece informações valiosas sobre como a remoção do *downsampling* impacta o desempenho da rede. Os resultados indicam que, embora a remoção do *downsampling* possa levar a uma pequena redução na acurácia, o desempenho geral do modelo ainda é competitivo em comparação com os modelos de referência.

Em resumo, o Cenário I demonstra um desempenho sólido e consistente em relação aos modelos de ponta, destacando sua viabilidade como uma alternativa eficaz na classificação de dados 3D.

## 5.3 Cenário II

O Cenário II 4.4.2 se destaca com a maior acurácia entre os cenários, atingindo 93,8%, o que o coloca em uma posição favorável para ser considerado como a versão APES-Soft proposta neste estudo de Ablação. Suas métricas de precisão (93,7%), sensibilidade (93,8%), *F1-Score* (93,7%), MCC (93,5%) e Jaccard (89,2%) são competitivas e próximas às do APES (Global), sugerindo que ele pode herdar as vantagens do modelo original.

Comparado ao Cenário I, o Cenário II apresenta resultados ligeiramente melhores em todas as métricas, indicando uma performance geralmente superior. Essa diferença pode ser atribuída à modificação adicional realizada no Cenário II, além da remoção do *downsampling* presente no Cenário I. Essas modificações permitiu ao Cenário II capturar e utilizar melhor as informações dos dados, resultando em um desempenho aprimorado.

No contexto da Análise de Ablação 4.4, o Cenário II fornece detalhes valiosos

sobre como as modificações na arquitetura da rede, e assim entender como influenciam seu desempenho, como é possível visualizar na Figura 12, as modificações na extração dos atributos teve influencia direta nas características obtidas pelo modelo para o treinamento, sendo possível visualizar a distribuição mais uniforme dos pontos, e preservando o formato original do objeto. Os resultados destacam a eficácia dessas modificações em melhorar a acurácia geral do modelo, além de manter um equilíbrio entre precisão e sensibilidade.

Em resumo, o Cenário II demonstra uma performance excepcional e consistente em relação aos modelos de referência, consolidando sua posição como um candidato viável para o modelo proposto APES-Soft. Sua capacidade de competir com o modelo de referência e os resultados sólidos em todas as métricas destacam sua viabilidade como uma alternativa eficaz na classificação de dados 3D.

#### **5.4 Cenário III**

O Cenário III 4.4.3, embora apresente uma acurácia de 93,4%, um pouco abaixo do Cenário II, ainda demonstra um desempenho sólido em relação aos modelos de referência. Suas métricas de precisão (93,5%), sensibilidade (93,4%), *F1-Score* (93,4%), MCC (93,2%) e Jaccard (88,7%) são próximas às do APES (Global), sugerindo que ele pode oferecer resultados competitivos em comparação com o modelo original.

Comparado ao Cenário II, o Cenário III mostra uma ligeira queda no desempenho em todas as métricas. Isso pode ser atribuído à modificação adicional realizada no Cenário III onde é possível entender em detalhes na Análise de Ablação 4.4, onde manteve-se uma arquitetura baseado no Cenário II exceto pela redução do vetor de características na fase de extração de características da rede, impactando negativamente na capacidade do modelo em capturar e utilizar efetivamente as informações dos dados.

Apesar da ligeira queda no desempenho em relação ao Cenário II, o Cenário III ainda se mantém como uma opção viável para o modelo proposto APES-Soft. Sua acurácia próxima aos 93,4% e suas métricas consistentes sugerem que ele pode ser uma alternativa eficaz na classificação de dados 3D, especialmente considerando a complexidade e a variedade desses conjuntos de dados.

## 5.5 Testes Estatísticos

Os Testes Estatísticos 4.6 desempenham um papel fundamental na análise dos resultados obtidos nos cenários. Eles permitem determinar se as diferenças observadas entre os métodos são estatisticamente significativas, fornecendo uma base sólida para a interpretação e validação dos resultados. Neste estudo, foram realizados testes *one-way* ANOVA 4.6.1.1, teste Tukey's HSD 4.6.1.2, teste de Kruskal-Wallis 4.6.2.1 e teste de Friedman 4.6.2.2 para avaliar a significância das diferenças entre os métodos em relação a várias métricas de desempenho, para todos os testes estatísticos foram definidos o Limiar de Significância ( $\alpha$ ) de  $\alpha = 0,05$  como discutido na literatura.

### 5.5.1 ANOVA

A Tabela 5 apresenta os resultados do teste *one-way* ANOVA, uma análise estatística usada para avaliar a significância das diferenças entre os modelos em relação a várias métricas de desempenho. Cada métrica, incluindo acurácia, precisão, sensibilidade, *F1-Score*, MCC e Índice Jaccard, foi avaliada separadamente.

Para cada métrica, foram calculados os graus de liberdade (*Df*), as somas de quadrados (*SS*), os quadrados médios (*MS*), o valor *F* (*F-value*) e o valor *p* (*p-value*). O valor *p* é particularmente importante, pois indica a probabilidade de observar diferenças tão grandes quanto as observadas entre os modelos, se a hipótese nula  $H_0$  de que não há diferenças reais entre os grupos for verdadeira ou que, pelo menos, uma das métricas apresenta diferenças significativas  $H_1$ .

