
Análise de classificadores aplicados às projeções na Garrafa de Klein

Carlos Humberto Martins Junior



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Uberlândia - MG
2024

Carlos Humberto Martins Junior

**Análise de classificadores aplicados às projeções
na Garrafa de Klein**

Trabalho de Conclusão de Curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau de
Bacharel em Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Prof. Dr. Thiago Pirola Ribeiro

Uberlândia - MG

2024

Este trabalho é dedicado principalmente a minha mãe Katiuce que me deu grande apoio no início da minha jornada, meu grande amigo Paulo Henrique que sempre esteve ao meu lado, minha namorada Izabela que é minha companheira, ao meu irmão Rafael, minha avó Valda e ao meu falecido avô Jair.

Agradecimentos

Agradeço a minha mãe pelo início da minha jornada na graduação.

Agradeço aos meus amigos e colegas do curso, Paulo Henrique, Matheus Costa, Mateus Rosa, Lucas Gabriel e Davide Sgalambro, que me proporcionaram grandes momentos juntos.

À minha namorada, Izabela Brasiel, tenho muito que agradecer pela ajuda e pela motivação para ir até o final.

Agradeço ao meu irmão, Rafael Humberto, que é uma pessoa muito inteligente.

Agradeço aos meus avós maternos por todo o carinho e apoio.

Agradeço ao meu orientador, Prof. Dr. Thiago Pirola, pela paciência, orientação e ajuda durante o trabalho.

Agradeço à Faculdade de Computação (FACOM) e à Universidade Federal de Uberlândia (UFU) pela oportunidade de cursar Sistemas de Informação, que para mim foi algo além de um curso, e sim uma experiência de vida.

"O conhecimento que se adquire com o sofrimento é o mais duradouro."
(Plato)

Resumo

Este trabalho propõe o estudo e a implementação de técnicas avançadas para a caracterização de texturas artificiais e naturais em imagens digitais, adaptando as técnicas de Ribeiro, Oliveira e Barcelos (2018) ao contexto das projeções no espaço da garrafa de Klein. Foram aplicados novos classificadores, incluindo modelos não supervisionados como KNN, K-Means Clustering, modelos de Deep Learning. O trabalho também abordou técnicas de Aprendizado por Reforço, como Q-Learning, LabelPropagation, MLPClassifier.

Algumas Bases de Imagens públicas foram testadas e foram obtidos resultados, por exemplo, na análise detalhada dos experimentos e das métricas revelou o desempenho dos diferentes algoritmos, destacando as vantagens e desvantagens de cada abordagem em contextos específicos. Os modelos de *Machine Learning* propostos (Rede Neural Convolutiva (CNN), *K-Nearest Neighbors* (KNN) e Classificador baseado em Redes Neurais Multicamadas (MLP-Classifer)) atingindo acurácias de 87,02% na Base CURET e 88,34% na Base KTH-TIPS.

Durante as análises dos resultados obtidos, obteve-se *insights* sobre a aplicação de algoritmos de *machine learning* para problemas de classificação em imagens. Pretende-se aplicar IA Generativa e Modelos de Fine-Tuning para melhoria dos resultados.

Palavras-chave: Texturas, Garrafa de Klein, Aprendizado de Máquina, Deep Learning, Convolutional Neural Networks.

Lista de ilustrações

Figura 1 – Representação visual da garrafa de Klein obtida por meio do MathWorks MATLAB® pelo comando <code>xpklein</code> , extraída por (PEREA; CARLSSON, 2014b).	20
Figura 2 – Diagrama ilustrando a relação entre Inteligência Artificial e <i>Machine Learning</i>	22
Figura 3 – Estrutura de um modelo supervisionado com um conjunto de variáveis de entrada ($X_1, X_2, \dots, X_p$) e a variável-alvo (Y), que está separada das entradas e rotulada como <i>Target</i> . O objetivo do modelo é aprender a mapear as variáveis de entrada para a variável-alvo.	23
Figura 4 – Estrutura de um modelo não supervisionado com múltiplas variáveis de entrada ($X_1, X_2, \dots, X_p$), mas sem a presença de um <i>Target</i> (variável-alvo). O objetivo do modelo é identificar agrupamentos ou padrões nos dados com base nas similaridades entre as entradas.	24
Figura 5 – Ilustração de um jogo de ping-pong, que pode ser utilizado como analogia ao aprendizado por reforço. Nesse contexto, o agente (um dos jogadores) realiza ações (como rebater a bola) em resposta ao estado atual do jogo, recebendo recompensas (ganhar pontos) com o objetivo de otimizar seu desempenho ao longo das interações.	26
Figura 6 – Fluxo de um modelo de <i>Deep Learning</i> aplicado à classificação de imagens. O input (uma imagem de carro) passa por uma rede neural convolucional, composta por estágios de extração de características e classificação. O output final indica se a imagem representa um carro ou não.	27

Figura 7 – Comparação entre modelos supervisionado, semi-supervisionado e não supervisionado. O modelo supervisionado (esquerda) possui todos os dados rotulados, onde as bolinhas azuis estão à esquerda e as amarelas à direita. O modelo semi-supervisionado (meio) tem predominantemente bolinhas cinzas (não rotuladas), com algumas bolinhas rotuladas (azuis à esquerda e amarelas à direita). O modelo não supervisionado (direita) trabalha apenas com bolinhas cinzas, ou seja, dados sem rótulos. . . .	28
Figura 8 – Diagrama representando a abordagem de Aprendizado por Transferência, onde um classificador MLP é utilizado para categorizar imagens de gatos e cachorros.	29
Figura 9 – Saída da curva ROC do Modelo CNN, Base BrodaTz	45
Figura 10 – Saída da curva ROC do Modelo KNN, Base BrodaTz	46
Figura 11 – Saída da curva ROC do Modelo MLP, Base BrodaTz	46
Figura 12 – Saída da curva ROC do Modelo LabelPropagation, Base BrodaTz . . .	46
Figura 13 – Saída da curva ROC do Modelo QLearning, Base BrodaTz	46
Figura 14 – Saída da Matriz de Confusão do Modelo KMeans para a Base Brodatz. A matriz exibe a correspondência entre os clusters verdadeiros (eixo vertical) e os clusters previstos (eixo horizontal). A intensidade da cor representa a frequência de ocorrências para cada par de valores, conforme indicado pela barra de cores à direita. Valores mais escuros indicam maior número de classificações em um determinado cluster previsto.	47
Figura 15 – Saída da curva ROC do Modelo CNN, Base CURET	48
Figura 16 – Saída da curva ROC do Modelo KNN, Base CURET	49
Figura 17 – Saída da curva ROC do Modelo MLP, Base CURET	49
Figura 18 – Saída da curva ROC do Modelo LabelPropagation, Base CURET . . .	49
Figura 19 – Saída da curva ROC do Modelo Q Learning, Base CURET	49
Figura 20 – Saída da Matriz de Confusão do Modelo KMEANS, Base CURET . . .	50
Figura 21 – Saída da curva ROC do Modelo CNN, Base KTH-TIPS	51
Figura 22 – Saída da curva ROC do Modelo KNN, Base KTH-TIPS	51
Figura 23 – Saída da curva ROC do Modelo MLP, Base KTH-TIPS	51
Figura 24 – Saída da curva ROC do Modelo LabelPropagation, Base KTH-TIPS . .	51
Figura 25 – Saída da curva ROC do Modelo Q Learning, Base KTH-TIPS	51
Figura 26 – Saída da Matriz de Confusão do Modelo KMEANS, Base KTH-TIPS .	52
Figura 27 – Saída da curva ROC do Modelo CNN, Base Outex	53
Figura 28 – Saída da curva ROC do Modelo KNN, Base Outex	54
Figura 29 – Saída da curva ROC do Modelo MLP, Base Outex	54
Figura 30 – Saída da curva ROC do Modelo LabelPropagation, Base Outex	54
Figura 31 – Saída da curva ROC do Modelo Q Learning, Base Outex	54

Figura 32 – Saída da Matriz de Confusão do Modelo KMEANS, Base Outex	55
Figura 33 – Saída da curva ROC do Modelo CNN, Base Sport Event	56
Figura 34 – Saída da curva ROC do Modelo KNN, Base Sport Event	56
Figura 35 – Saída da curva ROC do Modelo MLP, Base Sport Event	56
Figura 36 – Saída da curva ROC do Modelo Q Learning, Base Sport Event	56
Figura 37 – Saída da curva ROC do Modelo LabelPropagation, Base Sport Event .	56
Figura 38 – Saída da Matriz de Confusão do Modelo KMEANS, Base Sport Event .	57
Figura 39 – Saída da curva ROC do Modelo CNN, Base UIUCTex	58
Figura 40 – Saída da curva ROC do Modelo KNN, Base UIUCTex	59
Figura 41 – Saída da curva ROC do Modelo MLP, Base UIUCTex	59
Figura 42 – Saída da curva ROC do Modelo LabelPropagation, Base UIUCTex . .	59
Figura 43 – Saída da curva ROC do Modelo Q Learning, Base UIUCTex	59
Figura 44 – Saída da Matriz de Confusão do Modelo KMEANS, Base UIUCTex . .	60
Figura 45 – Saída da curva ROC do Modelo CNN, Base Vistex	61
Figura 46 – Saída da curva ROC do Modelo KNN, Base Vistex	62
Figura 47 – Saída da curva ROC do Modelo MLP, Base Vistex	62
Figura 48 – Saída da curva ROC do Modelo LabelPropagation, Base Vistex	62
Figura 49 – Saída da curva ROC do Modelo Q Learning, Base Vistex	62
Figura 50 – Saída da Matriz de Confusão do Modelo KMEANS, Base Vistex	63

Lista de tabelas

Tabela 1 – Matriz de Confusão	43
Tabela 2 – Resultados dos Modelos de Machine Learning - BrodaTz	45
Tabela 3 – Resultados do K-MEANS Base BrodaTz	45
Tabela 4 – Resultados dos Modelos de Machine Learning - CureT	47
Tabela 5 – Resultados do K-MEANS Base CURET	48
Tabela 6 – Resultados dos Modelos de Machine Learning - KTH-TIPS	49
Tabela 7 – Resultados do K-MEANS Base KTH-TIPS	50
Tabela 8 – Resultados dos Modelos de Machine Learning - OUTEX	52
Tabela 9 – Resultados do K-MEANS Base Outex	53
Tabela 10 – Resultados dos Modelos de Machine Learning - Sport Event	55
Tabela 11 – Resultados do K-MEANS Base Sport Event	55
Tabela 12 – Resultados dos Modelos de Machine Learning - UIUCTex	57
Tabela 13 – Resultados do K-MEANS Base UIUCTex	58
Tabela 14 – Resultados dos Modelos de Machine Learning - Vistex	59
Tabela 15 – Resultados do K-MEANS Base Vistex	60
Tabela 16 – Resultados dos Métodos Propostos e de Métodos da Literatura para as Seis Bases de Imagens.	61

Lista de siglas

ARI *Adjusted Rand Index*

CNN Rede Neural Convolucional - *Convolutional Neural Network*

CPAD Controlador Paralelo com Alto Desempenho

CUReT *Columbia-Utrecht Reflectance and Texture Database*

FN Falsos Negativos - *False Negative*

FP Falsos Positivos - *False Positive*

FPR Taxa de Falsos Positivos - *False Positive Rate*

GANs *Generative Adversarial Networks* - Redes Adversariais Generativas

GSN *Goal Seeking Neuron*

IA Inteligência Artificial - *Artificial intelligence*

KNN *K-Nearest Neighbors* - *K-Vizinhos mais Próximos*

LLMs Modelos de Linguagem de Grande Escala - *Large Language Model*

ML *Machine Learning* - Aprendizado de Máquina

MLP-Classifer Classificador baseado em Redes Neurais Multicamadas - *Multilayer Perceptron Classifier*

NMI *Normalized Mutual Information*

ROC *Receiver Operating Characteristic*

TN Verdadeiros Negativos - *True Negative*

TPR Taxa de Verdadeiros Positivos - *True Positive Rate*

TP Verdadeiros Positivos - *True Positive*

TCC Trabalho de Conclusão de Curso

VisTex *The Vision Texture*

Sumário

1	INTRODUÇÃO	15
1.1	Motivação	16
1.2	Problema	16
1.3	Hipótese	17
1.4	Objetivos	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Garrafa de Klein	19
2.2	Python	20
2.2.1	Numpy	21
2.2.2	Matplotlib	21
2.2.3	OpenCV	21
2.2.4	Scikit-learn	21
2.2.5	TensorFlow	21
2.3	Inteligência Artificial e Machine Learning	22
2.3.1	Modelos Supervisionados	23
2.3.2	Modelos Não Supervisionados	24
2.3.3	Modelos de Aprendizado por Reforço	25
2.3.4	Modelos de Deep Learning	26
2.3.5	Modelos de Aprendizado Semi-Supervisionado	28
2.3.6	Modelos de Aprendizado por Transferência	29
2.4	Processamento de Imagens	30
2.5	Trabalhos Correlatos	31
2.5.1	Considerações Finais	32
3	METODOLOGIA	34
3.1	Experimentos e Análise dos Resultados	34
3.1.1	Pseudocódigo para Treinamento de Modelos de Aprendizado de Máquina	34

3.2	Reimplementação do Algoritmo em Python	35
3.3	Aplicação de Modelos de Aprendizado de Máquina	36
3.3.1	Modelos Supervisionados	36
3.3.2	Modelos Não Supervisionados	36
3.3.3	Modelos de Deep Learning	36
3.3.4	Modelos de Aprendizado por Reforço	36
3.3.5	Modelos de Aprendizado Semi-Supervisionado	36
3.3.6	Modelos de Aprendizado por Transferência	36
3.3.7	Conjunto de Parâmetros	37
3.4	Bases utilizadas	37
3.5	Análise dos Resultados	39
3.6	Método para a Avaliação	39
3.7	Experimentos e Avaliação dos Resultados	44
3.7.1	Brodatz	44
3.7.2	Curet	46
3.7.3	KTH-TIPS	48
3.7.4	OUTEX	50
3.7.5	Sport Event	53
3.7.6	UIUCTex	54
3.7.7	Vistex	58
4	CONCLUSÃO	64
4.1	Principais Contribuições	64
4.2	Trabalhos Futuros	65
4.3	Contribuições em Produção Bibliográfica	67
	REFERÊNCIAS	68

Introdução

Este trabalho propõe o estudo e implementação de técnicas avançadas para caracterizar texturas artificiais e naturais em imagens digitais, com base na modificação e aprimoramento das técnicas apresentadas em Ribeiro, Oliveira e Barcelos (2018). A proposta centra-se na aplicação dessas técnicas no contexto das Projeções no Espaço da Garrafa de Klein, uma topologia que oferece propriedades geométricas e matemáticas favoráveis à análise de texturas. O objetivo é adaptar as projeções e explorar novos classificadores estatísticos e de redes neurais aplicados ao vetor de características previamente definido pelas projeções. Essas melhorias visam aumentar a precisão da classificação e o reconhecimento de texturas, aproveitando tanto a natureza estatística quanto a capacidade de redes neurais em identificar padrões complexos.

A análise de texturas em imagens tem sido uma área de pesquisa importante, especialmente devido às suas aplicações em diversas áreas, como reconhecimento de padrões, análise de imagens médicas e segmentação de superfícies. O trabalho de RIBEIRO (2019) proporcionou uma abordagem inicial significativa ao tratar da obtenção e seleção de patches de textura, porém, com os avanços recentes em aprendizado de máquina, é possível aprimorar esses métodos e obter resultados superiores. A motivação para este trabalho reside na busca por técnicas modernas que possam otimizar e melhorar o desempenho dos classificadores.

A introdução de novos classificadores é um ponto central desta pesquisa. Embora o vetor de características definido por Ribeiro, Oliveira e Barcelos (2018) já tenha apresentado bons resultados, o uso de técnicas mais modernas, como Redes Neurais Convolucionais (*Convolutional Neural Networks* - *CNN*) e outros modelos de aprendizado de máquina, pode gerar uma melhoria significativa na precisão da classificação. As CNNs, por exemplo, são amplamente conhecidas por sua capacidade de identificar padrões em imagens de forma eficiente, o que as torna uma escolha promissora para a análise de texturas. Além disso, serão explorados classificadores supervisionados, não supervisionados, semi-supervisionados e de aprendizado por transferência, garantindo uma abordagem ampla e completa para o problema de classificação.

1.1 Motivação

A motivação principal deste trabalho é aprimorar os métodos desenvolvidos por Ribeiro, Oliveira e Barcelos (2018), integrando técnicas modernas de aprendizado de máquina e inteligência artificial para melhorar a precisão da classificação de texturas. O uso de classificadores mais avançados, como redes neurais, permitirá uma análise mais detalhada e precisa, captando características mais sutis e complexas presentes nas texturas artificiais e naturais, contudo a necessidade de desenvolver modelos mais robustos e interpretáveis para prever padrões complexos é um dos principais obstáculos e motivadores dessa pesquisa.

Além disso, a crescente importância da análise de texturas em diversos domínios, como medicina, engenharia de materiais e visão computacional, reforça a necessidade de desenvolver técnicas mais eficazes para a caracterização dessas texturas. A combinação de métodos estatísticos tradicionais com técnicas de aprendizado profundo abre novas possibilidades para o aprimoramento dos resultados já obtidos.

1.2 Problema

O problema central a ser enfrentado neste trabalho está relacionado à melhoria da eficácia dos classificadores utilizados na análise de texturas. Embora o vetor de características desenvolvido por Ribeiro, Oliveira e Barcelos (2018) já tenha fornecido uma boa base, o avanço das técnicas de aprendizado de máquina permite a exploração de novas abordagens para a classificação de texturas. Entre os classificadores que serão implementados e testados, destacam-se:

- ❑ Modelos Supervisionados: como o *K-Nearest Neighbors (KNN)*, que classifica dados com base na proximidade entre os exemplos, sendo útil em diversos problemas de classificação.
- ❑ Modelos Não Supervisionados: como o *K-Means Clustering*, que agrupa os dados em clusters, permitindo a descoberta de padrões ocultos.
- ❑ Modelos de Deep Learning: em especial as *Convolutional Neural Networks (CNNs)*, amplamente utilizadas no processamento de imagens e reconhecimento de padrões visuais.
- ❑ Modelos de Aprendizado por Reforço: como o *Q-Learning*, que otimiza uma função de recompensa para tarefas sequenciais.
- ❑ Modelos Semi-Supervisionados: como o *LabelPropagation*, que utiliza dados rotulados para propagar informações para dados não rotulados.

- ❑ Modelos de Aprendizado por Transferência: como o *MLPClassifier*, que transfere conhecimento entre diferentes problemas, acelerando o processo de classificação.

A inclusão de métodos de aprendizado de máquina avançados visa aumentar a eficácia do reconhecimento de texturas, fornecendo uma solução mais robusta e precisa.

1.3 Hipótese

A hipótese deste trabalho é que a introdução de novos classificadores e técnicas de aprendizado de máquina resultará em uma melhoria significativa na precisão da classificação de texturas artificiais e naturais. A utilização de redes neurais convolucionais e métodos modernos de aprendizado supervisionado e não supervisionado possibilitará a identificação de padrões mais complexos e a obtenção de melhores resultados em comparação aos métodos tradicionais.

A adaptação das Projeções no Espaço da Garrafa de Klein também proporcionará uma nova perspectiva na análise de texturas, permitindo explorar propriedades geométricas que ainda não foram amplamente estudadas nesse contexto. A expectativa é que essas modificações melhorem a capacidade de separação entre diferentes classes de texturas, resultando em uma classificação mais precisa.

1.4 Objetivos

O principal objetivo deste trabalho é aprimorar o algoritmo de classificação desenvolvido por Ribeiro, Oliveira e Barcelos (2018), incorporando técnicas modernas de aprendizado de máquina para melhorar a precisão na classificação de texturas em imagens digitais. Para atingir esse objetivo, será necessário:

- ❑ Explorar e implementar uma variedade de modelos de aprendizado de máquina, incluindo classificadores supervisionados, não supervisionados, semi-supervisionados e de aprendizado profundo.
- ❑ Comparar o desempenho dos diferentes classificadores e selecionar aqueles que oferecem os melhores resultados na classificação de texturas.
- ❑ Adaptar o modelo de Ribeiro, Oliveira e Barcelos (2018), integrando as Projeções no Espaço da Garrafa de Klein, e testar sua eficácia na melhoria da separação entre diferentes classes de texturas.

A expectativa é que a implementação desses métodos resulte em uma solução eficiente e precisa, capaz de classificar uma ampla gama de texturas em diferentes aplicações. Ao final do projeto, espera-se obter uma solução inovadora que possa ser aplicada em

diversos cenários, contribuindo significativamente para a área de análise de imagens e reconhecimento de padrões visuais.

Fundamentação Teórica

A seção de Fundamentação Teórica do trabalho aborda dois tópicos principais: a Garrafa de Klein e a linguagem de programação Python. A Garrafa de Klein é apresentada como uma superfície matemática não orientável, cujas características são exploradas e relacionadas a outras estruturas semelhantes. Em seguida, é introduzida uma nova abordagem para a caracterização de texturas artificiais e naturais em imagens digitais, utilizando projeções topológicas na Garrafa de Klein.

Quanto à linguagem Python, destaca-se sua ênfase na simplicidade e legibilidade do código, além de sua ampla disponibilidade de bibliotecas, que a tornam popular em diversas áreas, incluindo desenvolvimento web, análise de dados e aprendizado de máquina.

2.1 Garrafa de Klein

Em matemática, a garrafa de Klein é um exemplo de uma superfície não orientável; informalmente, ela é uma superfície (uma variedade bidimensional) em que as noções de direita, esquerda, cima, baixo, dentro e fora não podem ser definidas de maneira consistente. Entre as estruturas relacionadas que também não são orientáveis estão incluídos o plano projetivo real e a faixa de Möbius. Enquanto uma faixa de Möbius é uma superfície com borda, uma garrafa de Klein não possui borda (a título de comparação, uma esfera é uma superfície orientável sem borda). Uma garrafa de Klein é um espaço topológico obtido pela colagem de duas fitas de Möbius (Figura 1) (GARFUNKEL, 2021).

Neste trabalho, apresenta-se uma abordagem para a caracterização de imagens baseadas em texturas através de projeções topológicas na garrafa de Klein de pequenas regiões de alto contraste (patches) extraídas das imagens. Várias configurações de tamanhos de patch foram analisadas a fim de reduzir o tamanho do vetor de recursos e aumentar a precisão. Experimentos usando o método proposto para classificação de textura, em vários conjuntos de dados estabelecidos, mostram que o método proposto não apenas gerencia reduzir o tamanho do vetor de características, mas também melhora as taxas de

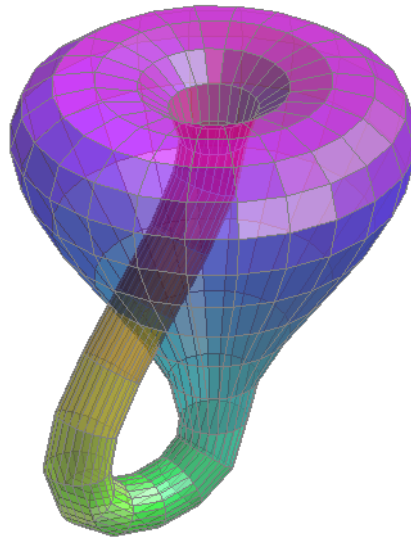


Figura 1 – Representação visual da garrafa de Klein obtida por meio do MathWorks MATLAB® pelo comando `xpklein`, extraída por (PEREA; CARLSSON, 2014b).

classificação correta quando comparadas a outros métodos de última geração (RIBEIRO; OLIVEIRA; BARCELOS, 2018; RIBEIRO, 2019).

2.2 Python

A linguagem **Python** foi projetada para enfatizar a produtividade do desenvolvedor, facilitando a escrita de código limpo, simples e legível, tanto em aplicações menores quanto em programas mais complexos. Ela oferece tipagem dinâmica, suporte à orientação a objetos, multiparadigmas e uma vasta coleção de bibliotecas poderosas. Seu código é aberto e de uso gratuito, podendo ser executado em praticamente qualquer sistema operacional. Neste trabalho, utilizamos diversas bibliotecas instaláveis do Python (Python Software Foundation, 2023).

Um dos maiores atrativos do Python¹ é a grande quantidade de bibliotecas disponíveis, tanto nativas quanto de terceiros, o que o torna amplamente utilizado em áreas como desenvolvimento web, análise de dados, aprendizado de máquina e Inteligência Artificial. Grandes empresas como Google, Meta, Netflix, Spotify e Globo utilizam **Python** em conjunto com outras linguagens e tecnologias, destacando-se pela sua versatilidade. Além disso, cerca de 16% de todas as perguntas no Stack Overflow estão marcadas como “Python”, facilitando a resolução rápida de problemas (KRIGER, 2022).

¹ Mais informações no site <<https://www.python.org>>.

2.2.1 Numpy

A biblioteca NumPy² foi desenvolvida para facilitar o trabalho com matrizes e arrays em Python, superando a complexidade de manipular listas de listas. Especialmente na área de processamento de imagens, o NumPy oferece funções eficientes para manipulação rápida de dados, incluindo arrays e matrizes. Além disso, suas funções são otimizadas, uma vez que grande parte do código é escrita na linguagem C, que é de baixo nível e tipada, proporcionando um desempenho superior (NumPy Developers, 2023).

2.2.2 Matplotlib

Matplotlib³ é uma biblioteca essencial em Python para a criação de gráficos e visualizações de dados. Nesta pesquisa, foi utilizada para gerar visualizações como a matriz de confusão e gráficos de linha, incluindo a curva ROC. A biblioteca se destaca por sua versatilidade e simplicidade, facilitando a comparação e análise visual dos diferentes resultados obtidos nos experimentos (MATPLOTLIB, 2022).

2.2.3 OpenCV

OpenCV⁴ é uma biblioteca de código aberto amplamente utilizada para processamento de imagens e visão computacional. Ela permite o processamento de imagens e vídeos para identificar objetos, rostos ou até mesmo reconhecer escrita humana (OPENCV, 2022).

2.2.4 Scikit-learn

Scikit-learn⁵ é uma biblioteca de aprendizado de máquina de código aberto que oferece uma ampla gama de algoritmos de classificação, regressão e agrupamento. Entre os algoritmos disponíveis estão máquinas de vetores de suporte, florestas aleatórias, *gradient boosting*, *k-means* e DBSCAN. Essa biblioteca será utilizada para deixar o código mais inteligente, buscando melhores resultados para o algoritmo proposto (LEARN, 2022).

2.2.5 TensorFlow

TensorFlow⁶ é uma biblioteca de aprendizado de máquina voltada principalmente para *deep learning* e redes neurais. Diferentemente do **Scikit-learn**, que é mais genérica, o **TensorFlow** é mais utilizado em aplicações que envolvem redes neurais profundas, com foco em detectar padrões complexos e correlações. A combinação dessas duas bibliotecas

² Link da Biblioteca Numpy <<https://numpy.org>>

³ Link da Biblioteca Matplotlib <<https://matplotlib.org>>

⁴ Link da Biblioteca OpenCV <<https://opencv.org>>

⁵ Link da Biblioteca Scikit-learn <<https://scikit-learn.org>>

⁶ Link da Biblioteca TensorFlow <<https://www.tensorflow.org>>

permitirá a realização de diversas análises, como avaliação de desempenho, processamento, uso de memória e tempo (TENSORFLOW, 2022).

2.3 Inteligência Artificial e Machine Learning

A Inteligência Artificial (IA) refere-se a sistemas ou máquinas que imitam a inteligência humana para executar tarefas e se aprimoram iterativamente com base nas informações que coletam. A IA está mais relacionada ao processo e à capacidade de análise e processamento de dados do que a um formato ou função específicos. Embora a IA seja frequentemente associada a robôs sofisticados, seu objetivo é complementar as habilidades humanas, melhorando a capacidade de análise e tomada de decisão em diversos domínios (RUSSELL; NORVIG, 2010).

Por sua vez, o *Machine Learning* (ML) é um subcampo da Inteligência Artificial que permite que sistemas aprendam com dados e melhorem seu desempenho de forma autônoma. Ele é amplamente utilizado em áreas como a saúde, através de dispositivos *wearables*, que monitoram dados de pacientes em tempo real (BISHOP, 2006). A combinação de Inteligência Artificial e *Machine Learning* permite desenvolver modelos capazes de realizar previsões precisas e eficientes, como no caso da classificação de imagens.

A Figura 2 ilustra a relação entre Inteligência Artificial e *Machine Learning*. O círculo maior representa a Inteligência Artificial como um campo abrangente que engloba diversas técnicas e abordagens. Dentro dele, o círculo menor destaca o *Machine Learning*, que é uma subárea da Inteligência Artificial focada no desenvolvimento de algoritmos que permitem que sistemas aprendam com dados. Essa relação mostra como o *Machine Learning* é uma das ferramentas utilizadas para implementar soluções de Inteligência Artificial (Dicionário Tec, 2022).

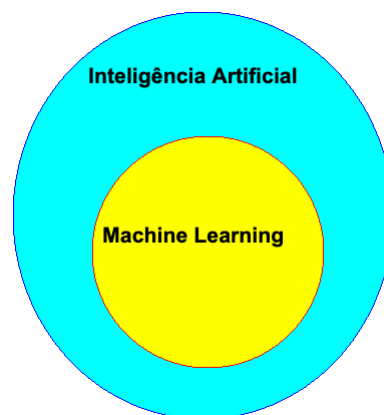


Figura 2 – Diagrama ilustrando a relação entre Inteligência Artificial e *Machine Learning*.

Fonte: Imagem retirada de (Dicionário Tec, 2022)

2.3.1 Modelos Supervisionados

Os modelos supervisionados são amplamente utilizados em problemas de classificação e regressão, nos quais o objetivo é prever uma variável-alvo com base em um conjunto de variáveis de entrada. Neste tipo de modelo, os dados de treinamento consistem em pares de entrada e saída, sendo que a saída é conhecida e utilizada para ajustar o modelo. Um exemplo clássico de modelo supervisionado é o algoritmo *K-Nearest Neighbors* (KNN), que foi aplicado neste trabalho para prever a classificação de imagens com base em vetores de características.

A Figura 3 ilustra a estrutura de um modelo supervisionado típico. As variáveis de entrada, denotadas como X_1 , X_2 , até X_p , representam as características que são utilizadas pelo modelo para aprender o comportamento do sistema. A variável-alvo, Y , também conhecida como *Target*, é o valor que o modelo supervisionado busca prever. Esse tipo de abordagem permite que o algoritmo seja treinado para generalizar padrões a partir dos dados rotulados.



Figura 3 – Estrutura de um modelo supervisionado com um conjunto de variáveis de entrada (X_1 , X_2 , ..., X_p) e a variável-alvo (Y), que está separada das entradas e rotulada como *Target*. O objetivo do modelo é aprender a mapear as variáveis de entrada para a variável-alvo.

Fonte: Imagem retirada de (SERVICES, 2023).

Durante os testes realizados com imagens, utilizou-se o algoritmo KNN para classificar os vetores de características extraídos das imagens. O KNN baseia-se na premissa de que as amostras semelhantes encontram-se próximas umas das outras no espaço de características. Assim, a classe de uma nova amostra é atribuída com base nas classes dos vizinhos mais próximos no conjunto de treinamento.

O algoritmo KNN demonstrou um bom desempenho na tarefa de classificação de imagens, especialmente em cenários onde os vetores de características apresentavam uma distribuição bem definida. Um dos principais pontos fortes desse método é a sua simplicidade, sendo fácil de implementar e compreender. No entanto, a precisão do KNN depende

fortemente da escolha do número de vizinhos (k) a serem considerados, o que exigiu ajustes finos para garantir os melhores resultados possíveis (FIX; JR., 1951; PEDREGOSA et al., 2011a).

Em suma, o KNN destacou-se como uma técnica eficaz para o problema de classificação de imagens neste contexto, proporcionando uma performance satisfatória desde que os parâmetros fossem cuidadosamente ajustados. A simplicidade e a eficiência computacional desse método o tornam uma escolha viável para problemas onde o número de dimensões não é excessivamente elevado.

2.3.2 Modelos Não Supervisionados

Modelos não supervisionados, ao contrário dos supervisionados, não dependem de dados rotulados durante o treinamento. Em vez disso, esses modelos buscam identificar padrões ocultos ou estruturas subjacentes nos dados com base nas variáveis de entrada. Um dos algoritmos mais amplamente utilizados em tarefas de agrupamento (*clustering*) é o *K-Means Clustering*.

Utilizar esse tipo de algoritmo para testar a capacidade de segmentação e identificação de padrões em texturas artificiais e naturais, ajuda a analisar sua eficácia na distinção entre diferentes classes de texturas sem a necessidade de rótulos pré-definidos.



Figura 4 – Estrutura de um modelo não supervisionado com múltiplas variáveis de entrada ($X_1, X_2, \dots, X_p$), mas sem a presença de um *Target* (variável-alvo). O objetivo do modelo é identificar agrupamentos ou padrões nos dados com base nas similaridades entre as entradas.

Fonte: Imagem retirada de (SERVICES, 2023).

A Figura 4 ilustra a estrutura de um modelo não supervisionado, onde as variáveis de entrada são denotadas por X_1, X_2 até X_p . Diferentemente dos modelos supervisionados, aqui não há uma variável-alvo (como um *Target*) pré-definida para guiar o treinamento. Em vez disso, o modelo busca agrupar as amostras de acordo com similaridades nas suas características, descobrindo padrões ocultos nos dados de forma autônoma.

No contexto de agrupamento de dados não rotulados, o algoritmo *K-Means Clustering* foi aplicado para identificar padrões e agrupar imagens em diferentes clusters. O *K-Means* funciona particionando os dados em k grupos, onde cada amostra pertence ao cluster cujo centro está mais próximo dela. Este processo é iterativo, e os centros dos clusters são atualizados até que o modelo converja para uma solução estável.

Embora o *K-Means* não utilize rótulos durante o treinamento, ele se mostrou útil para agrupar imagens com características semelhantes, fornecendo insights adicionais sobre a estrutura dos vetores de características. Isso foi especialmente valioso para compreender como as imagens se distribuem em diferentes grupos de características, o que posteriormente auxiliou na tarefa de classificação supervisionada, ao facilitar a identificação de padrões nos dados.

Além disso, o K-Means provou ser eficiente na descoberta de padrões ocultos, permitindo uma melhor compreensão da estrutura dos dados. Esse tipo de análise possibilitou ajustes mais precisos no modelo geral de classificação, melhorando sua acurácia ao fornecer uma visão mais aprofundada sobre como as características das imagens se correlacionam (MACQUEEN, 1967; PEDREGOSA et al., 2011b).

Assim, o *K-Means Clustering* desempenhou um papel fundamental na análise exploratória de dados, contribuindo para o processo de modelagem ao fornecer uma categorização inicial dos dados que ajudou a guiar o desenvolvimento de modelos supervisionados e a refinar a tarefa de classificação subsequente.

2.3.3 Modelos de Aprendizado por Reforço

Os modelos de aprendizado por reforço diferem dos modelos supervisionados e não supervisionados, pois se baseiam na interação contínua com o ambiente para aprender. Nesse tipo de abordagem, o modelo recebe feedback em forma de recompensas ou penalidades, permitindo que ele ajuste suas ações para maximizar os retornos acumulados ao longo do tempo. Um dos algoritmos mais populares nessa categoria é o *Q-Learning*.

A Figura 5 ilustra um jogo de ping-pong, frequentemente utilizado como uma analogia visual para o aprendizado por reforço. Nesse cenário, o agente (um dos jogadores) aprende a tomar ações (rebater a bola) com base no estado do ambiente (a posição da bola e do adversário), e recebe recompensas ou penalidades (ganhar ou perder pontos) com o objetivo de melhorar sua estratégia ao longo do tempo. O modelo de aprendizado por reforço, da mesma forma, aprende a otimizar suas decisões por meio de interações sucessivas com o ambiente, ajustando suas previsões com o passar do tempo para obter melhores resultados.

O algoritmo *Q-Learning* foi utilizado para explorar estratégias de melhoria contínua na classificação de imagens. Diferente dos métodos supervisionados, o *Q-Learning* não depende de pares de entrada-saída conhecidos, mas ajusta suas previsões com base em interações repetidas com o conjunto de dados de imagens, de modo a maximizar uma

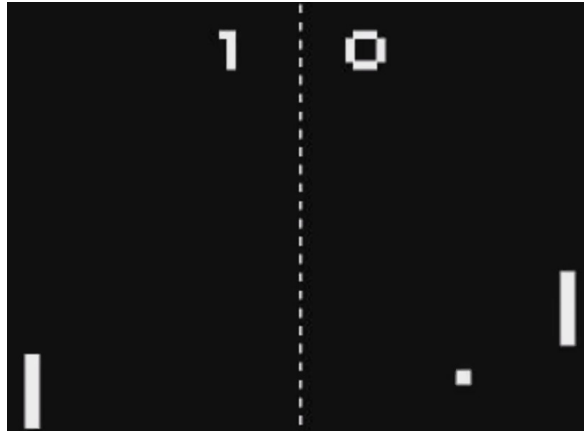


Figura 5 – Ilustração de um jogo de ping-pong, que pode ser utilizado como analogia ao aprendizado por reforço. Nesse contexto, o agente (um dos jogadores) realiza ações (como rebater a bola) em resposta ao estado atual do jogo, recebendo recompensas (ganhar pontos) com o objetivo de otimizar seu desempenho ao longo das interações.

Fonte: Imagem retirada de (NEVES, 2020)

função de recompensa. Essa função recompensa o modelo por associar corretamente os vetores de características às classes previstas, incentivando a correção de erros e o aprimoramento progressivo do modelo.

Além disso, uma das vantagens do *Q-Learning* é que ele não requer conhecimento prévio do ambiente, o que torna este algoritmo particularmente útil em situações onde a estrutura dos dados não é completamente compreendida. Isso possibilitou ao modelo aprender com seus próprios erros, explorando e aproveitando os padrões nos dados de imagens de maneira mais eficiente (WATKINS; DAYAN, 1992; OPENAI, 2016).

Em resumo, o aprendizado por reforço, com a aplicação do algoritmo *Q-Learning*, proporcionou uma abordagem alternativa e eficaz para melhorar a performance do modelo de classificação, adaptando-se continuamente com base em feedbacks obtidos a partir de suas interações com o ambiente.

2.3.4 Modelos de Deep Learning

Os modelos de *Deep Learning*, em particular as *Convolutional Neural Networks* (CNNs), tornaram-se ferramentas indispensáveis para o processamento de imagens. Esses modelos são capazes de capturar características complexas e sutis que são difíceis de serem detectadas por algoritmos tradicionais, principalmente devido à sua capacidade de trabalhar com grandes volumes de dados e com estruturas altamente complexas.

Na Figura 6 é apresentado o fluxo de um modelo de *Deep Learning* que realiza a classificação de imagens. O processo começa com o *input*, representado por uma imagem de um carro. Em seguida, a imagem é processada pelas camadas de uma CNN. Essas camadas são responsáveis pela extração de características (*feature extraction*), onde pa-

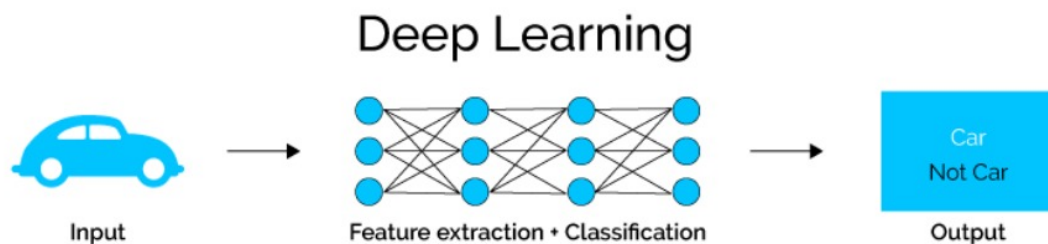


Figura 6 – Fluxo de um modelo de *Deep Learning* aplicado à classificação de imagens. O input (uma imagem de carro) passa por uma rede neural convolucional, composta por estágios de extração de características e classificação. O output final indica se a imagem representa um carro ou não.

Fonte: Diagrama adaptado de (PRETO, 2023).

drões visuais são detectados e abstraídos em diferentes níveis de complexidade. Após essa extração, as informações são passadas para as camadas de classificação (*classification*), que determinam o rótulo final da imagem, ou seja, se a imagem representa um carro ou não (*output*).

As CNNs são compostas por várias camadas convolucionais que atuam extraindo informações espaciais das imagens, como bordas, texturas e formas, e gradualmente aprendendo representações mais abstratas e complexas. Isso permite que o modelo capture detalhes visuais com maior precisão, facilitando a distinção entre classes de imagens.

Durante os testes realizados, as *Convolutional Neural Networks* demonstraram ser a abordagem mais eficaz para a tarefa de classificação de imagens. A arquitetura convolucional das CNNs permitiu que o modelo identificasse padrões específicos, como as características de um carro (rodas, janelas, faróis), com alta precisão. Essa capacidade de extração detalhada de características fez com que o desempenho das CNNs superasse outros modelos de aprendizado de máquina, especialmente quando se tratava de lidar com grandes volumes de dados e vetores de características com alta dimensionalidade.