Todos os valores de *p* são maiores do que o  $\alpha$  pré-definido de 0,05, o que sugere que não há evidências estatísticas para rejeitar a hipótese nula de que não existem diferenças reais entre os grupos. Essa descoberta é encorajadora, pois indica que, apesar das variações nos resultados, as modificações feitas nos modelos, como no Cenário II, não afetaram negativamente as métricas de desempenho. Em vez disso, as variações observadas podem ser atribuídas à aleatoriedade nos dados. Essa análise estatística robusta fornece uma compreensão mais profunda da variação entre os métodos, destacando a consistência e a equivalência dos resultados entre os diferentes modelos avaliados. Essa consistência fortalece a confiança na interpretação dos resultados dos cenários, reforçando a validade e a robustez das conclusões obtidas.

Tabela 5 – Resultados do teste *one-way* ANOVA para os cenários.

Métrica		Df	SS	MS	F-value	p-value
Acurácia	C(Modelos)	1,0	$6,56 \times 10^{-8}$	$6,56 \times 10^{-8}$	0,012001	0,919685
	Residual	3,0	$1,64 \times 10^{-5}$	$5,47 \times 10^{-6}$	-	-
Precisão	C(Modelos)	1,0	$8,82 \times 10^{-7}$	$8,82 \times 10^{-7}$	0,094487	0,778652
	Residual	3,0	$2,80 \times 10^{-5}$	$9,34 \times 10^{-6}$	-	-
Sensibilidade	C(Modelos)	1,0	$6,56 \times 10^{-8}$	$6,56 \times 10^{-8}$	0,012001	0,919685
	Residual	3,0	$1,64 \times 10^{-5}$	$5,47 \times 10^{-6}$	-	-
F1-Score	C(Modelos)	1,0	$8,28 \times 10^{-8}$	$8,28 \times 10^{-8}$	0,010071	0,926393
	Residual	3,0	$2,47 \times 10^{-5}$	$8,22 \times 10^{-6}$	-	-
MCC	C(Modelos)	1,0	$7,23 \times 10^{-8}$	$7,23 \times 10^{-8}$	0,012373	0,918454
	Residual	3,0	$1,75 \times 10^{-5}$	$5,84 \times 10^{-6}$	-	-
Jaccard	C(Modelos)	1,0	$2,59 \times 10^{-7}$	$2,59 \times 10^{-7}$	0,012977	0,916499
	Residual	3,0	$5,99 \times 10^{-5}$	$1,99 \times 10^{-5}$	-	-

Fonte: Elaborado pelo autor.

### 5.5.2 Tukey's HSD

O teste de Tukey's HSD é uma técnica estatística utilizada para determinar quais pares de médias são significativamente diferentes entre si após a realização de uma ANOVA. Ele é particularmente útil quando se deseja comparar múltiplos grupos e identificar onde as diferenças significativas estão ocorrendo. Como o  $\alpha$  usualmente adotado é de  $\alpha = 0,05$ , se o valor de  $p$  for maior do que 0,05, não há diferença estatisticamente significativa entre os grupos comparados.

Na Tabela 6, são apresentados os resultados do valor de  $p$  (*p-value*) do teste de Tukey's HSD para os cenários realizados. Cada linha da tabela compara dois grupos diferentes e exhibe os valores médios de várias métricas de desempenho, como acurácia, precisão, sensibilidade, F1-Score, MCC e Índice Jaccard. Assim como na ANOVA a hipótese nula  $H_0$  de que não há diferenças reais entre os grupos for verdadeira ou que, pelo menos, uma das métricas apresenta diferenças significativas  $H_1$ .

Quanto aos resultados, sugere que, apesar das variações nos valores médios das métricas de desempenho entre os diferentes grupos, essas variações não são estatisticamente significativas. Em outras palavras, os modelos comparados não apresentam diferenças estatisticamente significativas em seu desempenho com base nas métricas consideradas.

Essa análise reforça a conclusão de que as modificações realizadas nos cenários (Cenário I, II e III) não afetaram significativamente o desempenho dos modelos em relação ao

Tabela 6 – Resultados de *p-value* dos cenários para o teste Tukey's HSD.

Grupo 1	Grupo 2	Acurácia	Precisão	Sensibilidade	F1-Score	MCC	Jaccard
APES (Global)	APES (Local)	0.688675	0.629384	0.688675	0.649400	0.687352	0.681710
APES (Global)	Cenário I	0.900000	0.900000	0.900000	0.900000	0.900000	0.900000
APES (Global)	Cenário II	0.900000	0.900000	0.900000	0.900000	0.900000	0.900000
APES (Global)	Cenário III	0.893774	0.900000	0.893774	0.900000	0.893164	0.900000
APES (Local)	Cenário I	0.841867	0.791183	0.841867	0.814576	0.842934	0.821470
APES (Local)	Cenário II	0.636769	0.605164	0.636769	0.616363	0.637127	0.600901
APES (Local)	Cenário III	0.900000	0.809593	0.900000	0.900000	0.900000	0.900000
Cenário I	Cenário II	0.900000	0.900000	0.900000	0.900000	0.900000	0.900000
Cenário I	Cenário III	0.900000	0.900000	0.900000	0.900000	0.900000	0.900000
Cenário II	Cenário III	0.841867	0.900000	0.841867	0.900000	0.842934	0.899624

Fonte: Elaborada pelo autor.

modelo de referência (APES), conforme avaliado pelas métricas de desempenho consideradas.

### 5.5.3 Teste de Kruskal-Wallis

O teste de Kruskal-Wallis é uma versão não paramétrica da ANOVA, especialmente útil quando os dados não seguem uma distribuição normal ou quando a homogeneidade de variâncias não é atendida. Neste estudo, o teste foi conduzido para investigar se existem diferenças significativas entre as redes de classificação de nuvem de pontos em relação ao desempenho na tarefa de Classificação de objetos 3D. O nível de significância adotado foi de  $\alpha = 0,05$ , onde a hipótese nula  $H_0$  de que não há diferenças reais entre os grupos for verdadeira ou que, pelo menos, uma das métricas apresenta diferenças significativas  $H_1$ .

O teste é usado para comparar três ou mais grupos independentes quando a suposição de normalidade não é atendida. No Quadro 1, a estatística de teste é 5,0 e o valor *p* é 0,41588. Isso significa que, com um nível de significância de 0,05, não temos evidência suficiente para rejeitar a hipótese nula. Portanto, não podemos concluir que há diferença significativa entre os grupos em termos das métricas de desempenho consideradas.

Quadro 1 – Resultados do Teste de Kruskal-Wallis.

Modelos	Estatística	<i>p-value</i>
Tabela 4	5,0	0,41588

Fonte: Elaborado pelo autor

#### 5.5.4 Teste de Friedman

O Teste de Friedman é uma extensão não paramétrica da ANOVA, aplicável quando os dados não seguem uma distribuição normal ou quando a homogeneidade de variâncias não é satisfeita. Neste estudo, o teste foi empregado para examinar se existem diferenças significativas entre as redes de classificação de nuvem de pontos em relação ao desempenho na tarefa de Classificação de objetos 3D. O nível de significância utilizado foi de  $\alpha = 0,05$ .

A hipótese nula  $H_0$  indica que não há diferenças significativas no desempenho entre as redes de classificação de nuvem de pontos, já a  $H_1$  indica que menos uma das redes de classificação de nuvem de pontos apresenta desempenho significativamente diferente.

No Quadro 2, a estatística de teste é 5.0 e o valor de  $p$  é 0.41588. Com um nível de significância de 0,05, não há evidência suficiente para rejeitar a hipótese nula. Portanto, não podemos concluir que há diferença significativa entre os grupos em termos das métricas de desempenho consideradas.

Quadro 2 – Resultados do Teste de Friedman.

Modelos	Estatística	<i>p-value</i>
Tabela 4	5,0	0,41588

Fonte: Elaborado pelo autor

Os testes de Kruskal-Wallis e Friedman produziram os mesmos valores de estatística de teste e valores  $p$ , assim como é possível visualizar nos Quadros 1 e 2. Isso ocorre porque ambos os testes são conceitualmente semelhantes e podem ser utilizados para comparar grupos independentes quando a suposição de normalidade não é atendida. No caso específico deste estudo, ambos os testes foram aplicados para examinar se havia diferenças significativas entre as redes de classificação de nuvem de pontos em relação ao desempenho na tarefa de Classificação de objetos 3D.

O teste de Kruskal-Wallis é uma extensão não paramétrica da ANOVA e é aplicado quando se deseja comparar três ou mais grupos independentes. Ele avalia se as medianas dos grupos são iguais, em vez de comparar as médias, como na ANOVA. Por outro lado, o teste de Friedman é uma versão não paramétrica do teste ANOVA de medidas repetidas e é usado quando as observações são emparelhadas ou relacionadas.

Portanto, ambos os testes são adequados para este contexto e produziram resultados similares, indicando que não há evidência suficiente para rejeitar a hipótese nula de que

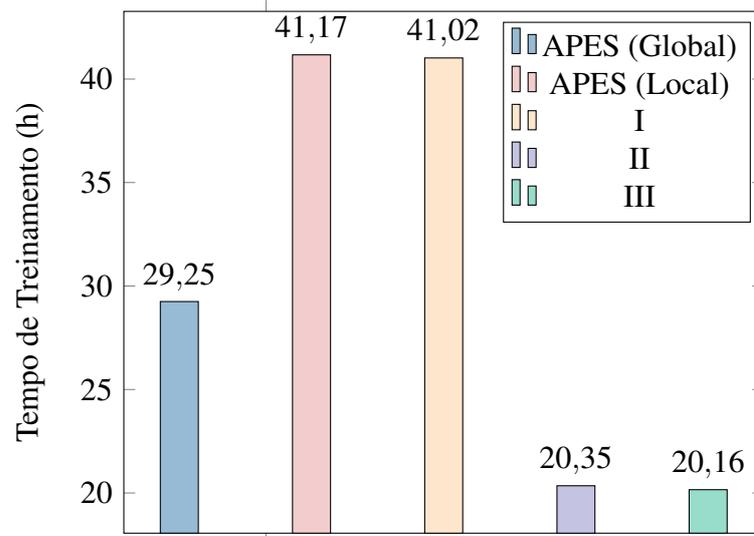
não existem diferenças significativas entre os grupos em termos das métricas de desempenho consideradas.