Além disso, as CNNs foram particularmente eficientes em processar dados de imagem com variações em escala, rotação e iluminação, o que contribuiu significativamente para a robustez do modelo. O treinamento em grandes quantidades de dados, aliado à estrutura hierárquica das camadas convolucionais, possibilitou que o modelo aprendesse a diferenciar entre diversas classes de objetos com um nível de acurácia superior.

Com base nos resultados dos testes, as CNNs proporcionaram um aumento significativo na taxa de acerto na classificação de imagens, destacando-se como a melhor solução para problemas que envolvem grandes quantidades de dados visuais complexos. Assim, o *Deep Learning*, por meio do uso de CNNs, provou ser essencial na solução de problemas de visão computacional (LECUN et al., 1998; ABADI et al., 2015).

2.3.5 Modelos de Aprendizado Semi-Supervisionado

O aprendizado semi-supervisionado representa uma abordagem intermediária entre o aprendizado supervisionado e o não supervisionado. Nesse tipo de modelo, uma parte dos dados é rotulada, enquanto a outra permanece não rotulada, e o algoritmo aprende com ambas as fontes. Esse método é particularmente útil em cenários onde obter dados rotulados é custoso ou difícil, mas existe uma abundância de dados não rotulados.



Figura 7 – Comparação entre modelos supervisionado, semi-supervisionado e não supervisionado. O modelo supervisionado (esquerda) possui todos os dados rotulados, onde as bolinhas azuis estão à esquerda e as amarelas à direita. O modelo semi-supervisionado (meio) tem predominantemente bolinhas cinzas (não rotuladas), com algumas bolinhas rotuladas (azuis à esquerda e amarelas à direita). O modelo não supervisionado (direita) trabalha apenas com bolinhas cinzas, ou seja, dados sem rótulos.

Fonte: Diagrama adaptado de (ALURA, 2023).

Na Figura 7 são apresentados três exemplos que ilustram as diferenças entre os modelos supervisionado, semi-supervisionado e não supervisionado. No modelo supervisionado, à esquerda, todas as amostras possuem rótulos (bolinhas azuis à esquerda e bolinhas amarelas à direita), permitindo que o algoritmo aprenda diretamente com as classes definidas. No modelo semi-supervisionado, ao centro, a maior parte dos dados não é rotulada (bolinhas cinzas), mas existem algumas amostras rotuladas que orientam o aprendizado, com predominância de bolinhas azuis à esquerda e algumas amarelas à direita. Já no modelo não supervisionado, à direita, nenhum dado é rotulado, representado por bolinhas inteiramente cinzas.

Um dos algoritmos mais comuns nesse tipo de aprendizado é o *Label Propagation*, que foi aplicado neste trabalho para melhorar a classificação de imagens em cenários com dados rotulados limitados. O *Label Propagation* funciona propagando os rótulos dos dados rotulados para os dados não rotulados com base na similaridade entre eles. Isso permite que o modelo utilize as informações dos dados rotulados para “preencher as lacunas” e melhorar a acurácia geral do sistema.

Durante os testes realizados, o *Label Propagation* demonstrou ser eficaz ao utilizar um pequeno conjunto de dados rotulados, propagando os rótulos para os dados não rotulados com base em padrões identificados nos vetores de características das imagens. Esse

processo de propagação de rótulos aumentou a capacidade do modelo de generalizar e melhorar a acurácia sem a necessidade de rotular manualmente uma grande quantidade de dados.

Em cenários onde os dados rotulados eram escassos, o modelo semi-supervisionado foi capaz de aproveitar os dados não rotulados para ajustar as fronteiras de decisão, o que resultou em uma classificação mais precisa. Este modelo provou ser particularmente vantajoso em aplicações onde a rotulagem manual de grandes conjuntos de dados é impraticável ou muito custosa. Dessa forma, o aprendizado semi-supervisionado oferece uma solução eficiente ao combinar a robustez do aprendizado supervisionado com a flexibilidade do aprendizado não supervisionado (ZHU; GHAMRANI; LAFFERTY, 2003; PEDREGOSA et al., 2011c).

2.3.6 Modelos de Aprendizado por Transferência

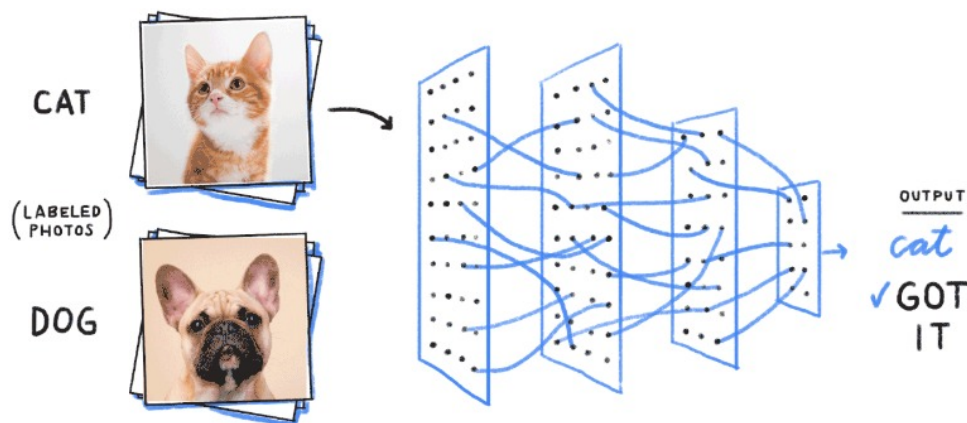


Figura 8 – Diagrama representando a abordagem de Aprendizado por Transferência, onde um classificador MLP é utilizado para categorizar imagens de gatos e cachorros.

Fonte: Diagrama adaptado de (GUPTA, 2020).

O aprendizado por transferência é uma técnica valiosa que visa otimizar o processo de treinamento de modelos de aprendizado de máquina, especialmente quando os dados disponíveis para treinamento são limitados. No presente estudo, foi empregado o MLP-Classifer (Classificador baseado em Redes Neurais Multicamadas) em uma abordagem de *Transfer Learning*.

Neste contexto, o conhecimento adquirido em um conjunto de dados de imagens previamente rotuladas foi transferido para um novo conjunto de dados relacionado, permitindo que o modelo iniciasse de uma base já treinada, ao invés de ter que aprender do zero. Essa estratégia não apenas reduz o tempo de treinamento, mas também melhora significativamente a eficiência e a precisão da classificação. A utilização do *MLPClassifier* neste cenário possibilitou que o modelo aproveitasse características aprendidas anterior-

mente, acelerando o processo de aprendizado e resultando em alta precisão nas previsões de classes de imagens, como a distinção entre gatos e cachorros.

Além disso, o uso de aprendizado por transferência se mostrou particularmente benéfico em tarefas onde a quantidade de dados rotulados é escassa, uma situação comum em muitos cenários de classificação de imagens. Assim, essa abordagem não apenas otimiza o desempenho do modelo, mas também facilita a adaptação a novas tarefas e contextos com menos esforço de treinamento (RUMELHART; HINTON; WILLIAMS, 1986; PEDREGOSA et al., 2011d; GUPTA, 2020).

2.4 Processamento de Imagens

O processamento digital de imagens consiste em um conjunto de técnicas voltadas para a análise de dados multidimensionais adquiridos por diversos tipos de sensores, como câmeras, scanners e sensores remotos. Essas técnicas envolvem a manipulação de uma imagem por meio de um computador, de forma que tanto a entrada quanto a saída do processo são representações visuais de imagens (GONZALEZ; WOODS, 2009). O objetivo principal dessas técnicas é a melhoria da qualidade da imagem, seja para uma análise visual mais eficaz, seja para a extração de informações que possam ser utilizadas em outros processos de interpretação ou tratamento de dados.

O uso dessas técnicas é amplamente difundido em diversas áreas, como medicina, geoprocessamento, e robótica, onde a qualidade e precisão das imagens são fundamentais para a tomada de decisões. No contexto médico, por exemplo, o processamento de imagens permite a detecção precoce de doenças a partir de exames como ressonâncias magnéticas e tomografias (SONKA; HLAVAC; BOYLE, 2007). Já no campo da geoprocessamento, ele é utilizado para a análise de imagens de satélite, facilitando a identificação de mudanças ambientais e o monitoramento de áreas de interesse (GONZALEZ; WOODS, 2009).

Além da melhoria do aspecto visual de certas feições estruturais para a análise humana, o processamento de imagens fornece subsídios para a sua interpretação automática. Ferramentas computacionais são empregadas para facilitar a identificação e extração de informações relevantes das imagens. A extração de características, segmentação e reconhecimento de padrões são algumas das técnicas aplicadas, que muitas vezes requerem sistemas dedicados de computação, principalmente quando a quantidade de dados é grande ou as tarefas exigem alta precisão e velocidade (SZELISKI, 2010).

Nos últimos anos, o uso de Inteligência Artificial (IA) e aprendizado de máquina tem revolucionado o campo do processamento de imagens. Algoritmos de aprendizado profundo, como redes neurais convolucionais (CNNs), são capazes de realizar tarefas complexas como a classificação de objetos, detecção de faces e reconhecimento de textos em imagens com alta acurácia (LECUN; BENGIO; HINTON, 2015). Esses avanços têm permitido que o processamento de imagens se torne cada vez mais automatizado e eficiente,

com aplicações em áreas que vão desde veículos autônomos até sistemas de segurança baseados em reconhecimento facial.

A relação entre o Processamento de Imagens e a Inteligência Artificial tem se tornado cada vez mais estreita, sendo difícil dissociar uma da outra nos dias atuais. Com a crescente capacidade computacional e o desenvolvimento de algoritmos mais sofisticados, os sistemas inteligentes conseguem processar e interpretar imagens em tempo real, realizando tarefas que antes eram exclusivas da percepção humana. Dessa forma, o processamento digital de imagens não apenas melhora a qualidade das imagens, mas também habilita a criação de sistemas autônomos capazes de tomar decisões baseadas nas informações extraídas das imagens (GONZALEZ; WOODS, 2009).

2.5 Trabalhos Correlatos

O trabalho realizado por Ribeiro, Oliveira e Barcelos (2018) baseados no trabalho de (PEREA; CARLSSON, 2014a), apresenta uma nova abordagem para caracterização de imagens baseadas em texturas através de projeções topológicas na garrafa de Klein de pequenas regiões (patches) de alto contraste extraídas das imagens. Várias configurações de tamanhos de patch foram analisadas para reduzir o tamanho do vetor de feições e aumentar a precisão. Experimentos utilizando o método proposto para classificação de texturas, em vários conjuntos de dados estabelecidos, mostram que o método proposto não só consegue reduzir o tamanho do vetor de feições, mas também melhora as taxas de classificação correta quando comparado a outros métodos do estado da arte.

Segundo o teorema de El Soufi-Ilias, que foi utilizado no trabalho de Adams e Moy (2021), esse teorema estabelece uma conexão entre subvariedades mínimas de esferas e métricas extremas para autovalores do operador de Laplace-Beltrami. Recentemente, essa conexão foi usada para fornecer vários exemplos explícitos de métricas extremas. Investigamos as propriedades dessas métricas e provamos que nenhuma delas é máxima.

Segundo Lu (2014) as técnicas de processamento de imagens subaquáticas são importantes para melhorar a qualidade das imagens subaquáticas. Como mencionado anteriormente, as imagens subaquáticas têm pouca visibilidade devido à dispersão do meio e à distorção da luz. Em contraste com as fotografias comuns, as imagens ópticas subaquáticas sofrem de pouca visibilidade devido ao meio, que causa dispersão, distorção de cor e absorção. Grandes partículas suspensas causam dispersão semelhante à dispersão da luz em neblina ou água turva que contém muitas partículas suspensas. A distorção de cor ocorre porque diferentes comprimentos de onda são atenuados em diferentes graus na água; conseqüentemente, as imagens do ambiente nos ambientes subaquáticos são dominadas por um tom azulado, pois comprimentos de onda maiores são atenuados mais rapidamente. A absorção de luz na água reduz substancialmente sua intensidade. A atenuação aleatória da luz causa uma aparência nebulosa, pois a luz retroespalhada pela

água ao longo da linha de visão degrada consideravelmente o contraste da imagem. Especialmente, objetos a uma distância de mais de 10 metros do ponto de observação são quase ilegíveis porque as cores são desbotadas como comprimentos de onda característicos, que são filtrados de acordo com a distância percorrida pela luz na água. Portanto, os métodos tradicionais de processamento de imagens não são adequados para processá-las bem. Esta tese propõe estratégias e soluções para resolver os problemas acima mencionados dos sistemas de pesquisa subaquática.

No trabalho de Tronco e Porto (1999) da implementação de um sistema de reconhecimento de imagens, baseado em Redes Neurais, para ser utilizado em operações de montagem automatizada. O sistema é baseado na utilização do modelo GSN de Rede Neural, incorporando modificações no modelo básico, as quais são baseadas em conceitos de Algoritmos Genéticos. Tais modificações proporcionam o controle dinâmico da geração de redes iniciais, da ordem de apresentação das imagens ao sistema e da geração de códigos de saída. A principal característica apresentada pelo sistema é a de invariância geométrica, possibilitando ao mesmo trabalhar com imagens transladadas, escalonadas e rotacionadas em relação as imagens previamente aprendidas. Outra característica importante do sistema é sua capacidade de trabalhar com imagens ruidosas. Todos estes aspectos levam em conta também o requisito de tempo real de operação, necessário para os requisitos impostos pela aplicação em questão. Em sua estrutura final, o sistema foi projetado de forma a poder ser implementado em uma arquitetura paralela, em especial a Arquitetura CPAD.

No Trabalho de Martins e Rillo (1992), a detecção de bordas e contornos de uma imagem é uma tarefa fundamental na estimação de propriedades físicas dos objetos que compõem o cenário original. As técnicas convencionais de detecção de bordas (*edge detection*) mostram pouca eficiência quando seu desempenho é comparado aos dois sistemas naturais de processamento. Neste trabalho, foi proposta e desenvolvida uma nova técnica de detecção e realçamento de bordas, baseada em operadores diferenciais direcionais de primeira ordem e em mecanismos locais de otimização global através de cooperação e competição, característicos de sistemas de processamento paralelo e distribuído conhecidos como redes neurais (*neural networks*). Tal detector foi implementado como um simulador, executável sobre hardware convencional. Imagens sintéticas e reais foram processadas pelo detector. Critérios para a avaliação de desempenho foram definidos e aplicados a fim de se estabelecer o mérito e a aplicabilidade do detector proposto.

2.5.1 Considerações Finais

A análise dos trabalhos correlatos evidencia a evolução das técnicas utilizadas na caracterização de imagens baseadas em texturas e no uso de redes neurais para processamento de imagens. Os estudos apresentados demonstram diferentes abordagens para aprimorar a extração e a representação de feições, seja por meio de projeções topológicas, como no

caso da Garrafa de Klein, seja por técnicas mais recentes de aprendizado de máquina e redes neurais artificiais.

Além disso, os desafios enfrentados em domínios específicos, como processamento de imagens subaquáticas e detecção de bordas em cenários complexos, reforçam a necessidade do desenvolvimento de métodos mais robustos e adaptáveis. A integração de técnicas estatísticas tradicionais com modelos baseados em redes neurais abre novas possibilidades para melhorar a precisão e a eficiência na classificação de imagens, permitindo aplicações mais abrangentes em diversas áreas, como visão computacional, engenharia e inteligência artificial.

Dessa forma, este trabalho busca contribuir com essa evolução ao explorar métodos modernos de aprendizado de máquina aplicados à caracterização de texturas, considerando tanto os avanços na literatura quanto os desafios ainda em aberto na área.

Metodologia

3.1 Experimentos e Análise dos Resultados

Neste capítulo, apresentam-se os experimentos realizados e a análise dos resultados obtidos ao aplicar diferentes modelos de aprendizado de máquina na classificação de imagens, utilizando o conceito de Projeções no Espaço da Garrafa de Klein. A abordagem experimental visa comparar e avaliar a eficácia de vários algoritmos em termos de precisão e desempenho na tarefa de caracterização de texturas artificiais e naturais.

Os experimentos foram conduzidos com uma variedade de modelos, cada um com características e abordagens distintas para o problema de classificação. A seguir, detalha-se os processos envolvidos na reimplementação dos algoritmos e a aplicação dos modelos de aprendizado de máquina.

3.1.1 Pseudocódigo para Treinamento de Modelos de Aprendizado de Máquina

A seguir, apresenta-se o pseudocódigo utilizado para a aplicação dos modelos de aprendizado de máquina. Este processo envolve o carregamento e pré-processamento da base de dados, a escolha do modelo, a definição dos parâmetros e a avaliação do desempenho dos modelos utilizando métricas como Accuracy, Precision, Recall e F1-score. O pseudocódigo segue os seguintes passos:

❑ **Entrada:**

- **dataset:** Base de dados
- **modelo:** Algoritmo de aprendizado de máquina (diversos algoritmos foram testados)
- **parametros:** Parâmetros do modelo

❑ **Processo:**

1. Carregar o dataset
2. Pré-processar os dados:
 - Normalizar ou padronizar variáveis
 - Dividir o dataset em treino e teste
3. Treinar o modelo com os dados de treino utilizando os parâmetros especificados
4. Fazer previsões nos dados de teste
5. Avaliar o desempenho do modelo:
 - Calcular as métricas: Accuracy, Precision, Recall, F1-score
 - Exibir as métricas obtidas
6. Retornar o modelo treinado e as métricas de avaliação

❑ **Saída:**

- Modelo treinado
- Métricas de avaliação (Accuracy, Precision, Recall, F1-score)

Este pseudocódigo reflete o processo de treinamento do modelo, desde o carregamento da base de dados até a avaliação do modelo, garantindo uma abordagem sistemática e eficiente na aplicação de aprendizado de máquina para classificação de imagens.

3.2 Reimplementação do Algoritmo em Python

O primeiro passo deste trabalho consistiu na reimplementação do algoritmo desenvolvido pelo (RIBEIRO; OLIVEIRA; BARCELOS, 2018), que inicialmente estava em MATLAB, para a linguagem Python. Esta etapa foi crucial para garantir a compatibilidade com as ferramentas modernas de aprendizado de máquina e para permitir a integração com diversos pacotes e bibliotecas disponíveis em Python. A reimplementação envolveu:

- ❑ Tradução do código MATLAB para Python, mantendo a lógica e a estrutura do algoritmo original.
- ❑ Utilização de bibliotecas Python, como NumPy, joblib, multiprocessing, openCV, para realizar operações matemáticas e computacionais equivalentes às do código MATLAB.
- ❑ Validação da implementação comparando os resultados obtidos com aqueles gerados pelo código MATLAB original.

3.3 Aplicação de Modelos de Aprendizado de Máquina

Com o código reimplementado e validado, aplicou-se diversos modelos de aprendizado de máquina para avaliar o desempenho na tarefa de classificação de imagens. Os modelos foram classificados em diferentes categorias, conforme descrito abaixo:

3.3.1 Modelos Supervisionados

- ❑ **K-Nearest Neighbors (KNN)**: Este modelo supervisionado foi utilizado para classificar as imagens com base na proximidade das amostras no espaço de características. O desempenho do KNN foi avaliado em termos de precisão.

3.3.2 Modelos Não Supervisionados

- ❑ **K-Means Clustering**: Aplicado para agrupar as imagens em clusters baseados nas características extraídas. O K-Means ajuda a identificar padrões e estrutura nos dados sem a necessidade de rótulos pré-existentes.

3.3.3 Modelos de Deep Learning

- ❑ **Convolutional Neural Networks (CNN)**: Modelos de redes neurais convolucionais foram utilizados para explorar características complexas e padrões em imagens. A eficácia das CNNs foi medida com base na precisão da classificação e na capacidade de generalização.

3.3.4 Modelos de Aprendizado por Reforço

- ❑ **Q-Learning**: Implementado para otimizar a tomada de decisão em ambientes dinâmicos. O Q-Learning foi avaliado em termos de convergência e desempenho na maximização de recompensas.

3.3.5 Modelos de Aprendizado Semi-Supervisionado

- ❑ **LabelPropagation**: Utilizado para melhorar a classificação em cenários com dados rotulados limitados. A eficiência do LabelPropagation foi medida com base na precisão e na capacidade de lidar com rótulos ausentes.

3.3.6 Modelos de Aprendizado por Transferência

- ❑ **MLPClassifier**: A utilização de redes neurais multicamadas (MLP) permitiu a adaptação do conhecimento pré-existente a novas tarefas de classificação. A per-

formance do `MLPClassifier` foi analisada com base na precisão e na capacidade de generalização para novos dados.

3.3.7 Conjunto de Parâmetros

Os seguintes parâmetros foram configurados para os algoritmos testados:

- ❑ **CNN Básica:** Utilizou-se uma arquitetura com 4 camadas convolucionais, treinamento por 4000 épocas e o otimizador ADAM com taxa de aprendizado adaptativa.
- ❑ **MLPClassifier:** Parâmetros padrão foram usados, com camadas ocultas ajustadas para um total de 100 neurônios. A taxa de aprendizado foi configurada para 0,001.
- ❑ **KNN:** O número de vizinhos (k) foi ajustado para 5. A distância euclidiana foi usada como métrica de similaridade.
- ❑ **Label Propagation:** Parâmetros padrão foram utilizados, com uma configuração de propagação de rótulos sem ajuste adicional.
- ❑ **Q-Learning:** Configurações padrão foram usadas, com uma tabela Q com valores inicializados e uma taxa de aprendizado configurada para 0,1.
- ❑ **K-Means:** O número de clusters foi definido com base na análise do método do cotovelo. Inicializações foram realizadas com o método `k-means++` para melhorar a qualidade do clustering.

3.4 Bases utilizadas

As bases de dados utilizadas neste trabalho consistem em coleções de imagens com características variadas, principalmente relacionadas à textura e materiais. Essas bases são amplamente utilizadas na área de processamento de imagens e reconhecimento de padrões, contribuindo para o desenvolvimento de técnicas de análise e classificação. A seguir, são descritas as bases utilizadas, com detalhes sobre suas características, formatos e resoluções.

- ❑ Brodatz¹: Base de texturas proveniente do livro “Textures: A Photographic Album for Artists and Designers”(BRODATZ, 1966), composta por 1.776 amostras de texturas agrupadas em 111 classes. As imagens possuem 128×128 pixels em tons de cinza com 256 níveis de intensidade.

¹ Pode ser obtida em <<https://www.ux.uis.no/~tranden/brodatz.html>>

- ❑ CURET(*Columbia-Utrecht Reflectance and Texture Database*)²: Desenvolvida pelas Universidades Columbia e Utrecht, essa base contém imagens de 61 materiais diferentes, cada um fotografado sob 92 combinações distintas de iluminação e visualização. As imagens estão em escala de cinza, com dimensões de 200×200 pixels e 8 bits/pixel, em formato PNG (DANA et al., 1999).
- ❑ Outex³: Base composta por uma coleção de cenas naturais, contendo 1.360 imagens de textura em tons de cinza (128×128 pixels). Ela foi construída através da divisão de 68 imagens originais (746×538 pixels) em 20 sub-imagens não sobrepostas (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2002).
- ❑ KTH-TIPS⁴: Essa base inclui texturas sob variações de iluminação, posição e escala, consistindo de 10 materiais. Cada material foi capturado em 9 escalas diferentes e sob 9 condições de iluminação e posições distintas, totalizando 810 imagens com dimensões de 200×200 pixels e 8 bits/pixel no formato PNG (HAYMAN et al., 2004).
- ❑ UIUCTex: Desenvolvida pela Universidade de Illinois em Urbana-Champaign, esta base contém 25 classes de texturas, com 40 amostras de cada classe. Todas as imagens são em escala de cinza, com dimensões de 640×480 pixels e 8 bits/pixel no formato JPG. A base inclui superfícies com variações de albedo, formas tridimensionais e misturas de ambas (LAZEBNIK; SCHMID; PONCE, 2005).
- ❑ VisTex(*The Vision Texture*)⁵: Mantida pelo grupo de Visão e Modelagem do MIT Media Lab, essa base contém imagens representando texturas do mundo real, capturadas sob condições práticas de iluminação e perspectiva. As imagens têm resolução de 512×512 pixels e foram divididas em 16 sub-imagens não sobrepostas de 128×128 pixels (MIT Media Lab, 2009).
- ❑ Sport Event: Base composta por 1.579 imagens em 8 categorias de eventos esportivos⁶: remo (250 imagens), badminton (200 imagens), polo (182 imagens), bocha (137 imagens), snowboard (190 imagens), croquet (236 imagens), vela (190 imagens) e escalada (194 imagens). As imagens não tem dimensões padronizadas (LI; FEI-FEI, 2007).

² Pode ser obtida em <<https://www.cs.columbia.edu/CAVE/software/curet/>>

³ Pode ser obtida em <<https://color.univ-lille.fr/datasets/extended-outex>>

⁴ Pode ser obtida em <<https://www.csc.kth.se/cvap/databases/kth-tips/download.html>>

⁵ Pode ser obtida em <<http://vismod.media.mit.edu/pub/VisTex/VisTex.tar.gz>>

⁶ Pode ser obtida em <<https://www.kaggle.com/datasets/trolukovich/uiuc-sports-event-dataset>>

3.5 Análise dos Resultados

Com base nos experimentos realizados, apresenta-se as conclusões sobre a eficácia dos modelos aplicados e suas implicações para futuras pesquisas e desenvolvimentos na área de aprendizado de máquina para classificação de imagens.

Os experimentos foram conduzidos com uma variedade de modelos, cada um com características e abordagens distintas para o problema de classificação.

3.6 Método para a Avaliação

Para a avaliação dos algoritmos de aprendizado de máquina, foram utilizadas as seguintes medidas de desempenho:

- **Acurácia:** A acurácia é uma métrica de desempenho que quantifica a proporção de previsões corretas realizadas por um modelo em relação ao total de previsões feitas. É uma medida geral da eficácia do modelo, expressa como a razão entre o número de previsões corretas e o número total de previsões. A fórmula para calcular a acurácia é dada por:

$$\text{Acurácia} = \frac{N_{\text{corretas}}}{N_{\text{total}}} \quad (1)$$

onde N_{corretas} representa o número de previsões corretas feitas pelo modelo e N_{total} é o número total de previsões realizadas. Embora a acurácia seja uma métrica amplamente utilizada, ela pode ser enganosa em cenários onde as classes estão desbalanceadas. Em tais casos, um modelo pode alcançar alta acurácia simplesmente prevendo a classe majoritária, mesmo que não tenha um bom desempenho em relação à classe minoritária (SOKOLOVA; LAPALME, 2009).

- **Precisão:** A precisão é uma métrica que avalia a proporção de previsões positivas corretas feitas pelo modelo em relação ao total de previsões positivas realizadas. Ela mede a exatidão do modelo na identificação de classes positivas, ajudando a entender a capacidade do modelo de evitar falsos positivos. A fórmula para calcular a precisão é dada por:

$$\text{Precisão} = \frac{N_{\text{verdadeiros positivos}}}{N_{\text{verdadeiros positivos}} + N_{\text{falsos positivos}}} \quad (2)$$

onde $N_{\text{verdadeiros positivos}}$ é o número de instâncias corretamente previstas como positivas e $N_{\text{falsos positivos}}$ é o número de instâncias incorretamente previstas como positivas. A precisão é especialmente importante em contextos onde o custo de falsos positivos é alto, como em diagnósticos médicos, onde identificar corretamente os casos positivos é crucial (CHICCO; JURMAN, 2021).

- **Recall:** O *recall* é uma métrica que quantifica a proporção de previsões positivas corretas em relação ao total de instâncias reais positivas. Ele mede a capacidade do modelo de identificar todas as instâncias positivas presentes nos dados, ou seja, a sua eficácia em capturar o máximo possível de verdadeiros positivos.

A fórmula para calcular o *recall* é dada por:

$$\text{Recall} = \frac{N_{\text{verdadeiros positivos}}}{N_{\text{verdadeiros positivos}} + N_{\text{falsos negativos}}} \quad (3)$$

onde $N_{\text{verdadeiros positivos}}$ é o número de instâncias corretamente previstas como positivas e $N_{\text{falsos negativos}}$ é o número de instâncias positivas que foram incorretamente previstas como negativas.

O *recall* é particularmente importante em situações onde é crucial identificar todos os casos positivos, como em detecção de fraudes ou em diagnósticos de doenças, onde falhar em identificar casos positivos pode ter consequências significativas (SOKOLOVA; LAPALME, 2009).

- **F1 Score:** O F1 Score é a média harmônica entre precisão e *recall*, fornecendo uma métrica única que combina ambas as medidas em uma única avaliação. Ele é especialmente útil quando se busca um equilíbrio entre precisão e *recall*, especialmente em situações onde há um trade-off entre evitar falsos positivos e capturar todos os positivos reais. A fórmula para calcular o F1 Score é dada por:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (4)$$

onde Precisão é a proporção de previsões positivas corretas em relação ao total de previsões positivas e *recall* é a proporção de previsões positivas corretas em relação ao total de instâncias reais positivas. O F1 Score é uma métrica útil em casos onde a distribuição das classes é desigual e tanto a precisão quanto o *recall* são importantes para a performance geral do modelo (SOKOLOVA; LAPALME, 2009).

- **Adjusted Rand Index (ARI):** O *Adjusted Rand Index* (ARI) é uma métrica de avaliação para algoritmos de clusterização que mede a similaridade entre o agrupamento realizado e uma verdade conhecida, ajustada para o acaso. O ARI é uma versão ajustada do *Rand Index*, que considera a chance de acordo aleatório ao calcular a similaridade entre dois agrupamentos. A fórmula para calcular o ARI é dada por:

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]} \quad (5)$$

onde RI é o *Rand Index* calculado para o agrupamento, e $\mathbb{E}[\text{RI}]$ é o valor esperado do *Rand Index* se os agrupamentos fossem atribuídos aleatoriamente. A fórmula ajusta

o *Rand Index* para levar em consideração a chance de coincidências aleatórias, resultando em um valor que pode variar de -1 (nenhuma concordância ajustada) a 1 (concordância perfeita). O ARI é particularmente útil para avaliar a qualidade de um agrupamento quando se tem uma verdade conhecida sobre os grupos e é importante levar em conta o acaso na avaliação da similaridade (HUBERT; ARABIE, 1985).

- **Normalized Mutual Information (NMI):** A *Normalized Mutual Information* (NMI) é uma métrica utilizada para avaliar a quantidade de informação compartilhada entre o agrupamento realizado e a verdade conhecida. A NMI normaliza o valor da informação mútua para o intervalo $[0, 1]$, proporcionando uma medida de quão bem o agrupamento realizado corresponde à verdade conhecida. A fórmula para calcular a NMI é dada por:

$$\text{NMI} = \frac{I(U, V)}{\sqrt{H(U)H(V)}} \quad (6)$$

onde: - $I(U, V)$ é a informação mútua entre os agrupamentos U e V , que quantifica a quantidade de informação compartilhada entre os dois agrupamentos. - $H(U)$ e $H(V)$ são as entropias dos agrupamentos U e V , respectivamente, que medem a incerteza ou a variabilidade dentro de cada agrupamento. A NMI é útil para comparar a qualidade do clustering com uma verdade conhecida, oferecendo um valor que pode variar de 0 (nenhuma informação compartilhada) a 1 (informação perfeitamente compartilhada). Esta métrica é valiosa em contextos onde se deseja avaliar a relação entre os clusters obtidos e uma referência conhecida, ajustando a medida para a variabilidade natural dos dados (STREHL; GHOSH, 2002).

- **Homogeneity:** A *Homogeneity* é uma métrica que avalia a homogeneidade dos clusters formados, isto é, o quão bem as instâncias dentro de um cluster pertencem à mesma classe. Em outras palavras, mede a pureza dos clusters em termos de classe, indicando a proporção de instâncias dentro de um cluster que pertencem à mesma classe. A fórmula para calcular a *Homogeneity* é dada por:

$$\text{Homogeneity} = \frac{1}{N} \sum_{k=1}^K \frac{\sum_{i \in C_k} \text{count}(y_i = \text{mode}(y_{C_k}))}{|C_k|} \quad (7)$$

onde: - N é o número total de instâncias. - K é o número de clusters. - C_k é o conjunto de instâncias no cluster k . - $\text{mode}(y_{C_k})$ é a classe mais frequente (modo) entre as instâncias no cluster k . - $\text{count}(y_i = \text{mode}(y_{C_k}))$ é o número de instâncias no cluster k que pertencem à classe mais frequente. A *Homogeneity* varia de 0 a 1, onde 1 indica que todos os clusters contêm apenas instâncias de uma única classe, e 0 indica que os clusters são completamente misturados em termos de classes. Esta métrica é útil para avaliar a qualidade do clustering quando se tem conhecimento da

classe verdadeira das instâncias e deseja-se verificar a pureza dos clusters formados (ROSENBERG; HIRSCHBERG, 2007).

- **Completeness:** A *Completeness* é uma métrica que avalia a completude dos clusters formados, indicando se todas as instâncias de uma mesma classe são atribuídas ao mesmo cluster. Em outras palavras, mede a capacidade do agrupamento de reunir todas as instâncias que pertencem a uma mesma classe dentro de um único cluster. A fórmula para calcular a *Completeness* é dada por:

$$\text{Completeness} = \frac{1}{N} \sum_{k=1}^K \frac{\sum_{i \in C_k} \text{count}(y_i = \text{mode}(y_{C_k}))}{|C_k|} \quad (8)$$

onde: - N é o número total de instâncias. - K é o número de clusters. - C_k é o conjunto de instâncias no cluster k . - $\text{mode}(y_{C_k})$ é a classe mais frequente (modo) entre as instâncias no cluster k . - $\text{count}(y_i = \text{mode}(y_{C_k}))$ é o número de instâncias no cluster k que pertencem à classe mais frequente.

A *Completeness* varia de 0 a 1, onde 1 indica que todas as instâncias de cada classe foram agrupadas no mesmo cluster, e 0 indica que as instâncias de uma classe estão distribuídas em vários clusters. Esta métrica é útil para avaliar a eficácia do agrupamento em termos de reunir todas as instâncias de uma mesma classe em um único cluster (ROSENBERG; HIRSCHBERG, 2007).

- **V-Measure:** A V-Measure é uma métrica que combina as medidas de *Homogeneity* e *Completeness* em uma única métrica para avaliar a qualidade geral dos clusters. Ela fornece uma avaliação equilibrada do agrupamento, considerando tanto a pureza dos clusters (*Homogeneity*) quanto a completude dos clusters (*Completeness*). A fórmula para calcular a V-Measure é dada por:

$$\text{V-Measure} = \frac{\text{Homogeneity} \cdot \text{Completeness}}{\text{Homogeneity} + \text{Completeness}} \quad (9)$$

ou, de forma mais detalhada, a V-Measure é calculada como a média ponderada entre a *Homogeneity* e a *Completeness*:

$$\text{V-Measure} = \frac{2 \cdot \text{Homogeneity} \cdot \text{Completeness}}{\text{Homogeneity} + \text{Completeness}} \quad (10)$$

onde: - *Homogeneity* é a métrica de homogeneidade dos clusters. - *Completeness* é a métrica de completude dos clusters. A V-Measure varia de 0 a 1, onde 1 indica que o agrupamento é perfeito em termos de *Homogeneity* e *Completeness*, e 0 indica que não há correspondência entre o agrupamento realizado e a verdade conhecida. Esta métrica é útil para fornecer uma visão geral da qualidade do agrupamento ao combinar as duas medidas importantes em um único valor (ROSENBERG; HIRSCHBERG, 2007).

□ **Curva ROC:** A Curva *Receiver Operating Characteristic* (ROC) é uma ferramenta gráfica usada para avaliar a performance de modelos de classificação binária. Ela ilustra a capacidade do modelo de distinguir entre as classes positivas e negativas em diferentes limiares de decisão. A curva ROC é traçada ao plotar a Taxa de Verdadeiros Positivos (TPR) em relação à Taxa de Falsos Positivos (FPR) para diferentes valores de limiar. A fórmula para calcular as taxas de verdadeiros positivos e falsos positivos é dada por:

$$TPR = \frac{TP}{TP + FN} \quad (11)$$

$$FPR = \frac{FP}{FP + TN} \quad (12)$$

onde:

- TP é o número de verdadeiros positivos.
- FN é o número de falsos negativos.
- FP é o número de falsos positivos.
- TN é o número de verdadeiros negativos.

Em um contexto de múltiplas classes, a Curva ROC pode ser estendida para cada classe individualmente. Neste caso, a curva ROC para cada classe é traçada separadamente, e as métricas de TPR e FPR são calculadas para cada classe com respeito a todas as outras classes combinadas como uma única classe negativa.

A área sob a curva ROC (AUC-ROC) fornece uma medida quantitativa da capacidade do modelo de separar as classes, com valores próximos a 1 indicando um excelente desempenho e valores próximos a 0.5 sugerindo um desempenho não melhor que o acaso (FAWCETT, 2006).

□ **Matriz de Confusão:** A Matriz de Confusão é uma ferramenta fundamental usada para avaliar a performance de modelos de classificação. Ela fornece uma visão detalhada sobre as previsões feitas pelo modelo, mostrando a contagem de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN). A matriz é geralmente representada em uma tabela como segue: As fórmulas

Tabela 1 – Matriz de Confusão

	Classe Positiva	Classe Negativa
Classe Positiva	TP	FN
Classe Negativa	FP	TN

para calcular cada componente da matriz são:

TP = Número de amostras corretamente classificadas como positivas

TN = Número de amostras corretamente classificadas como negativas

FP = Número de amostras incorretamente classificadas como positivas

FN = Número de amostras incorretamente classificadas como negativas

Para algoritmos de clustering como o K-Means, que não têm rótulos verdadeiros associados a cada cluster, a matriz de confusão pode ser adaptada para comparar os clusters formados com os rótulos verdadeiros conhecidos. Neste caso, é necessário fazer uma correspondência entre os clusters e as classes verdadeiras para preencher a matriz de confusão. Isso envolve associar cada cluster a uma classe verdadeira com base na maioria das instâncias no cluster e então calcular TP, TN, FP e FN para essa correspondência. Esta adaptação permite avaliar a qualidade dos clusters em termos de como eles correspondem aos rótulos reais conhecidos (MANNING; RAGHAVAN, 2008).

3.7 Experimentos e Avaliação dos Resultados

Nesta seção são apresentadas e discutidas as métricas utilizadas para avaliar o desempenho dos modelos de classificação. Cada métrica oferece uma perspectiva distinta sobre a eficácia do modelo, permitindo uma análise abrangente dos resultados obtidos.

3.7.1 Brodatz

Ao analisar os resultados dos diferentes modelos aplicados ao Base Brodatz, fica evidente a disparidade em suas performances (Tabela 2, Tabela 3, Figura 9, Figura 10, Figura 11, Figura 12, Figura 13 e Figura 14). O modelo de CNN básica, utilizando o otimizador ADAM, destacou-se como o mais eficaz, alcançando uma acurácia de 84%, além de uma precisão de 86% e um F1 Score de 83%. Essa performance superior pode ser atribuída à capacidade da rede convolucional em aprender padrões complexos, especialmente em dados com estruturas hierárquicas.

Em contrapartida, o MLPClassifier apresentou um desempenho sólido, mas inferior ao da CNN, com uma acurácia de 81%. Sua arquitetura mais simples pode limitar a captura de padrões complexos, resultando em métricas ligeiramente mais baixas. O KNN também se posicionou de forma razoável, com uma acurácia de 79%, mas sua dependência das distâncias entre pontos pode prejudicar os resultados, especialmente em conjuntos de dados de alta dimensionalidade.

Por outro lado, o modelo de LabelPropagation ficou aquém das expectativas, registrando uma acurácia de apenas 75%. Sua abordagem não supervisionada e sensibilidade à estrutura dos dados podem ter contribuído para sua performance inferior. O Q-Learning, por sua vez, falhou completamente, não obtendo resultados significativos em nenhuma

das métricas. Essa falha pode ser decorrente da inadequação da abordagem ao tipo de problema ou à formatação inadequada dos dados para aprendizado por reforço.

Em termos de agrupamento, o K-MEANS mostrou um desempenho respeitável, apresentando um *Adjusted Rand Index* (ARI) de 0,46 e um *Normalized Mutual Information* (NMI) de 0,88. Esses resultados indicam que, apesar de ser um algoritmo não supervisionado, o K-MEANS conseguiu formar clusters bem definidos que correspondem às classes reais dos dados.

Em suma, a CNN básica se destaca como o melhor modelo, devido à sua capacidade de aprender características complexas, enquanto o Q-Learning emerge como o pior, sem conseguir produzir resultados úteis. Essa análise ressalta a importância de escolher o modelo adequado e de alinhar as abordagens às características específicas do conjunto de dados em questão.