## 5.6 Desempenho Computacional

A Figura 13 mostra o tempo necessário para treinar cada modelo. Observa-se que o modelo III foi o mais rápido, com um tempo de treinamento de aproximadamente 20 horas, seguido de perto pelo modelo II, com cerca de 20,35 horas. Em seguida, temos o modelo APES (Global) com aproximadamente 29,25 horas, seguido pelo modelo I com aproximadamente 41,02 horas, e, por fim, o modelo APES (Local) com cerca de 41,17 horas.

Esses dados sugerem que os modelos III e II foram os mais eficientes em termos de tempo de treinamento, enquanto os modelos I, APES (Global) e APES (Local) exigiram um tempo significativamente maior para atingir o mesmo nível de treinamento.

Figura 13 – Tempo de treinamento por modelo.

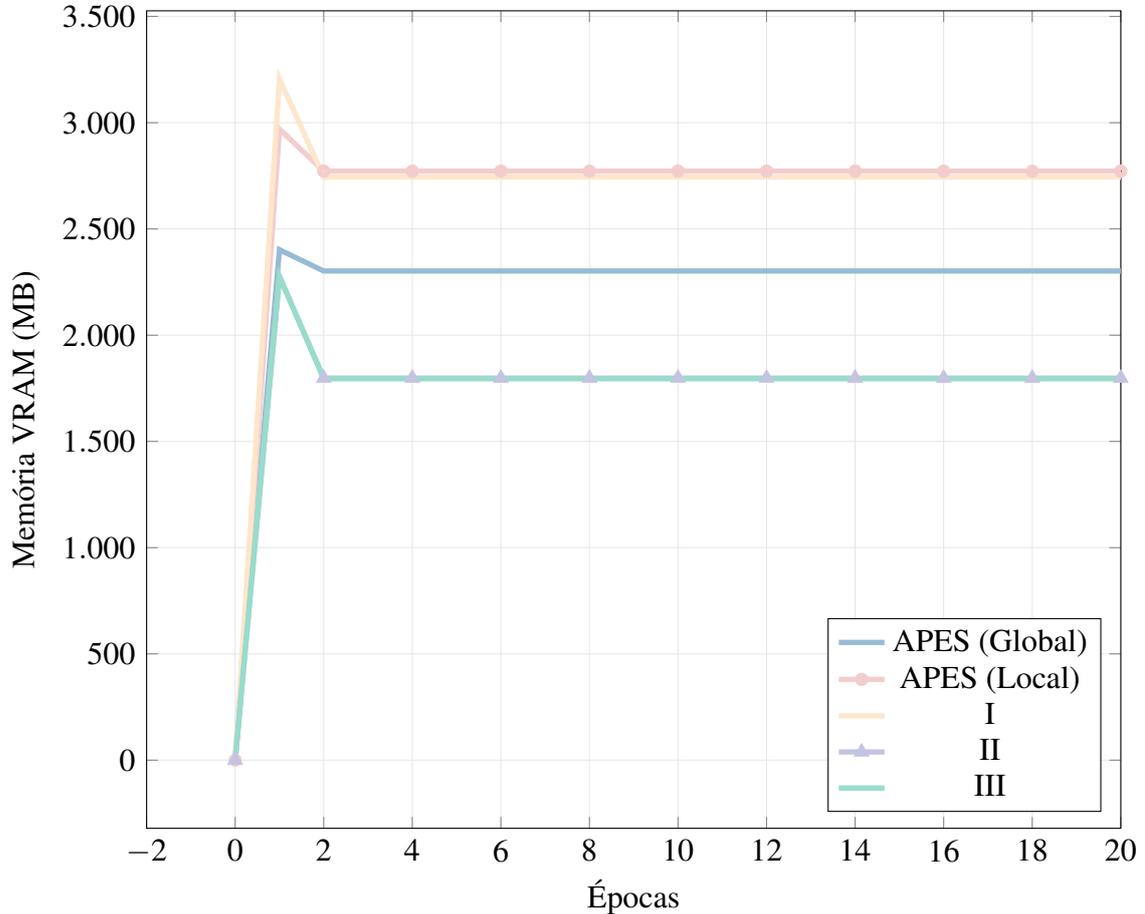


Fonte: Elaborada pelo autor.

A análise do desempenho em comparação com o Modelo APES (Global) revela informações importantes sobre os Cenários I, II e III. O Cenário I mostrou-se aproximadamente 40,24% mais lento que o Modelo APES (Global), o que o coloca em uma categoria semelhante ao APES (Local) em termos de eficiência de treinamento. Por outro lado, o Cenário II apresentou uma melhoria significativa, sendo aproximadamente 30,43% mais rápido que o Modelo APES (Global). O Cenário III seguiu uma tendência semelhante de eficiência, sendo cerca de 31,08% mais rápido que o Modelo APES (Global). Esses resultados destacam a importância de considerar

diferentes abordagens experimentais para otimizar o tempo de treinamento e, conseqüentemente, o desempenho do modelo.

Figura 14 – Uso de memória durante o treinamento.



Fonte: Elaborada pelo autor.