Tabela 2 – Resultados dos Modelos de Machine Learning - BrodaTz

Modelo	Acurácia	Precisão	Recall	F1 Score
CNN Básica (ADAM)	0,84	0,86	0,84	0,83
MLPClassifier	0,81	0,81	0,81	0,79
KNN	0,79	0,82	0,79	0,78
Q-Learning	0,00	0,00	0,00	0,00
LabelPropagation	0,75	0,71	0,75	0,71

Tabela 3 – Resultados do K-MEANS Base BrodaTz

Métrica	Valor
<i>Adjusted Rand Index</i> (ARI)	0,46
<i>Normalized Mutual Information</i> (NMI)	0,88
<i>Homogeneity</i>	0,85
<i>Completeness</i>	0,92
V-Measure	0,884996923

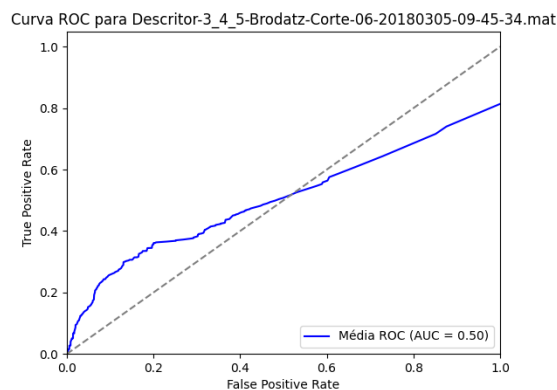


Figura 9 – Saída da curva ROC do Modelo CNN, Base BrodaTz

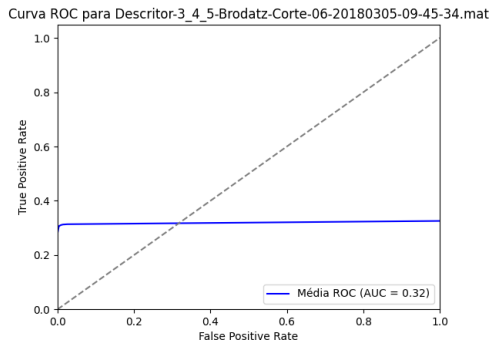


Figura 10 – Saída da curva ROC do Modelo KNN, Base BrodaTz

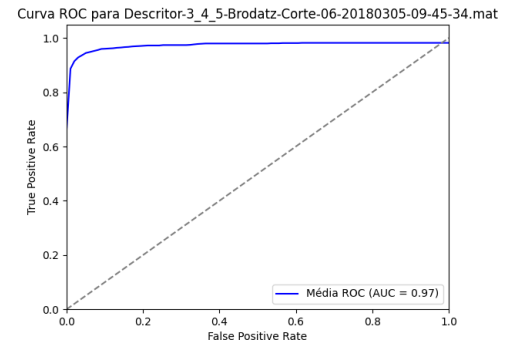


Figura 11 – Saída da curva ROC do Modelo MLP, Base BrodaTz

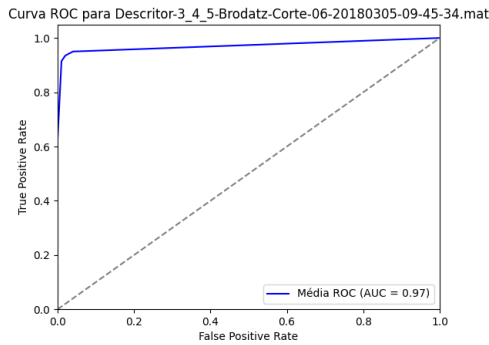


Figura 12 – Saída da curva ROC do Modelo LabelPropagation, Base BrodaTz

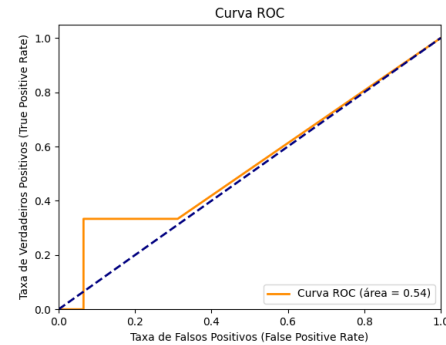


Figura 13 – Saída da curva ROC do Modelo QLearning, Base BrodaTz

3.7.2 Curet

A Base CURET é um conjunto de dados amplamente utilizado em estudos sobre detecção de doenças e análise de sinais fisiológicos. Ele contém medições de várias características, como pressão arterial e frequência cardíaca, coletadas de indivíduos ao longo do tempo. A análise deste conjunto pode oferecer *insights* valiosos sobre a saúde dos indivíduos e as relações entre diferentes variáveis.

Diversos modelos de aprendizado de máquina foram testados para avaliar seu desempenho na Base CURET (Tabela 4, Tabela 5, Figura 15, Figura 16, Figura 17, Figura 18, Figura 19 e Figura 20). A CNN básica, utilizando o otimizador ADAM, apresentou a melhor performance, alcançando uma acurácia de 86%, uma precisão de 87%, um *recall* de 86% e um F1 Score de 86%. Esses resultados destacam a eficácia da CNN em aprender padrões complexos nos dados.

Em comparação, o MLPClassifier obteve uma acurácia de 67%, com precisão e *recall* de 68% e 67%, respectivamente. O KNN também teve um desempenho razoável, alcançando uma acurácia de 81%, com precisão e *recall* de 83% e 81%. Por outro lado, o modelo de LabelPropagation registrou uma acurácia de 72%, evidenciando uma performance inferior

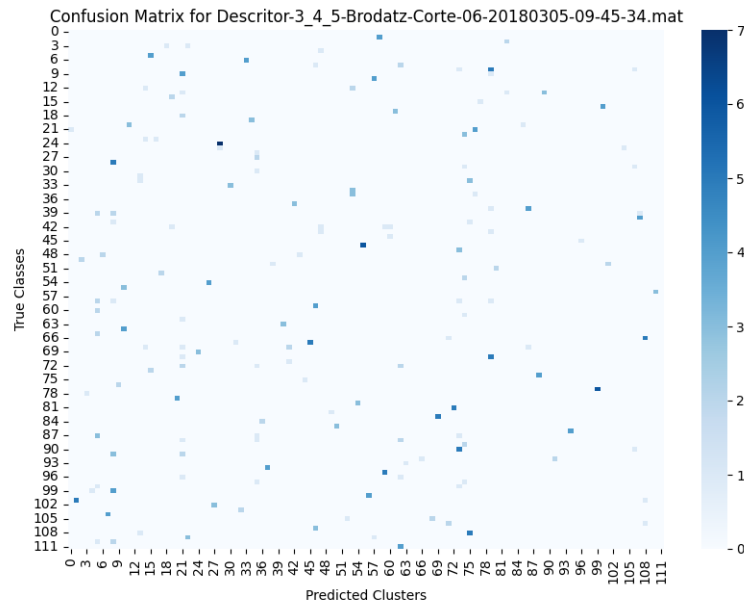


Figura 14 – Saída da Matriz de Confusão do Modelo KMeans para a Base Brodatz. A matriz exibe a correspondência entre os clusters verdadeiros (eixo vertical) e os clusters previstos (eixo horizontal). A intensidade da cor representa a frequência de ocorrências para cada par de valores, conforme indicado pela barra de cores à direita. Valores mais escuros indicam maior número de classificações em um determinado cluster previsto.

em relação aos modelos anteriores.

O Q-Learning, em contrapartida, apresentou um desempenho extremamente baixo, com acurácia, precisão, *recall* e F1 Score de apenas 0,01. Isso indica que a aplicação deste modelo não foi adequada para a análise do conjunto de dados Curet. O K-MEANS, que foi avaliado com métricas de agrupamento, obteve um *Adjusted Rand Index* de 0,20 e um *Normalized Mutual Information* de 0,59, mostrando um desempenho moderado na formação de clusters.

Em suma, o Base CURET representa uma oportunidade rica para a aplicação de técnicas de aprendizado de máquina, permitindo a identificação de padrões que podem contribuir para a detecção precoce de doenças e para a melhoria dos cuidados com a saúde.

Tabela 4 – Resultados dos Modelos de Machine Learning - CureT

Modelo	Acurácia	Precisão	Recall	F1 Score
CNN Básica (ADAM)	0,86	0,87	0,86	0,86
KNN	0,81	0,83	0,81	0,81
Q-Learning	0,01	0,01	0,01	0,01
MLPClassifier	0,67	0,68	0,67	0,66
LabelPropagation	0,72	0,66	0,72	0,68

Tabela 5 – Resultados do K-MEANS Base CURET

Métrica	Valor
<i>Adjusted Rand Index</i> (ARI)	0,20
<i>Normalized Mutual Information</i> (NMI)	0,59
<i>Homogeneity</i>	0,56
<i>Completeness</i>	0,62
V-Measure	0,5900800616

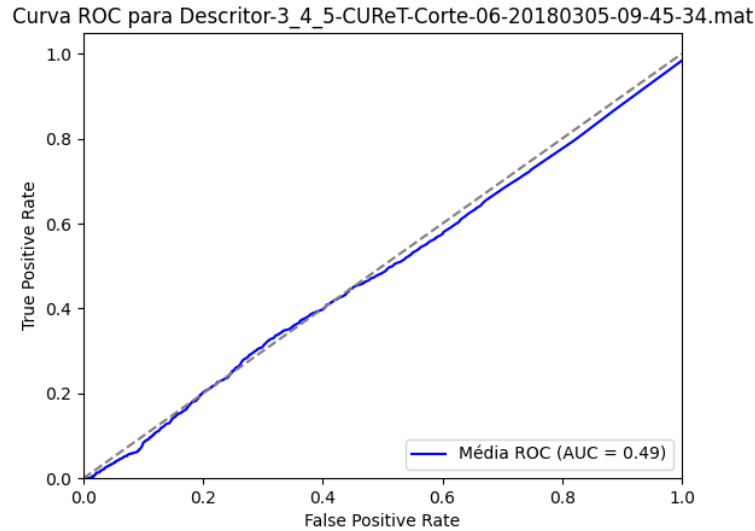


Figura 15 – Saída da curva ROC do Modelo CNN, Base CURET

3.7.3 KTH-TIPS

Diversos modelos de aprendizado de máquina foram testados para avaliar seu desempenho para a Base KTH-TIPS (Tabela 6, Tabela 7, Figura 21, Figura 22, Figura 23, Figura 24, Figura 25 e Figura 26). Na avaliação dos modelos, a CNN básica, configurada com o otimizador ADAM e composta por 4 camadas, destacou-se como a abordagem mais eficaz, alcançando uma acurácia de 88%. Além disso, apresentou precisão, *recall* e F1 Score também de 88%. Esses resultados indicam que a CNN conseguiu aprender de forma eficiente os padrões presentes nos dados.

Em comparação, o MLPClassifier obteve uma acurácia de 80%, com precisão e *recall* de 82% e 80%, respectivamente. Embora tenha mostrado um desempenho sólido, ficou aquém da CNN, possivelmente devido à sua arquitetura mais simples, que pode limitar a capacidade de capturar relações complexas.

O KNN apresentou resultados competitivos, com uma acurácia de 86%, precisão de 88% e *recall* de 86%. Esses resultados demonstram a eficácia do KNN em conjuntos de dados com características bem definidas. Por outro lado, o modelo de LabelPropagation registrou uma acurácia de 73%, com precisão e *recall* de 68% e 73%, indicando uma performance inferior aos modelos anteriores.

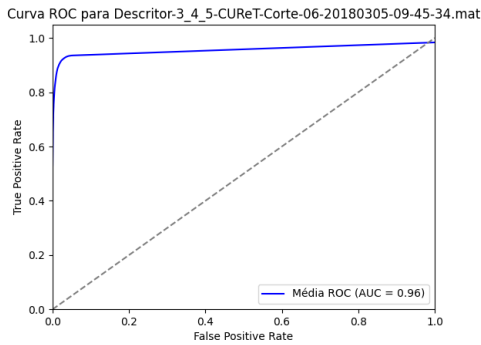


Figura 16 – Saída da curva ROC do Modelo KNN, Base CURET

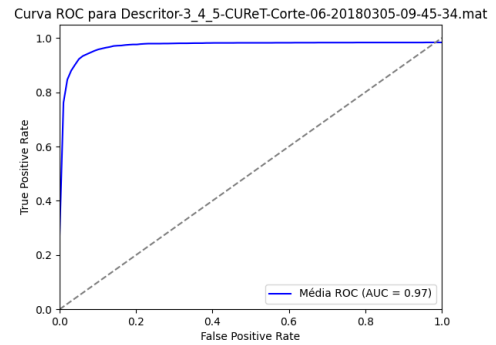


Figura 17 – Saída da curva ROC do Modelo MLP, Base CURET

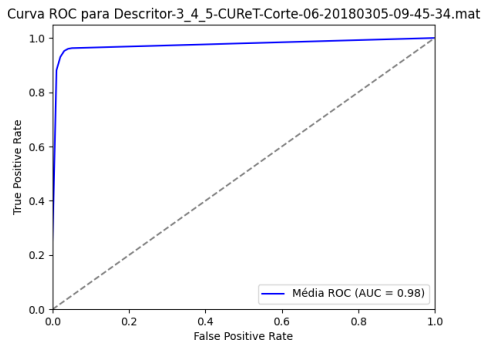


Figura 18 – Saída da curva ROC do Modelo LabelPropagation, Base CURET

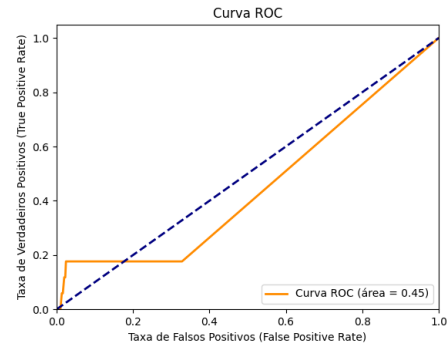


Figura 19 – Saída da curva ROC do Modelo Q Learning, Base CURET

O Q-Learning, no entanto, apresentou resultados alarmantemente baixos, com uma acurácia de apenas 2%, o que sugere que este modelo não é adequado para a análise do conjunto KTH-TIPS. As métricas de desempenho, como precisão e *recall*, também foram muito baixas, refletindo a incapacidade do modelo de aprender de forma eficaz a partir dos dados.

Por fim, o K-MEANS foi avaliado com métricas de agrupamento, apresentando um *Adjusted Rand Index* de 0,20 e um *Normalized Mutual Information* de 0,43. Esses resultados sugerem uma capacidade limitada de formar grupos que correspondam às classes reais.

Tabela 6 – Resultados dos Modelos de Machine Learning - KTH-TIPS

Modelo	Acurácia	Precisão	Recall	F1 Score
CNN Básica (ADAM)	0,88	0,88	0,88	0,88
KNN	0,86	0,88	0,86	0,86
Q-Learning	0,02	0,03	0,02	0,02
MLPClassifier	0,80	0,82	0,80	0,80
LabelPropagation	0,73	0,68	0,73	0,70

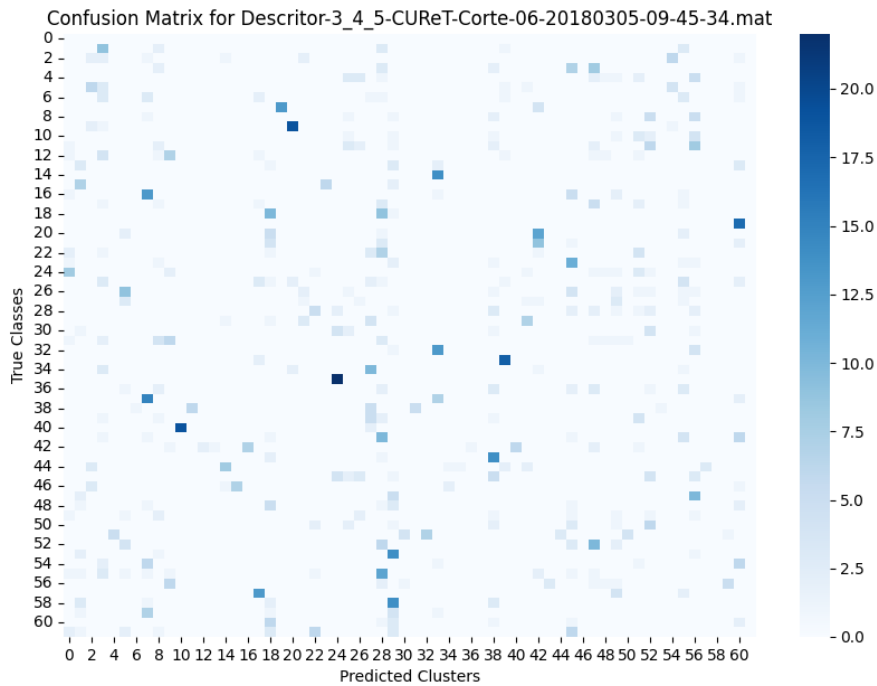


Figura 20 – Saída da Matriz de Confusão do Modelo KMEANS, Base CUReT

Tabela 7 – Resultados do K-MEANS Base KTH-TIPS

Métrica	Valor
<i>Adjusted Rand Index</i> (ARI)	0,20
<i>Normalized Mutual Information</i> (NMI)	0,43
<i>Homogeneity</i>	0,40
<i>Completeness</i>	0,47
V-Measure	0,4322699725

3.7.4 OUTEX

Na análise dos modelos aplicados à Base Outex (Tabela 8, Tabela 9, Figura 27, Figura 28, Figura 29, Figura 30, Figura 31 e Figura 32), a CNN básica, com o otimizador ADAM e composta por 4 camadas, apresentou um desempenho sólido, alcançando uma acurácia de 72%. Além disso, suas métricas de precisão, *recall* e F1 Score foram todas de 75%, 72% e 72%, respectivamente, evidenciando a capacidade da CNN em capturar padrões relevantes nas imagens.

O MLPClassifier teve um desempenho ligeiramente inferior, com uma acurácia de 70%, precisão de 74%, *recall* de 70% e F1 Score de 70%. Embora os resultados sejam competitivos, eles indicam que a arquitetura da MLP pode ser menos eficaz em comparação com a CNN para este tipo de tarefa.

O KNN, por sua vez, obteve uma acurácia de 64%, com precisão e *recall* de 67% e 64%. Esses resultados refletem a adequação do KNN para conjuntos de dados com caracterís-

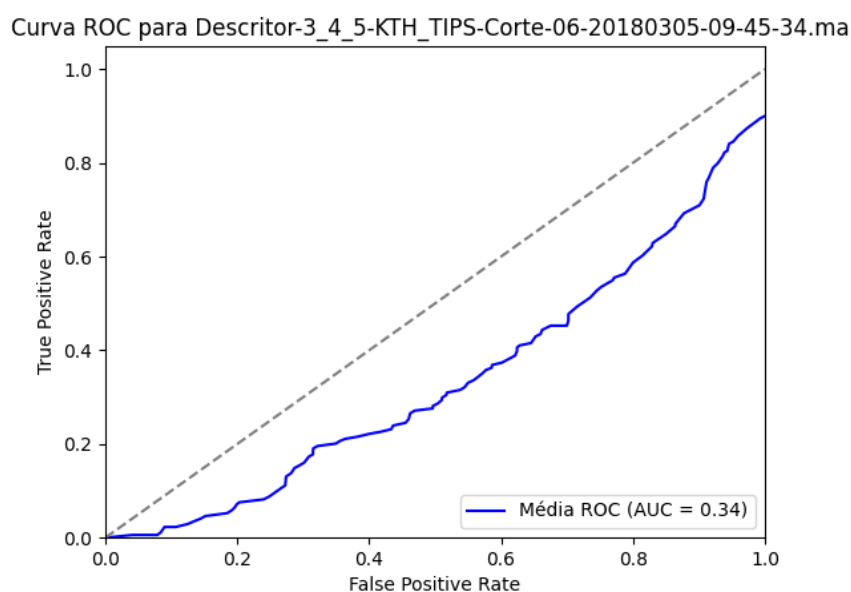


Figura 21 – Saída da curva ROC do Modelo CNN, Base KTH-TIPS

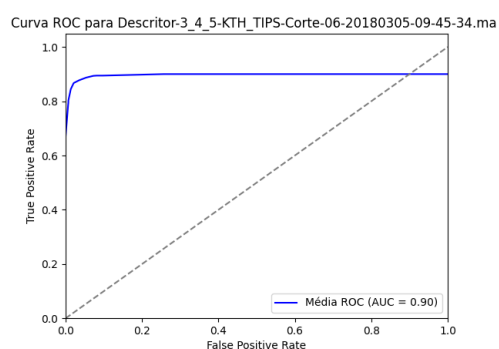


Figura 22 – Saída da curva ROC do Modelo KNN, Base KTH-TIPS

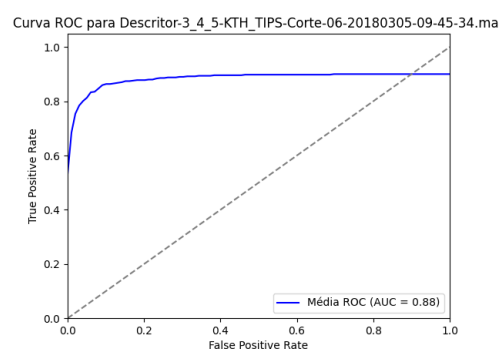


Figura 23 – Saída da curva ROC do Modelo MLP, Base KTH-TIPS

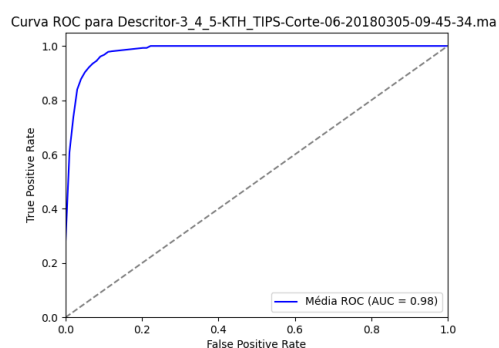


Figura 24 – Saída da curva ROC do Modelo LabelPropagation, Base KTH-TIPS

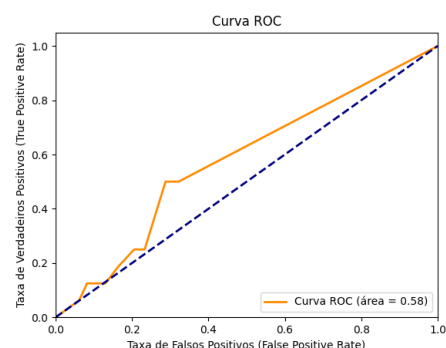


Figura 25 – Saída da curva ROC do Modelo Q Learning, Base KTH-TIPS

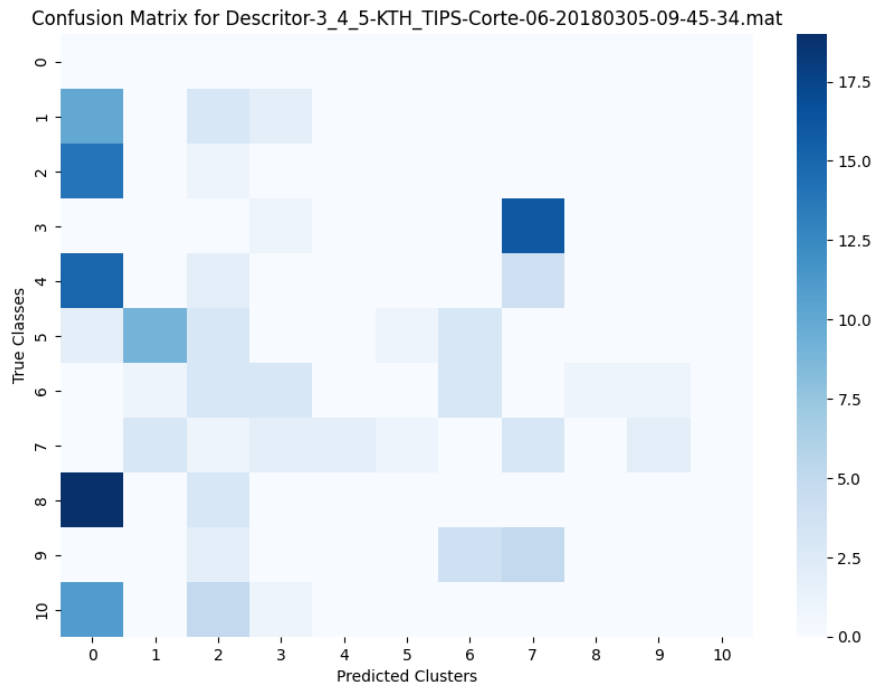


Figura 26 – Saída da Matriz de Confusão do Modelo KMEANS, Base KTH-TIPS

ticas bem definidas, mas seu desempenho ficou aquém da CNN e do MLPClassifier. O modelo de LabelPropagation apresentou uma acurácia ainda menor, de 58%, com precisão e *recall* de 53% e 58%, sugerindo que este método não se adapta bem às particularidades do conjunto de dados Outex.

O Q-Learning demonstrou um desempenho alarmantemente baixo, com uma acurácia de apenas 1%. As métricas de precisão e *recall* corroboram essa observação, indicando a inadequação do modelo para este tipo de análise. Finalmente, o K-MEANS foi avaliado utilizando métricas de agrupamento, apresentando um *Adjusted Rand Index* de 0,37 e um *Normalized Mutual Information* de 0,81. Esses resultados mostram que o K-MEANS conseguiu formar clusters razoavelmente coerentes, embora o desempenho ainda fosse modesto.

Tabela 8 – Resultados dos Modelos de Machine Learning - OUTEX

Modelo	Acurácia	Precisão	Recall	F1 Score
CNN Básica (ADAM)	0,72	0,75	0,72	0,72
KNN	0,64	0,67	0,64	0,64
Q-Learning	0,01	0,01	0,01	0,01
MLPClassifier	0,70	0,74	0,70	0,70
LabelPropagation	0,58	0,53	0,58	0,54

Tabela 9 – Resultados do K-MEANS Base Outex

Métrica	Valor
<i>Adjusted Rand Index</i> (ARI)	0,37
<i>Normalized Mutual Information</i> (NMI)	0,81
<i>Homogeneity</i>	0,76
<i>Completeness</i>	0,86
V-Measure	0,8080835455

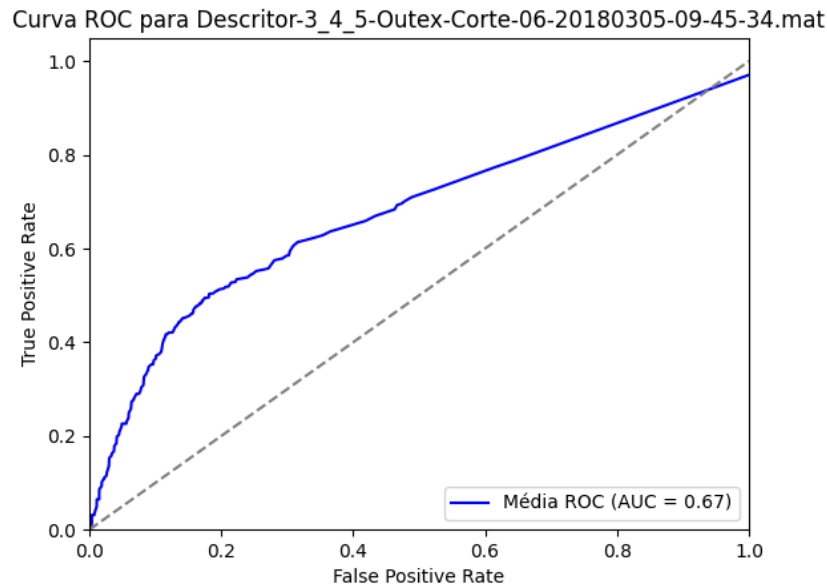


Figura 27 – Saída da curva ROC do Modelo CNN, Base Outex

3.7.5 Sport Event

Na análise dos resultados para os modelos aplicados à Base Sport Event (Tabela 10, Tabela 11, Figura 33, Figura 34, Figura 35, Figura 37, Figura 36 e Figura 38), a CNN básica, com o otimizador ADAM e composta por 4 camadas, apresentou a melhor performance entre os modelos, alcançando uma acurácia de 51%. As métricas de precisão, *recall* e F1 Score foram de 52%, 51% e 51%, respectivamente. Esses resultados indicam que a CNN foi capaz de extrair características relevantes dos dados, embora os valores ainda sejam relativamente baixos, o que pode sugerir a complexidade do conjunto de dados ou a necessidade de ajustes adicionais na modelagem.

O MLPClassifier obteve uma acurácia ligeiramente inferior, de 45%, com precisão, *recall* e F1 Score de 44%, 45% e 42%. O desempenho inferior do MLP pode ser atribuído à sua arquitetura, que pode não ter sido suficientemente complexa para capturar as nuances dos dados do evento esportivo.

O KNN mostrou uma acurácia de 47%, com precisão de 49%, *recall* de 47% e F1 Score de 45%. Embora tenha se saído melhor que o MLPClassifier, o KNN ainda não conseguiu superar a CNN, demonstrando a dificuldade do modelo em lidar com o conjunto

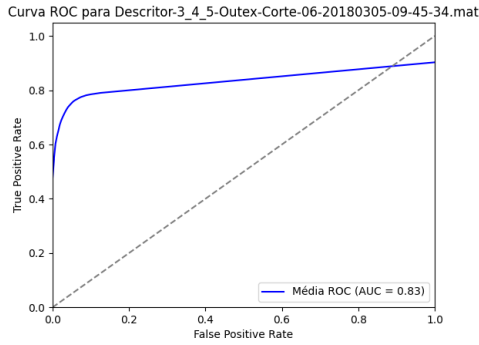


Figura 28 – Saída da curva ROC do Modelo KNN, Base Outex

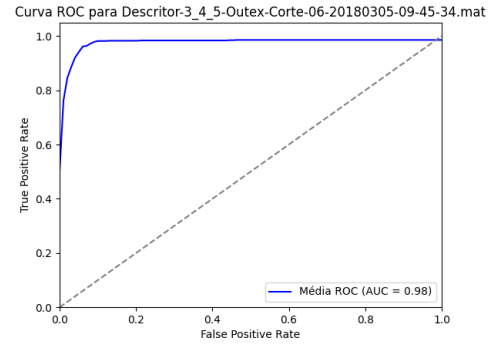


Figura 29 – Saída da curva ROC do Modelo MLP, Base Outex

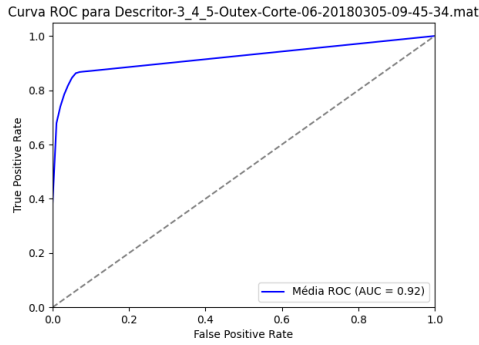


Figura 30 – Saída da curva ROC do Modelo LabelPropagation, Base Outex

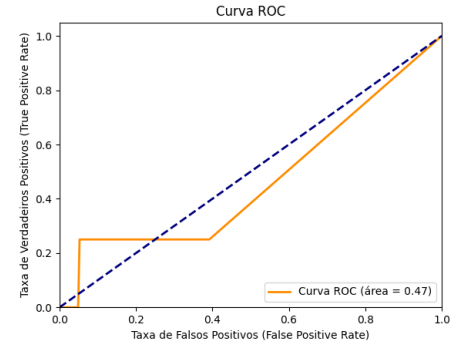


Figura 31 – Saída da curva ROC do Modelo Q Learning, Base Outex

específico.

O modelo de LabelPropagation teve um desempenho inferior, com uma acurácia de 39% e métricas de precisão, *recall* e F1 Score de 36% e 39%, refletindo sua incapacidade de se adaptar bem às características dos dados de eventos esportivos.

Por fim, o Q-Learning obteve resultados alarmantemente baixos, com uma acurácia de apenas 3%. As métricas de precisão e *recall* foram igualmente baixas, em 8% e 3%, respectivamente. Isso indica que o modelo não conseguiu aprender com os dados, possivelmente devido à natureza inadequada do algoritmo para esta tarefa. O K-MEANS também apresentou resultados modestos em termos de agrupamento, com um *Adjusted Rand Index* de 0,08 e um *Normalized Mutual Information* de 0,19, sugerindo que a separação dos dados em clusters não foi eficaz.

3.7.6 UIUCTex

Na análise dos resultados para os modelos aplicados à Base UIUCTex (Tabela 12, Tabela 13, Figura 39, Figura 40, Figura 41, Figura 42, Figura 43 e Figura 44), a CNN básica, utilizando o otimizador ADAM e composta por 4 camadas com 4000 épocas, obteve

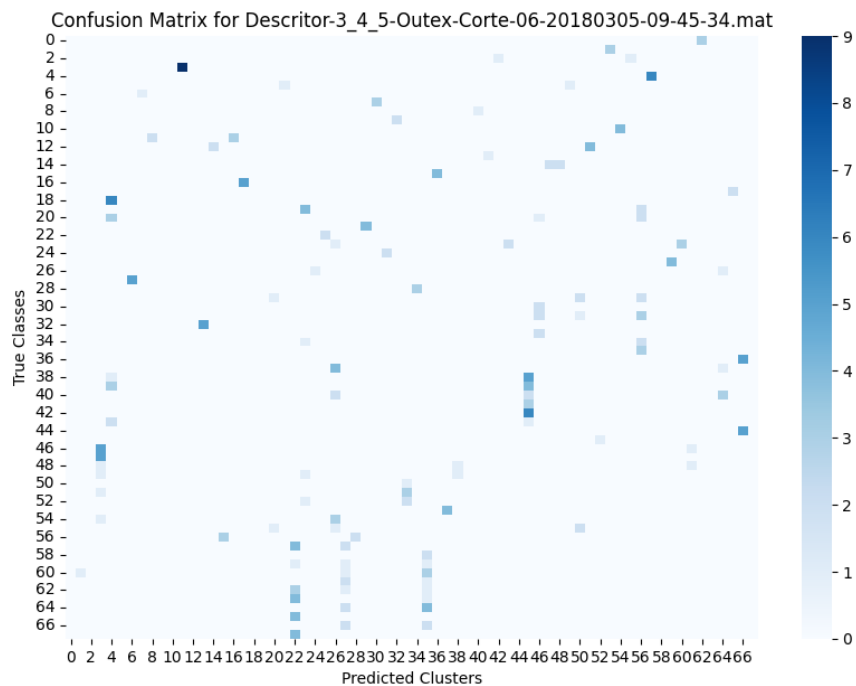


Figura 32 – Saída da Matriz de Confusão do Modelo KMEANS, Base Outex

Tabela 10 – Resultados dos Modelos de Machine Learning - Sport Event

Modelo	Acurácia	Precisão	Recall	F1 Score
CNN Básica (ADAM)	0,51	0,52	0,51	0,51
KNN	0,47	0,49	0,47	0,45
Q-Learning	0,03	0,08	0,03	0,04
MLPClassifier	0,45	0,44	0,45	0,42
LabelPropagation	0,39	0,36	0,39	0,36

Tabela 11 – Resultados do K-MEANS Base Sport Event

Métrica	Valor
<i>Adjusted Rand Index</i> (ARI)	0,08
<i>Normalized Mutual Information</i> (NMI)	0,19
<i>Homogeneity</i>	0,19
<i>Completeness</i>	0,20
V-Measure	0,1910815254

uma acurácia de 49%. As métricas de precisão, *recall* e F1 Score foram de 52%, 49% e 49%, respectivamente. Embora os resultados sejam moderados, eles indicam que a CNN conseguiu extrair informações relevantes do conjunto de dados.

O MLPClassifier apresentou uma acurácia ligeiramente inferior de 45%, com precisão, *recall* e F1 Score de 46%, 45% e 44%. Essa performance sugere que o MLP não conseguiu se adaptar adequadamente às características dos dados.

O KNN teve um desempenho semelhante, com uma acurácia de 45% e métricas de

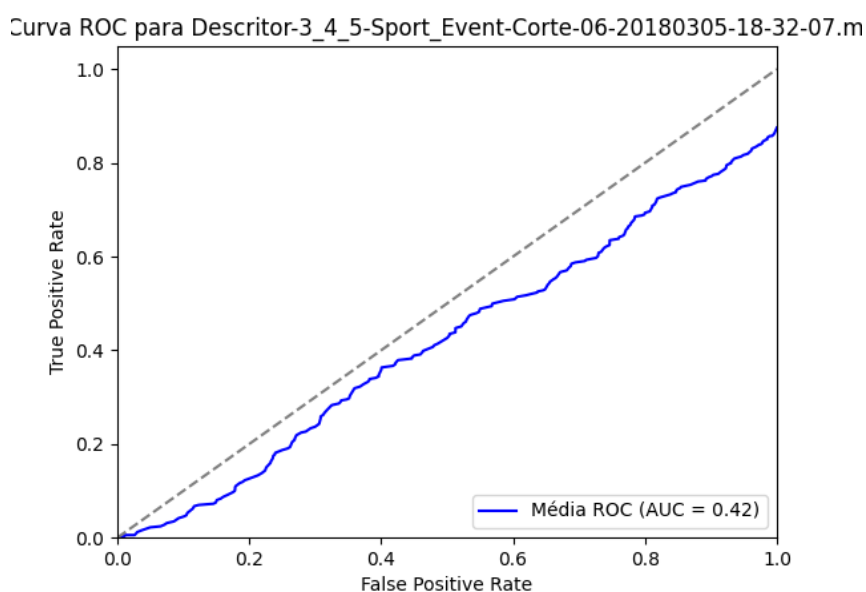


Figura 33 – Saída da curva ROC do Modelo CNN, Base Sport Event

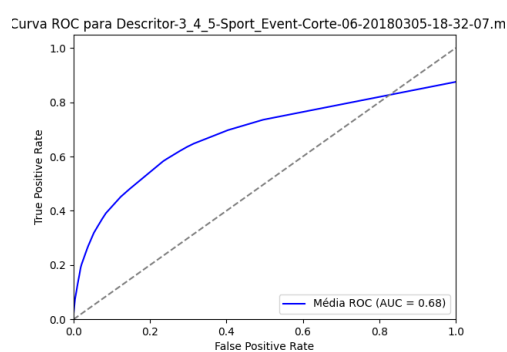


Figura 34 – Saída da curva ROC do Modelo KNN, Base Sport Event

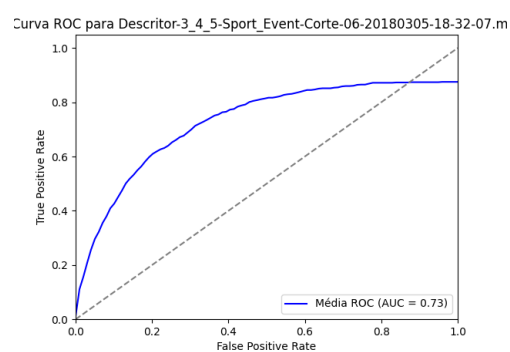


Figura 35 – Saída da curva ROC do Modelo MLP, Base Sport Event

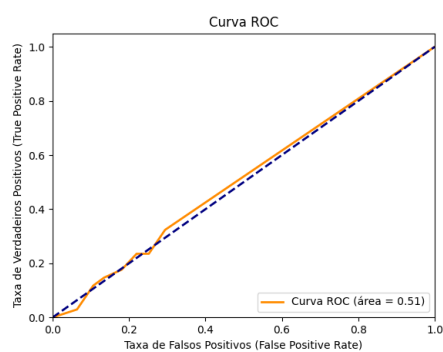


Figura 36 – Saída da curva ROC do Modelo Q Learning, Base Sport Event

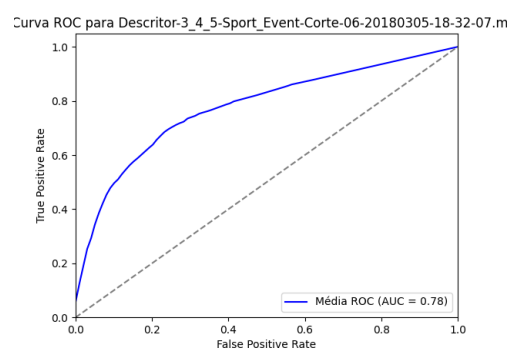


Figura 37 – Saída da curva ROC do Modelo LabelPropagation, Base Sport Event

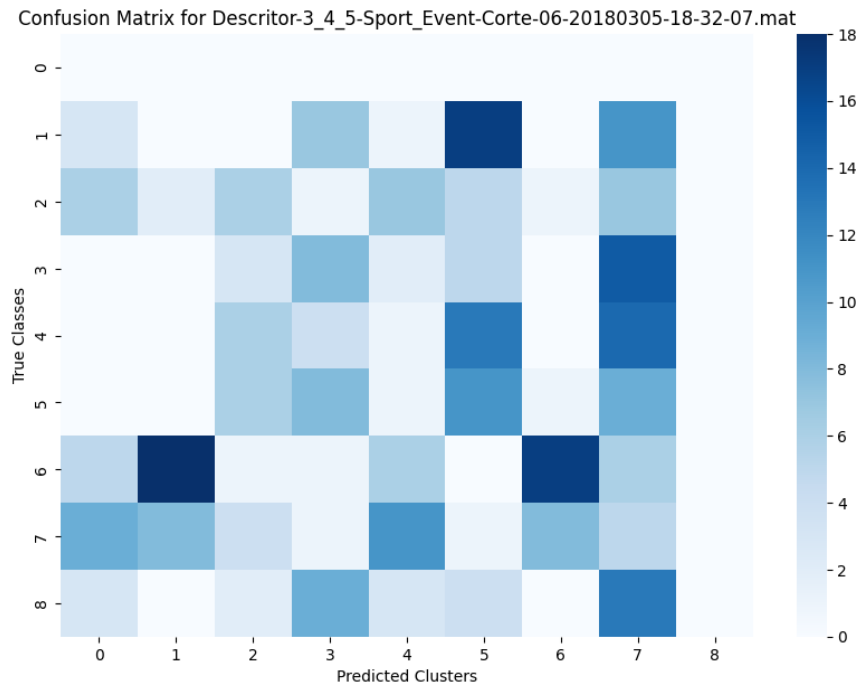


Figura 38 – Saída da Matriz de Confusão do Modelo KMEANS, Base Sport Event

precisão, *recall* e F1 Score de 55%, 45% e 45%. Embora o KNN tenha apresentado uma precisão um pouco maior, isso não se traduziu em uma acurácia superior.