A Figura 14 ilustra a evolução do consumo de memória ao longo de 200 épocas durante o treinamento de um modelo. Cada linha representa uma estratégia de treinamento distinta, refletindo diferentes padrões de alocação de memória. Os picos que ocorrem no início do treinamento, decorrem da necessidade do sistema de alocar memória para armazenar todos os parâmetros do modelo, os dados de entrada, *buffers* intermediários necessários para o cálculo dos gradientes, entre outros. Além disso, as consultas de informações como detalhes sobre a plataforma (como versões do Python, CUDA, PyTorch etc.), configurações do modelo (como tipos de camadas, hiperparâmetros de otimização, esquema de treinamento), e informações de inicialização de parâmetros, também consomem memória, o que reflete nos requisitos mínimos de *hardware* para execução do modelo.

A rede APES (Global) mantém um nível de consumo de memória relativamente

estável em torno de 2302 MB, evidenciando uma eficiência consistente ao longo do tempo. Por outro lado, a estratégia APES (Local) também mantém uma estabilidade, porém com um consumo de memória ligeiramente superior, aproximando-se de 2772 MB. Os Cenários II e III compartilham padrões semelhantes de consumo de memória, caracterizados por uma queda acentuada no início, seguida por uma estabilização em cerca de 1798 MB. Embora o Cenário I apresente um consumo inicial de memória ligeiramente mais elevado em comparação com os Cenários II e III, o Cenário III se destaca por demonstrar uma eficiência ligeiramente superior, mantendo-se em torno de 1795 MB.

Tabela 7 – Estatísticas do uso de memória VRAM (MB).

<b>Modelo</b>	<b>Mediana</b>	<b>Valor Máximo</b>	<b>Valor Mínimo</b>	<b>1º Quartil</b>	<b>3º Quartil</b>
APES (Global)	2302	2402	2302	2302	2302
APES (Local)	2772	2970	2772	2772	2772
Cenário I	2744	3206	2744	2744	2744
Cenário II	1798	2277	1798	1798	1798
Cenário III	1795	2275	1795	1795	1795

Fonte: Elaborada pelo autor.

Com base nas estatísticas detalhadas na Tabela 7, podemos extrair várias conclusões sobre o consumo de memória VRAM (MB) em relação a cada modelo ou cenário. As medidas de tendência central, como a mediana, o valor máximo e o valor mínimo, exibem variações consideráveis entre os diferentes modelos e cenários. Por exemplo, o modelo APES (Local) apresenta uma mediana de 2772 MB, enquanto os Cenários II e III registram medianas em torno de 1798 MB e 1795 MB, respectivamente. Essa diversidade de valores sugere demandas distintas de memória entre os modelos e cenários.

Ao analisar os quartis, observamos que tanto a mediana quanto o primeiro e terceiro quartis permanecem consistentes para cada modelo ou cenário. Isso indica uma estabilidade no consumo de memória ao longo do tempo ou das iterações, sugerindo pouca variabilidade significativa durante os testes ou iterações. Por outro lado, os valores máximos e mínimos revelam variações extremas no consumo de memória em determinados casos. Por exemplo, o Cenário I registrou um valor máximo de 3206 MB, enquanto os Cenários II e III alcançaram valores máximos em torno de 2277 MB e 2275 MB, respectivamente.

Essas diferenças nos valores máximos e mínimos se refletem nos ganhos percentuais de memória em relação ao modelo APES (Global). O Cenário I apresentou um aumento de 19,20% no consumo de memória em comparação com o APES (Global), aproximando-se do consumo do APES (Local). Já os Cenários II e III, registraram uma redução significativa no

consumo de memória, sendo de 21,89% e 22,02%, respectivamente.

Por fim, os valores do primeiro e terceiro quartis sugerem padrões de distribuição semelhantes entre os diferentes modelos e cenários. Isso pode indicar uma consistência nos padrões de distribuição dos dados de consumo de memória VRAM, independentemente do modelo ou cenário em questão.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação propôs o desenvolvimento da rede neural APES-Soft, com o objetivo de aprimorar a eficiência computacional e o desempenho na classificação de nuvens de pontos, a partir da análise de ablação da rede APES. Três cenários foram avaliados: o Cenário I examinou o impacto da remoção das camadas de *downsampling* na extração de características; o Cenário II, além de remover essas camadas, ajustou as camadas convolucionais e lineares e reduziu o vetor de características para analisar a redução gradual da dimensionalidade; e o Cenário III manteve as modificações do Cenário II, e reduziu ainda mais o vetor de características para avaliar o desempenho do modelo.

O Cenário I demonstrou um desempenho sólido, com uma acurácia de 93,6% e métricas consistentes próximas às do modelo de referência, sugerindo sua viabilidade como uma alternativa competitiva. A análise de ablação revelou que a remoção do *downsampling* afetou marginalmente o desempenho do modelo, mas ainda assim permaneceu competitivo em relação aos outros métodos considerados.

Por outro lado, o Cenário II se destacou com a maior acurácia entre os cenários, atingindo 93,8% e apresentando métricas superiores em comparação com os Cenários I e III, APES (Global) e APES (Local), e sendo equivalente com os melhores trabalhos relacionados como CurveNet e DeltaConv. A análise de ablação destacou a importância das modificações na arquitetura da rede para aprimorar o desempenho do modelo, especialmente na extração de características.