O modelo de LabelPropagation obteve uma acurácia de 46%, com precisão, *recall* e F1 Score de 48%, 46% e 43%. Este desempenho é um pouco mais encorajador, mas ainda insatisfatório em comparação com o desempenho da CNN.

Em contraste, o Q-Learning falhou em produzir resultados significativos, apresentando uma acurácia de 0% e métricas de precisão, *recall* e F1 Score de 0%. Isso sugere que o modelo não foi capaz de aprender a partir dos dados disponíveis.

O K-Means apresentou métricas de agrupamento, como *Adjusted Rand Index* (ARI) de 0,10 e *Normalized Mutual Information* (NMI) de 0,47, com *Homogeneity* de 0,44 e *Completeness* de 0,51. A matriz de confusão revelou padrões complexos, indicando dificuldades em classificar corretamente os dados.

Tabela 12 – Resultados dos Modelos de Machine Learning - UIUCTex

Modelo	Acurácia	Precisão	Recall	F1 Score
CNN Básica (ADAM)	0,49	0,52	0,49	0,49
KNN	0,45	0,55	0,45	0,45
Q-Learning	0,00	0,00	0,00	0,00
MLPClassifier	0,45	0,46	0,45	0,44
LabelPropagation	0,46	0,48	0,46	0,43

Tabela 13 – Resultados do K-MEANS Base UIUCTex

Métrica	Valor
<i>Adjusted Rand Index</i> (ARI)	0,10
<i>Normalized Mutual Information</i> (NMI)	0,47
<i>Homogeneity</i>	0,44
<i>Completeness</i>	0,51
V-Measure	0,4705263786

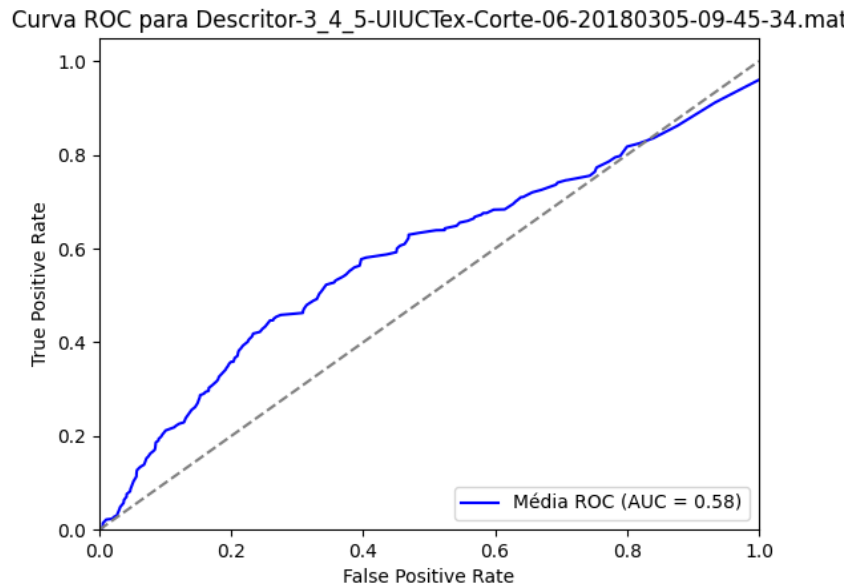


Figura 39 – Saída da curva ROC do Modelo CNN, Base UIUCTex

3.7.7 Vistex

Na análise dos resultados para os modelos aplicados à Base Vistex (Tabela 14, Tabela 15, Figura 45, Figura 46, Figura 47, Figura 48, Figura 49 e Figura 50), a CNN básica, configurada com o otimizador ADAM, quatro camadas e 4000 épocas, alcançou uma acurácia de 72%. As métricas de precisão, *recall* e F1 Score foram 76%, 72% e 71%, respectivamente. Esses resultados indicam um desempenho sólido, com a CNN conseguindo capturar características relevantes do conjunto de dados.

Em comparação, o MLPClassifier obteve uma acurácia superior de 80%, com precisão, *recall* e F1 Score de 81%, 80% e 79%. Esses resultados destacam a eficácia do MLP na classificação, superando a CNN em todas as métricas.

O KNN também apresentou um desempenho forte, alcançando uma acurácia de 80%, com precisão, *recall* e F1 Score de 82%, 80% e 79%. A performance deste modelo é comparável à do MLP, indicando uma boa capacidade de generalização.

O modelo de LabelPropagation apresentou uma acurácia de 73%, com métricas de precisão, *recall* e F1 Score de 69%, 73% e 70%. Embora tenha obtido resultados razoáveis, o desempenho foi inferior em comparação com a CNN, MLP e KNN.

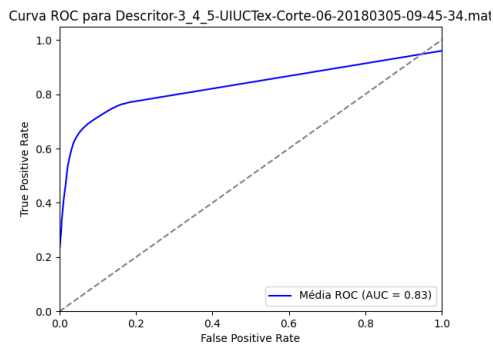


Figura 40 – Saída da curva ROC do Modelo KNN, Base UIUCTex

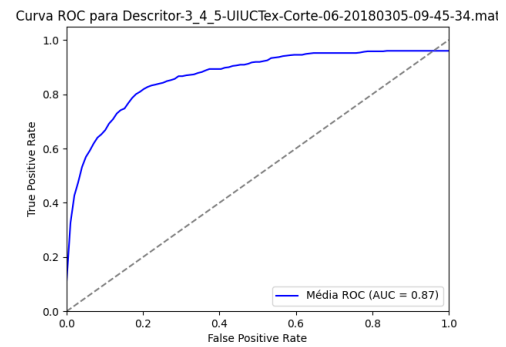


Figura 41 – Saída da curva ROC do Modelo MLP, Base UIUCTex

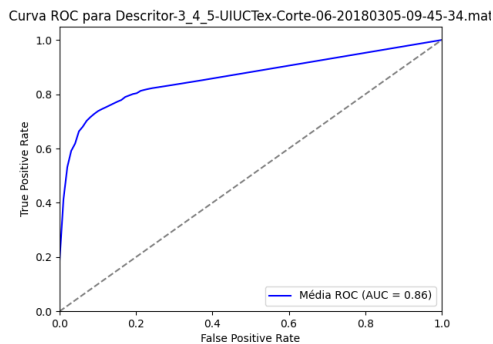


Figura 42 – Saída da curva ROC do Modelo LabelPropagation, Base UIUCTex

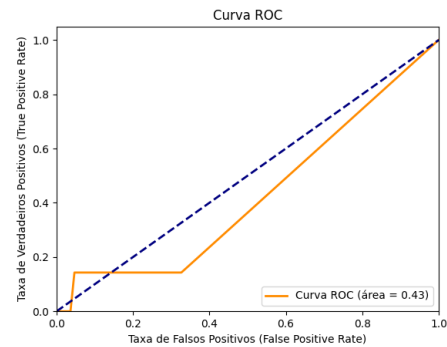


Figura 43 – Saída da curva ROC do Modelo Q Learning, Base UIUCTex

Por outro lado, o Q-Learning não teve sucesso, com uma acurácia de 0% e métricas de precisão, *recall* e F1 Score também de 0%. Esses resultados indicam que o modelo não foi capaz de aprender a partir dos dados disponíveis, o que limita sua utilidade.

O K-Means apresentou métricas de agrupamento, com *Adjusted Rand Index* (ARI) de 0,44 e *Normalized Mutual Information* (NMI) de 0,84. As métricas de *Homogeneity* e *Completeness* foram de 0,80 e 0,88, respectivamente. A matriz de confusão mostra dificuldades em classificar corretamente os dados, evidenciando a necessidade de otimização.

Tabela 14 – Resultados dos Modelos de Machine Learning - Vistex

Modelo	Acurácia	Precisão	Recall	F1 Score
CNN Básica (ADAM)	0,72	0,76	0,72	0,71
KNN	0,80	0,82	0,80	0,79
Q-Learning	0,01	0,01	0,01	0,01
MLPClassifier	0,80	0,81	0,80	0,79
LabelPropagation	0,73	0,69	0,73	0,70

A Tabela 16 compara o desempenho de diversos métodos de classificação aplicados a seis bases de imagens diferentes, destacando a acurácia de cada método. Entre os métodos

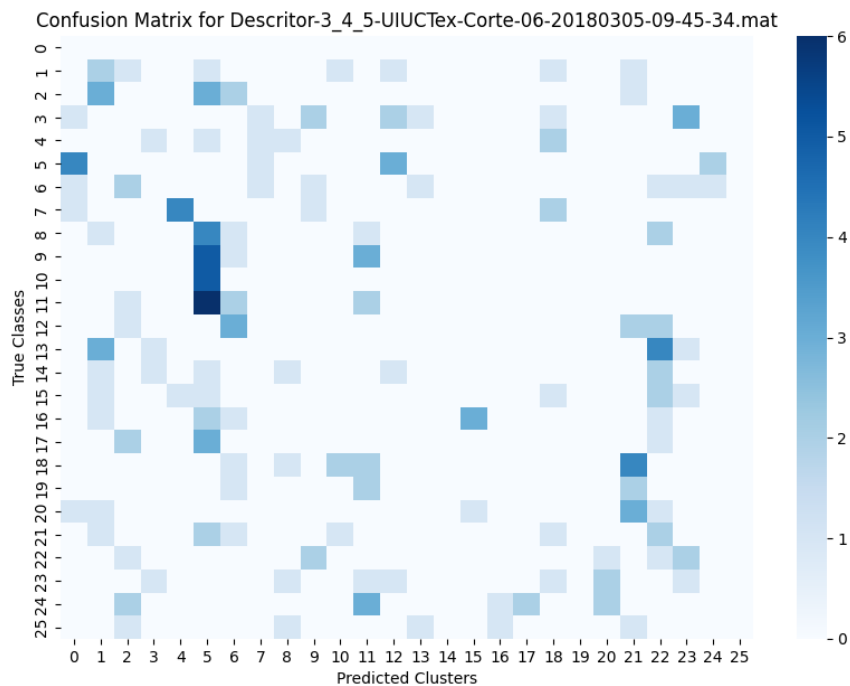


Figura 44 – Saída da Matriz de Confusão do Modelo KMEANS, Base UIUCTex

Tabela 15 – Resultados do K-MEANS Base Vistex

Métrica	Valor
<i>Adjusted Rand Index</i> (ARI)	0,44
<i>Normalized Mutual Information</i> (NMI)	0,84
<i>Homogeneity</i>	0,80
<i>Completeness</i>	0,88
V-Measure	0,8378431393

propostos e os métodos da literatura, são incluídos os modelos de Machine Learning (ML), como CNN, KNN e MLP que foram os três que me apresentaram maior performance, além de técnicas tradicionais de extração de características e classificação.

Os métodos tradicionais, como Primeira Ordem, Wavelets, LBP, e Gabor, utilizam descritores manuais, que envolvem a extração de características baseadas em propriedades específicas das imagens, como texturas ou padrões de frequência. Esses métodos apresentam acurácia variada nas diferentes bases de dados. Por exemplo, LBP apresenta um bom desempenho geral, alcançando até 89,58% de acurácia em Brodatz, enquanto métodos mais simples, como Primeira Ordem, alcançam apenas 40,20% de acurácia na mesma base.

Os métodos da literatura, como os propostos por Backes, Casanova e Bruno (2013), (PEREA; CARLSSON, 2014a), e outros, utilizam um número significativamente maior de descritores, o que pode explicar o aumento da acurácia em muitas das bases testadas. Esses métodos alcançam os melhores resultados em várias bases de imagens, com valores

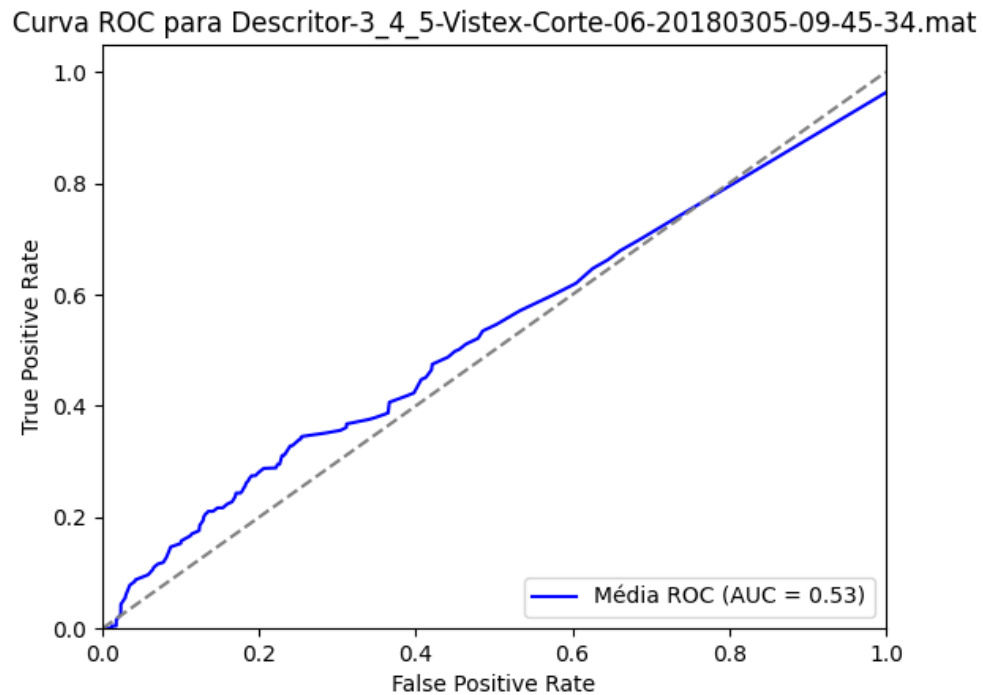


Figura 45 – Saída da curva ROC do Modelo CNN, Base Vistex

Tabela 16 – Resultados dos Métodos Propostos e de Métodos da Literatura para as Seis Bases de Imagens.

Métodos	Nº de Descritores	Acurácia (%)					
		Brodatz	Vistex	CUReT	Outex	UIUCTex	KTH-TIPS
Primeira Ordem	5	40,20	52,66	19,73	54,63	58,30	57,78
Wavelets	36	70,27	72,11	51,00	-	57,70	58,52
LBP	25	89,58	88,89	67,37	77,87	77,00	74,07
Gabor	64	82,49	91,67	82,61	78,60	59,50	80,12
(BACKES; CASA-NOVA; BRUNO, 2013)	108	95,27	98,03	84,32	86,76	86,50	90,37
(PEREA; CARLSSON, 2014a)	215	-	-	95,66 ± 0,45	-	91,23 ± 1,13	94,77 ± 1,3
(ZHANG et al., 2015)	648	-	-	94,44 ± 1,13	-	-	98,86 ± 1,12
(CROSIER; GRIFFIN, 2010)	1.296	-	-	98,6 ± 0,2	-	98,8 ± 0,5	98,5 ± 0,7
(VARMA; ZISSERMAN, 2004)	2.440	-	-	97,43	-	-	-
(RIBEIRO; OLIVEIRA; BARCELOS, 2018)	126	93,04	95,14	88,39	-	-	95,12
Modelo CNN	-	84,26	75,93	87,02	75,11	52,10	88,34
Modelo KNN	-	82,03	82,19	82,76	66,83	55,41	88,17
Modelo MLP	-	80,53	81,39	67,81	73,66	46,09	81,83

de acurácia que chegam a 98,86% em KTH-TIPS e 98,6% em CUReT, destacando-se pelo uso de descritores sofisticados e abordagens personalizadas para cada base de dados.

Em contraste, os modelos de Machine Learning propostos (CNN, KNN e MLP) têm como principal vantagem a capacidade de generalização e aprendizado a partir dos dados, sem a necessidade de extração manual de descritores. Esses modelos podem ser mais eficientes em cenários onde as bases de dados são diversas e complexas, já que conseguem adaptar-se melhor ao contexto das imagens. Por exemplo, a CNN, especializada em

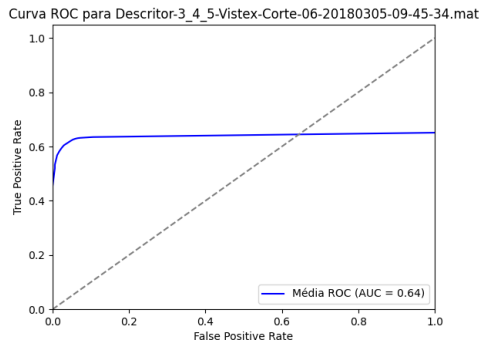


Figura 46 – Saída da curva ROC do Modelo KNN, Base Vistex

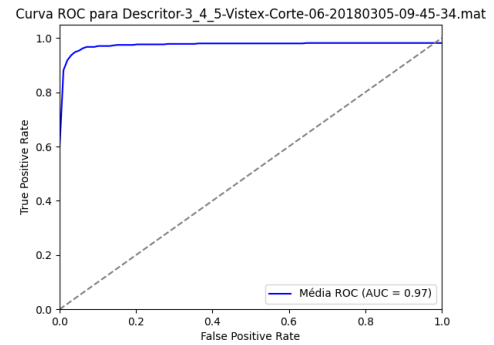


Figura 47 – Saída da curva ROC do Modelo MLP, Base Vistex

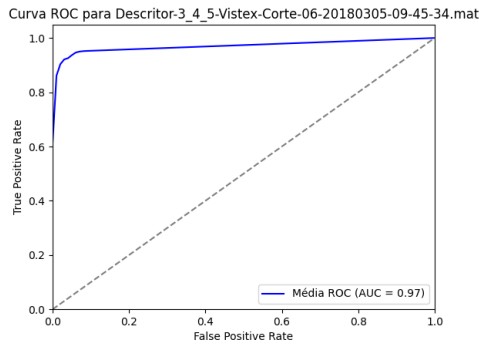


Figura 48 – Saída da curva ROC do Modelo LabelPropagation, Base Vistex

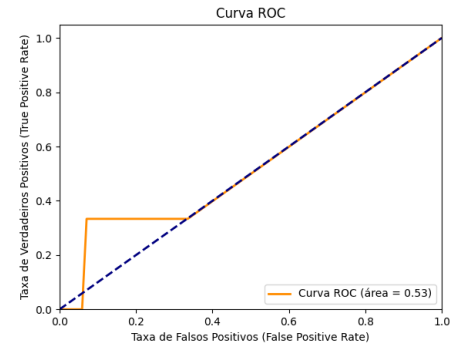


Figura 49 – Saída da curva ROC do Modelo Q Learning, Base Vistex

padrões de imagem, atinge uma acurácia de 87,02% em CURET e 88,34% em KTH-TIPS, o que é competitivo com os métodos tradicionais. Isso mostra que redes neurais convolucionais são muito eficazes para captar características mais profundas e complexas das imagens.

No entanto, em algumas bases, como UIUCTex, a CNN não performa tão bem quanto métodos tradicionais, como os propostos por Backes, Casanova e Bruno (2013) e Crosier e Griffin (2010). Isso pode ocorrer porque, em certos casos, os descritores manuais capturam características específicas da textura ou dos padrões das imagens que os modelos de ML ainda não conseguem capturar sem ajustes mais refinados nos hiperparâmetros ou arquiteturas.

O KNN e o MLP também apresentam resultados variados, com o KNN apresentando um desempenho decente em Vistex (82,19%) e KTH-TIPS (88,17%), enquanto o MLP tem uma acurácia ligeiramente inferior, sugerindo que a escolha do classificador em ML pode impactar o desempenho dependendo da base de dados e da natureza das imagens.