Embora o Cenário III tenha registrado uma acurácia ligeiramente inferior em comparação com o Cenário II, suas métricas ainda foram consistentes e próximas às do modelo de referência. A análise de ablação indicou que a redução do vetor de características teve um impacto negativo no desempenho do modelo, mas ainda assim manteve sua viabilidade como uma opção competente.

Além disso, os testes estatísticos, como da ANOVA, Tukey's HSD, Tesde de Kruskal-Wallis e Tesde de Friedman, revelaram que as diferenças observadas entre os métodos não foram estatisticamente significativas, indicando equivalência estatística entre eles.

Ao considerar o tempo de treinamento, os modelos III e II se destacaram como os mais eficientes, exigindo aproximadamente 20,16 horas e 20,35 horas, respectivamente. Em contraste, os modelos I, APES (Global) e APES (Local) demandaram um tempo significativamente maior para atingir um nível comparável de treinamento.

Com base nessas análises, o Cenário II emerge como a escolha mais promissora para ser proposto como APES-Soft. Seu desempenho superior em todas as métricas avaliadas, sua competitividade em relação ao modelo de referência e sua eficiência computacional destacam-no como uma alternativa robusta e eficaz na classificação de nuvens de pontos.

Para trabalhos futuros, é possível expandir o desenvolvimento da APES-Soft através de testes em conjuntos de dados diversos, permitindo avaliar sua adaptação a diferentes cenários reais. Além disso, explorar a aplicação da APES-Soft na segmentação de nuvens de pontos 3D pode ampliar seu uso em áreas como reconstrução 3D e realidade aumentada. Integração da APES-Soft em dispositivos IoT também representa uma oportunidade promissora, possibilitando seu uso em tempo real e em ambientes distribuídos. Essas pesquisas podem não só aprimorar a compreensão da APES-Soft, mas também expandir suas aplicações práticas em diversos campos, como análise geoespacial, monitoramento ambiental, robótica e indústria.

## REFERÊNCIAS

- AL-HUDA, Z.; PENG, B.; ALGBURI, R. N. A.; AL-ANTARI, M. A.; AL-JARAZI, R.; ZHAI, D. A hybrid deep learning pavement crack semantic segmentation. **Engineering Applications of Artificial Intelligence**, v. 122, p. 106142, 2023. ISSN 0952-1976. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0952197623003263>.
- ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: IEEE. **2017 international conference on engineering and technology (ICET)**. [S. l.], 2017. p. 1–6.
- BALADO, J.; MARTÍNEZ-SÁNCHEZ, J.; ARIAS, P.; NOVO, A. Road environment semantic segmentation with deep learning from mls point cloud data. **Sensors**, MDPI, v. 19, n. 16, p. 3466, 2019.
- BRIDLE, J. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. **Advances in neural information processing systems**, v. 2, 1989.
- BURTON, D.; DUNLAP, D. B.; WOOD, L. J.; FLAIG, P. P. Lidar intensity as a remote sensor of rock properties. **Journal of Sedimentary Research**, SEPM Society for Sedimentary Geology, v. 81, n. 5, p. 339–347, 2011.
- CAI, L.; GAO, J.; ZHAO, D. A review of the application of deep learning in medical image classification and segmentation. **Annals of translational medicine**, AME Publications, v. 8, n. 11, 2020.
- CHEN, Y.; LIU, S.; SHEN, X.; JIA, J. Fast point r-cnn. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S. l.: s. n.], 2019. p. 9775–9784.
- CHENG, S.; CHEN, X.; HE, X.; LIU, Z.; BAI, X. Pra-net: Point relation-aware network for 3d point cloud analysis. **IEEE Transactions on Image Processing**, IEEE, v. 30, p. 4436–4448, 2021.
- DEY, P.; NAG, K.; PAL, T.; PAL, N. R. Regularizing multilayer perceptron for robustness. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, IEEE, v. 48, n. 8, p. 1255–1266, 2017.
- DIAB, A.; KASHEF, R.; SHAKER, A. Deep learning for lidar point cloud classification in remote sensing. **Sensors**, MDPI, v. 22, n. 20, p. 7868, 2022.
- ENGEL, N.; BELAGIANNIS, V.; DIETMAYER, K. Point transformer. **IEEE Access**, Institute of Electrical and Electronics Engineers (IEEE), v. 9, p. 134826–134840, 2021. ISSN 2169-3536. Disponível em: <http://dx.doi.org/10.1109/ACCESS.2021.3116304>.
- FLETCHER, S.; ISLAM, M. Z. *et al.* Comparing sets of patterns with the jaccard index. **Australasian Journal of Information Systems**, Australian Computer Society, v. 22, 2018.
- FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. **Journal of the american statistical association**, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

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

GRANDINI, M.; BAGLI, E.; VISANI, G. Metrics for multi-class classification: an overview. **arXiv preprint arXiv:2008.05756**, 2020.

GUO, M.-H.; CAI, J.-X.; LIU, Z.-N.; MU, T.-J.; MARTIN, R. R.; HU, S.-M. Pct: Point cloud transformer. **Computational Visual Media**, Springer, v. 7, p. 187–199, 2021.

HAMEED, Z.; ZAHIA, S.; GARCIA-ZAPIRAIN, B.; AGUIRRE, J. J.; VANEGAS, A. M. Breast cancer histopathology image classification using an ensemble of deep learning models. **Sensors**, v. 20, n. 16, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/16/4373>.

HARRIS, J. E.; BOUSHEY, C.; BRUEMMER, B.; ARCHER, S. L. Publishing nutrition research: a review of nonparametric methods, part 3. **Journal of the American Dietetic Association**, Elsevier, v. 108, n. 9, p. 1488–1496, 2008.

HAYKIN, S. **Redes neurais: princípios e prática**. [S. l.]: Bookman Editora, 2001.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. **International conference on machine learning**. [S. l.], 2015. p. 448–456.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American statistical Association**, Taylor & Francis, v. 47, n. 260, p. 583–621, 1952.

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

LECUN, Y.; BOSER, B.; DENKER, J.; HENDERSON, D.; HOWARD, R.; HUBBARD, W.; JACKEL, L. Handwritten digit recognition with a back-propagation network. **Advances in neural information processing systems**, v. 2, 1989.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LI, Y.; BU, R.; SUN, M.; WU, W.; DI, X.; CHEN, B. Pointcnn: Convolution on x-transformed points. **Advances in neural information processing systems**, v. 31, 2018.

LIU, Z.; TANG, H.; LIN, Y.; HAN, S. Point-voxel cnn for efficient 3d deep learning. **Advances in neural information processing systems**, v. 32, 2019.

LUO, P.; WANG, X.; SHAO, W.; PENG, Z. Towards understanding regularization in batch normalization. **arXiv preprint arXiv:1809.00846**, 2018.

MA, C.; MENG, W.; WU, B.; XU, S.; ZHANG, X. Efficient joint gradient based attack against sor defense for 3d point cloud classification. In: **Proceedings of the 28th ACM International Conference on Multimedia**. [S. l.: s. n.], 2020. p. 1819–1827.

MA, C.; MENG, W.; WU, B.; XU, S.; ZHANG, X. Towards effective adversarial attack against 3d point cloud classification. In: IEEE. **2021 IEEE International Conference on Multimedia and Expo (ICME)**. [S. l.], 2021. p. 1–6.

- MEYES, R.; LU, M.; PUISEAU, C. W. de; MEISEN, T. Ablation studies in artificial neural networks. **arXiv preprint arXiv:1901.08644**, 2019.
- MONTAHA, S.; AZAM, S.; RAFID, A. K. M. R. H.; GHOSH, P.; HASAN, M. Z.; JONKMAN, M.; BOER, F. D. Breastnet18: A high accuracy fine-tuned vgg16 model evaluated using ablation study for diagnosing breast cancer from enhanced mammography images. **Biology**, v. 10, n. 12, 2021. ISSN 2079-7737. Disponível em: <https://www.mdpi.com/2079-7737/10/12/1347>.
- MOON, Y.; LEE, J.; BEAK, S.; JEONG, J. Point cloud-based real-time 3d object detection for predictive analytics of safety incidents in manufacturing industry. In: **2023 29th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)**. [S. l.: s. n.], 2023. p. 1–6.
- MURRAY, E. A. What have ablation studies told us about the neural substrates of stimulus memory? In: ELSEVIER. **Seminars in Neuroscience**. [S. l.], 1996. v. 8, n. 1, p. 13–22.
- MUZAHID, A.; WAN, W.; SOHEL, F.; WU, L.; HOU, L. Curvenet: Curvature-based multitask learning deep networks for 3d object recognition. **IEEE/CAA Journal of Automatica Sinica**, IEEE, v. 8, n. 6, p. 1177–1187, 2020.
- NANDA, A.; MOHAPATRA, B. B.; MAHAPATRA, A. P. K.; MAHAPATRA, A. P. K.; MAHAPATRA, A. P. K. Multiple comparison test by tukey's honestly significant difference (hsd): Do the confident level control type i error. **International Journal of Statistics and Applied Mathematics**, AkiNik Publications, v. 6, n. 1, p. 59–65, 2021.
- O'SHEA, K.; NASH, R. **An Introduction to Convolutional Neural Networks**. 2015.
- PRITT, M.; CHERN, G. Satellite image classification with deep learning. In: **2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)**. [S. l.: s. n.], 2017. p. 1–7.
- QI, C. R.; SU, H.; MO, K.; GUIBAS, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2017. p. 652–660.
- QI, C. R.; YI, L.; SU, H.; GUIBAS, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. **Advances in neural information processing systems**, v. 30, 2017.
- RASAMOELINA, A. D.; ADJAILIA, F.; SINČÁK, P. A review of activation function for artificial neural network. In: IEEE. **2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)**. [S. l.], 2020. p. 281–286.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mccllelland. vol. 1. 1986. **Biometrika**, v. 71, n. 599-607, p. 6, 1986.
- SCHERER, D.; MÜLLER, A.; BEHNKE, S. Evaluation of pooling operations in convolutional architectures for object recognition. In: SPRINGER. **International conference on artificial neural networks**. [S. l.], 2010. p. 92–101.

- SONATA, I.; HERYADI, Y.; LUKAS, L.; WIBOWO, A. Autonomous car using cnn deep learning algorithm. **Journal of Physics: Conference Series**, IOP Publishing, v. 1869, n. 1, p. 012071, apr 2021. Disponível em: <https://dx.doi.org/10.1088/1742-6596/1869/1/012071>.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, n. 56, p. 1929–1958, 2014. Disponível em: <http://jmlr.org/papers/v15/srivastava14a.html>.
- ST»HLE, L.; WOLD, S. Analysis of variance (anova). **Chemometrics and Intelligent Laboratory Systems**, v. 6, n. 4, p. 259–272, 1989. ISSN 0169-7439. Disponível em: <https://www.sciencedirect.com/science/article/pii/0169743989800954>.
- THOMAS, H.; QI, C. R.; DESCHAUD, J.-E.; MARCOTEGUI, B.; GOULETTE, F.; GUIBAS, L. J. Kpconv: Flexible and deformable convolution for point clouds. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S. l.: s. n.], 2019. p. 6411–6420.
- UY, M. A.; PHAM, Q.-H.; HUA, B.-S.; NGUYEN, T.; YEUNG, S.-K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S. l.: s. n.], 2019. p. 1588–1597.
- WANG, Q. Automatic checks from 3d point cloud data for safety regulation compliance for scaffold work platforms. **Automation in Construction**, v. 104, p. 38–51, 2019. ISSN 0926-5805. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0926580519300019>.
- WANG, Y.; SUN, Y.; LIU, Z.; SARMA, S. E.; BRONSTEIN, M. M.; SOLOMON, J. M. Dynamic graph cnn for learning on point clouds. **ACM Transactions on Graphics (tog)**, Acm New York, NY, USA, v. 38, n. 5, p. 1–12, 2019.
- WANG, Z.; XU, Y.; HE, Q.; FANG, Z.; XU, G.; FU, J. Grasping pose estimation for scara robot based on deep learning of point cloud. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 108, p. 1217–1231, 2020.
- WEBER, M.; WILD, D.; KLEESIEK, J.; EGGER, J.; GSAXNER, C. Deep learning-based point cloud registration for augmented reality-guided surgery. **arXiv preprint arXiv:2405.03314**, 2024.
- WIERSMA, R.; NASIKUN, A.; EISEMANN, E.; HILDEBRANDT, K. Deltaconv: anisotropic operators for geometric deep learning on point clouds. **ACM Transactions on Graphics (TOG)**, ACM New York, NY, USA, v. 41, n. 4, p. 1–10, 2022.
- WU, C.; ZHENG, J.; PFROMMER, J.; BEYERER, J. Attention-based point cloud edge sampling. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S. l.: s. n.], 2023. p. 5333–5343.
- WU, W.; QI, Z.; FUXIN, L. Pointconv: Deep convolutional networks on 3d point clouds. In: **Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition**. [S. l.: s. n.], 2019. p. 9621–9630.
- WU, Z.; SONG, S.; KHOSLA, A.; YU, F.; ZHANG, L.; TANG, X.; XIAO, J. 3d shapenets: A deep representation for volumetric shapes. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S. l.: s. n.], 2015. p. 1912–1920.

XU, B.; WANG, N.; CHEN, T.; LI, M. Empirical evaluation of rectified activations in convolutional network. **arXiv preprint arXiv:1505.00853**, 2015.

XU, M.; DING, R.; ZHAO, H.; QI, X. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S. l.: s. n.], 2021. p. 3173–3182.

XU, X.; ZHANG, L.; YANG, J.; CAO, C.; WANG, W.; RAN, Y.; TAN, Z.; LUO, M. A review of multi-sensor fusion slam systems based on 3d lidar. **Remote Sensing**, MDPI, v. 14, n. 12, p. 2835, 2022.

XU, Y.; FAN, T.; XU, M.; ZENG, L.; QIAO, Y. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: **Proceedings of the European conference on computer vision (ECCV)**. [S. l.: s. n.], 2018. p. 87–102.

YAN, X.; ZHENG, C.; LI, Z.; WANG, S.; CUI, S. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S. l.: s. n.], 2020. p. 5589–5598.

YUAN, M.; FU, K.; LI, Z.; WANG, M. Decoupled deep hough voting for point cloud registration. **Frontiers of Computer Science**, Springer, v. 18, n. 2, p. 182703, 2024.

ZHANG, H.; WANG, C.; TIAN, S.; LU, B.; ZHANG, L.; NING, X.; BAI, X. Deep learning-based 3d point cloud classification: A systematic survey and outlook. **Displays**, Elsevier, p. 102456, 2023.

ZHANG, J.; XIE, Y.; WU, Q.; XIA, Y. Medical image classification using synergic deep learning. **Medical image analysis**, Elsevier, v. 54, p. 10–19, 2019.

ZHANG, J.; ZHAO, X.; CHEN, Z.; LU, Z. A review of deep learning-based semantic segmentation for point cloud. **IEEE Access**, v. 7, p. 179118–179133, 2019.

ZHAO, H.; JIANG, L.; JIA, J.; TORR, P. H.; KOLTUN, V. Point transformer. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. [S. l.: s. n.], 2021. p. 16259–16268.

ZHU, Q.; FAN, L.; WENG, N. Advancements in point cloud data augmentation for deep learning: A survey. **Pattern Recognition**, Elsevier, p. 110532, 2024.