No geral, o Machine Learning pode superar métodos tradicionais em muitos cenários, especialmente quando há uma quantidade significativa de dados e complexidade nas ima-

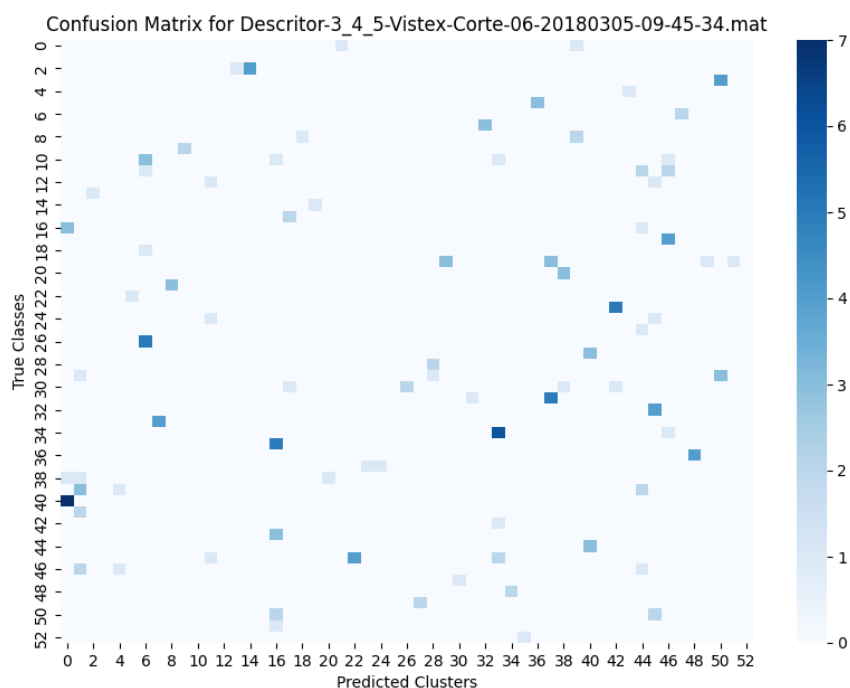


Figura 50 – Saída da Matriz de Confusão do Modelo KMEANS, Base Vistex

gens. Isso ocorre porque, enquanto métodos tradicionais dependem de descritores manuais que podem ser limitados em sua capacidade de capturar variações sutis nas imagens, os modelos de ML aprendem automaticamente padrões mais abstratos e robustos.

Conclusão

O objetivo geral deste trabalho foi comparar as saídas propostas pelos algoritmos descritos por Ribeiro, Oliveira e Barcelos (2018), RIBEIRO (2019) com técnicas e modelos de machine learning, incluindo Modelos Supervisionados, Modelos Não Supervisionados, Modelos de Deep Learning, Modelos de Aprendizado por Reforço, Modelos de Aprendizado Semi-Supervisionado e Modelos de Aprendizado por Transferência. Através dessa comparação, conseguiu-se obter resultados que se aproximaram significativamente dos modelos matemáticos propostos.

A análise detalhada dos experimentos e das métricas revelou o desempenho dos diferentes algoritmos, destacando as vantagens e desvantagens de cada abordagem em contextos específicos. Identificou-se o algoritmo mais eficaz para cada situação e foram discutidas as limitações encontradas durante o estudo. Além disso, apresentou-se possíveis direções para futuras pesquisas que poderão expandir e aprimorar os achados deste trabalho.

Em suma, o desenvolvimento deste trabalho não apenas cumpriu os objetivos estabelecidos, mas também forneceu *insights* valiosos sobre a aplicação de algoritmos de machine learning para problemas de classificação. A conclusão dos objetivos propostos confirmam a eficácia das abordagens adotadas e abre caminho para novas investigações nas áreas de Inteligência Artificial e Processamento de Imagem.

4.1 Principais Contribuições

Este trabalho proporcionou várias contribuições importantes tanto no campo da Inteligência Artificial quanto no de aprendizado de máquina. A seguir, estão as principais contribuições deste estudo:

❑ **Desenvolvimento e comparação de modelos de aprendizado de máquina:**

O estudo forneceu uma análise detalhada de diferentes categorias de algoritmos de aprendizado de máquina, incluindo Modelos Supervisionados, Não Supervisionados, Deep Learning, Aprendizado por Reforço e Transferência. Ao comparar a perfor-

mance desses modelos, foi possível identificar as situações em que cada um deles se destaca, fornecendo diretrizes práticas para sua aplicação em cenários reais.

- ❑ **Integração de técnicas matemáticas e machine learning:** Uma contribuição central foi a aplicação bem-sucedida de modelos de machine learning para resolver problemas matemáticos complexos, atingindo resultados comparáveis aos obtidos por meio de métodos matemáticos tradicionais. Isso reforça o papel do aprendizado de máquina como uma ferramenta eficaz para lidar com grandes volumes de dados e identificar padrões complexos.
- ❑ **Proposta de metodologias otimizadas para classificação:** Com base nos resultados experimentais, o trabalho sugeriu ajustes e aprimoramentos nos parâmetros dos algoritmos, como no KNN, visando aumentar sua precisão e aplicabilidade em diferentes contextos de classificação.
- ❑ **Incorporação de novas abordagens:** Ao incluir discussões sobre modelos emergentes, como as LLMs, o estudo explorou possíveis caminhos futuros, destacando o potencial dessas tecnologias para melhorar ainda mais o desempenho dos algoritmos de aprendizado de máquina.
- ❑ **Avaliação de métricas específicas:** Foram implementadas e discutidas métricas detalhadas para avaliar a performance dos algoritmos, incluindo acurácia, precisão, *recall*, *F1-Score*, e métricas para clustering, como o *Adjusted Rand Index* e a *Normalized Mutual Information*. Essa análise ofereceu uma visão robusta da performance dos modelos em diferentes cenários.

Essas contribuições demonstram o impacto do trabalho ao avançar o estado da arte em aprendizado de máquina e análise de dados, oferecendo resultados práticos e teóricos que podem ser aproveitados por pesquisadores e profissionais da área.

4.2 Trabalhos Futuros

As possíveis melhorias e direções para o desenvolvimento futuro do método proposto podemos abordar para aprimorar:

Modelos de Fine-Tuning: Uma abordagem promissora para melhorar a precisão e a adaptabilidade dos modelos é o uso de técnicas de fine-tuning. O fine-tuning permite ajustar modelos pré-treinados em um conjunto de dados específico, melhorando a performance em tarefas relacionadas ao contexto do problema. Implementar fine-tuning em modelos de machine learning pode resultar em melhores desempenhos ao adaptar os modelos às particularidades dos dados utilizados.

IA Generativa: Outra direção interessante para futuras pesquisas é a exploração de modelos de Inteligência Artificial Generativa. Modelos generativos, como Redes Adversariais Generativas (GANs) e Modelos de Difusão, podem ser utilizados para criar novos dados sintéticos que ajudem a enriquecer o conjunto de treinamento. Esses modelos podem gerar variações realistas das amostras existentes, contribuindo para a robustez e a generalização dos algoritmos de machine learning.

Integração de Métodos Híbridos: Considerar a integração de métodos híbridos que combinam diferentes tipos de modelos, como modelos supervisionados e não supervisionados, pode oferecer uma abordagem mais abrangente para a análise e classificação de dados. Esses métodos híbridos podem explorar a força de cada tipo de modelo, proporcionando melhores resultados e maior flexibilidade.

Aprimoramento das Métricas de Avaliação: Para uma avaliação mais detalhada, é recomendável a implementação de métricas adicionais e técnicas de análise mais sofisticadas. Incorporar métricas avançadas e ajustadas ao contexto específico dos dados pode oferecer uma visão mais completa da performance dos modelos e ajudar na identificação de áreas para melhoria.

Expansão do Conjunto de Dados: A ampliação e diversificação do conjunto de dados utilizado para treinamento e validação dos modelos pode resultar em melhorias significativas. A inclusão de novos exemplos e a ampliação da variedade dos dados podem ajudar a melhorar a generalização dos modelos e a lidar com casos de uso mais variados.

Modelos de Linguagem de Grande Escala: Uma área emergente que pode adicionar valor significativo à pesquisa é o uso de LLMs. Modelos como o GPT-3 da OpenAI (OPENAI, 2020) têm mostrado avanços impressionantes em processamento de linguagem natural e geração de texto. Esses modelos são capazes de entender e gerar linguagem humana com um nível de sofisticação que pode complementar e aprimorar as técnicas tradicionais de machine learning.

Integrar LLMs no processo de classificação e análise pode trazer benefícios como a capacidade de lidar com contextos complexos e a adaptação a novos padrões de dados com uma eficiência notável. No futuro, a exploração de como esses modelos podem ser usados para melhorar a precisão e a eficácia das análises será uma linha promissora de investigação.

Em resumo, os trabalhos futuros podem focar na implementação de técnicas de fine-tuning e modelos generativos para aprimorar o desempenho dos algoritmos, explorar métodos híbridos, refinar as métricas de avaliação e expandir os conjuntos de dados. Essas abordagens não apenas superarão as limitações identificadas, mas também abrirão novas possibilidades para avanços na área de Inteligência Artificial e Machine Learning.

4.3 Contribuições em Produção Bibliográfica

Durante o desenvolvimento desta pesquisa, diversas contribuições relevantes foram realizadas na forma de produção bibliográfica. Além disso, o trabalho apresentado neste TCC é um desdobramento de uma iniciação científica anterior, que tinha como objetivo adaptar o código em MATLAB proposto por RIBEIRO (2019) para Python, visando a classificação de imagens. Essa pesquisa inicial serviu de base para o presente estudo, no qual algoritmos de machine learning foram aplicados para a classificação de imagens. A seguir, estão listadas as publicações e apresentações resultantes:

- ❑ **Iniciação Científica:** “Adaptação de Código MATLAB para Python: Aplicação na Classificação de Imagens”, desenvolvida durante a iniciação científica no ano de 2022. O projeto propôs a adaptação do código MATLAB de RIBEIRO (2019) para a linguagem Python, focando na classificação de imagens. Este trabalho foi um ponto de partida para o desenvolvimento deste Trabalho de Conclusão de Curso, onde algoritmos de machine learning foram utilizados para expandir e melhorar os resultados de classificação.

Referências

- ABADI, M. et al. **TensorFlow: Large-scale machine learning on heterogeneous systems**. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>. Citado na página 27.
- ADAMS, H.; MOY, M. Topology applied to machine learning: From global to local. **Frontiers in Artificial Intelligence**, Frontiers Media SA, v. 4, p. 668302, 2021. Citado na página 31.
- ALURA. **O que é Aprendizado Semi-Supervisionado?** 2023. <<https://www.alura.com.br/artigos/machine-learning-aprendizado-semi-supervisionado>>. Acesso em: 19 de fevereiro de 2025. Citado na página 28.
- BACKES, A. R. P.; CASANOVA, M. A.; BRUNO, O. M. Method for texture analysis based on fractal descriptors. **Pattern Recognition**, v. 46, n. 2, p. 501–512, 2013. Citado 3 vezes nas páginas 60, 61 e 62.
- BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006. Citado na página 22.
- BRODATZ, P. **Textures: A Photographic Album for Artists and Designers**. New York: Dover Publications, 1966. Citado na página 37.
- CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. **BMC Genomics**, v. 22, n. 1, p. 1–13, 2021. Citado na página 39.
- CROSIER, M.; GRIFFIN, L. D. Using basic image features for texture classification. In: **International Journal of Computer Vision**. [S.l.: s.n.], 2010. v. 88, p. 447–460. Citado 2 vezes nas páginas 61 e 62.
- DANA, K. J. et al. Reflectance and texture of real-world surfaces. **ACM Transactions on Graphics (TOG)**, v. 18, n. 1, p. 1–34, 1999. Citado na página 38.
- Dicionário Tec. **Machine Learning**. 2022. Acessado em: 3 de outubro de 2024. Disponível em: <<https://dicionariotec.com/posts/machine-learning>>. Citado na página 22.
- FAWCETT, T. An introduction to roc analysis. **Pattern Recognition Letters**, v. 27, n. 8, p. 861–874, 2006. Citado na página 43.

FIX, E.; JR., J. L. H. **Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties**. [S.l.]: USA Air Force School of Aviation Medicine, 1951. Citado na página 24.

GARFUNKEL, J. **Introduction to Topology: Surfaces and Beyond**. 2nd. ed. [S.l.]: Springer, 2021. 45-60 p. Citado na página 19.

GONZALEZ, R.; WOODS, R. **Processamento Digital De Imagens**. ADDISON WESLEY BRA, 2009. ISBN 9788576054016. Disponível em: <<https://books.google.com.br/books?id=r5f0RgAACAAJ>>. Citado 2 vezes nas páginas 30 e 31.

GUPTA, A. **Transfer Learning for Cat and Dog Classifier**. 2020. <<https://www.linkedin.com/pulse/transfer-learning-cat-dog-classifier-anshika-gupta/>>. Acesso em: 19 de fevereiro de 2025. Citado 2 vezes nas páginas 29 e 30.

HAYMAN, E. et al. On the significance of real-world conditions for material classification. In: **European Conference on Computer Vision (ECCV)**. [S.l.]: Springer, 2004. p. 253–266. Citado na página 38.

HUBERT, L.; ARABIE, P. Comparing partitions. **Journal of Classification**, v. 2, n. 1, p. 193–218, 1985. Citado na página 41.

KRIGER, D. **O QUE É PYTHON, PARA QUE SERVE E POR QUE APRENDER?** 2022. <https://kenzie.com.br/blog/o-que-e-python/>. Accessed: 26/01/2022. Citado na página 20.

LAZEBNIK, S.; SCHMID, C.; PONCE, J. A sparse texture representation using local affine regions. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [S.l.: s.n.], 2005. v. 27, n. 8, p. 1265–1278. Citado na página 38.

LEARN, S. **scikitlearn**. 2022. <https://scikit-learn.org>. Accessed: 26/01/2022. Citado na página 21.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group, 2015. Citado na página 30.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 27.

LI, L.-J.; FEI-FEI, L. What, where and who? classifying events by scene and object recognition. In: **2007 IEEE 11th International Conference on Computer Vision**. [S.l.: s.n.], 2007. p. 1–8. Citado na página 38.

LU, H. **Study on Image Quality Improvement Methods for Underwater Imaging Systems**. Tese (Doutorado) — Kyushu University, 2014. Disponível em: <<https://ci.nii.ac.jp/naid/500000916504>>. Citado na página 31.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. v. 1, p. 281–297, 1967. Citado na página 25.

MANNING, C. D.; RAGHAVAN, P. **Introduction to Information Retrieval**. Cambridge: Cambridge University Press, 2008. Disponível em: <<https://nlp.stanford.edu/IR-book/html/>>. Citado na página 44.

MARTINS, F. C. M.; RILLO, M. Aplicação de redes neurais em processamento de imagens. detecção de bordas através de cooperação e competição. 1992. Citado na página 32.

MATPLOTLIB. **matplotlib**. 2022. <https://matplotlib.org>. Accessed: 26/01/2022. Citado na página 21.

MIT Media Lab. **The Vision Texture Database**. 2009. Disponível em: <<http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>>. Citado na página 38.

NEVES, E. C. **Aprendizado por Reforço #1 — Introdução**. 2020. Publicado em: Turing Talks. Disponível em: <<https://medium.com/turing-talks/aprendizado-por-reforço-1-introdução-7382ebb641ab>>. Acesso em: 19 de fevereiro de 2025. Citado na página 26.

NumPy Developers. **NumPy: The fundamental package for numerical computation in Python**. 2023. <<https://numpy.org>>. Accessed: 2024-10-01. Citado na página 21.

OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 7, p. 971–987, 2002. Citado na página 38.

OPENAI. **OpenAI Gym: A Toolkit for Developing and Comparing Reinforcement Learning Algorithms**. 2016. Disponível em: <<https://github.com/openai/gym>>. Citado na página 26.

_____. **GPT-3: Language Models are Few-Shot Learners**. 2020. <https://www.openai.com/research/gpt-3>. Citado na página 66.

OPENCV. **opencv**. 2022. <https://opencv.org>. Accessed: 26/01/2022. Citado na página 21.

PEDREGOSA, F. et al. **Scikit-learn: Machine Learning in Python**. 2011. 2825-2830 p. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>>. Citado na página 24.

_____. **Scikit-learn: Machine Learning in Python - KMeans**. 2011. 2825-2830 p. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>>. Citado na página 25.

_____. **Scikit-learn: Machine Learning in Python - LabelPropagation**. 2011. 2825-2830 p. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelPropagation.html>. Citado na página 29.

_____. **Scikit-learn: Machine Learning in Python - MLP Classifier**. 2011. 2825-2830 p. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>. Citado na página 30.

- PEREA, J. A.; CARLSSON, G. A klein-bottle-based dictionary for texture representation. **International Journal of Computer Vision**, Springer US, v. 107, n. 1, p. 75–97, 2014. ISSN 0920-5691. Disponível em: <<https://doi.org/10.1007/s11263-013-0676-2>>. Citado 3 vezes nas páginas 31, 60 e 61.
- PEREA, L.; CARLSSON, G. A simple algorithm for homology-based segmentation of high-dimensional data sets. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v. 76, n. 4, p. 651–671, 2014. Citado 2 vezes nas páginas 7 e 20.
- PRETO, U. F. de O. **Inteligência Artificial e Deep Learning**. 2023. <<https://www2.decom.ufop.br/imobilis/inteligencia-artificial-e-deep-learning/>>. Acesso em: 19 de fevereiro de 2025. Citado na página 27.
- Python Software Foundation. **Python: A Programming Language**. 2023. <<https://www.python.org/doc/>>. Accessed: 2024-10-01. Citado na página 20.
- RIBEIRO, T. P. **Caracterização de texturas com o auxílio de redes complexas, projeções topológicas e padrões semânticos**. 116 p. Tese (Tese (Doutorado em Ciência da Computação)) — Universidade Federal de Uberlândia, Uberlândia MG, Março 2019. Disponível em <<https://repositorio.ufu.br/handle/123456789/25340>>. Citado 4 vezes nas páginas 15, 20, 64 e 67.
- RIBEIRO, T. P.; OLIVEIRA, A. L. N. de; BARCELOS, C. A. Z. Texture characterization via projections onto a klein bottle topology. In: **2018 25th IEEE International Conference on Image Processing (ICIP)**. [S.l.]: IEEE, 2018. p. 2097–2101. Citado 9 vezes nas páginas 6, 15, 16, 17, 20, 31, 35, 61 e 64.
- ROSENBERG, A.; HIRSCHBERG, J. V-measure: A conditional entropy-based external cluster evaluation measure. In: **Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning**. [s.n.], 2007. p. 410–420. Disponível em: <<https://www.aclweb.org/anthology/D07-1043>>. Citado na página 42.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning Representations by Back-propagating Errors**. [S.l.]: Nature, 1986. v. 323. 533-536 p. Citado na página 30.
- RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3rd. ed. Upper Saddle River, NJ: Prentice Hall, 2010. Citado na página 22.
- SERVICES, A. W. **A Diferença entre Aprendizado de Máquina Supervisionado e Não Supervisionado**. 2023. <<https://aws.amazon.com/pt/compare/the-difference-between-machine-learning-supervised-and-unsupervised/>>. Acesso em: 19 de fevereiro de 2025. Citado 2 vezes nas páginas 23 e 24.
- SOKOLOVA, M.; LAPALME, G. **A systematic analysis of performance measures for classification tasks**. [S.l.]: Elsevier, 2009. v. 45. 427-437 p. Citado 2 vezes nas páginas 39 e 40.
- SONKA, M.; HLAVAC, V.; BOYLE, R. **Image Processing, Analysis, and Machine Vision**. [S.l.]: Cengage Learning, 2007. Citado na página 30.

- STREHL, A.; GHOSH, J. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. **Journal of Machine Learning Research**, v. 3, p. 583–617, 2002. Disponível em: <<http://www.jmlr.org/papers/volume3/strehl02a/strehl02a.pdf>>. Citado na página 41.
- SZELISKI, R. **Computer Vision: Algorithms and Applications**. [S.l.]: Springer Science & Business Media, 2010. Citado na página 30.
- TENSORFLOW. **tensorflow**. 2022. <https://www.tensorflow.org/?hl=pt-br>. Accessed: 26/01/2022. Citado na página 22.
- TRONCO, M. L.; PORTO, A. J. V. **Sistema de reconhecimento de imagens, baseado no modelo GSN de rede neural**. Tese (Tese (Doutorado em Engenharia)) — Universidade de São Paulo, São Carlos SP, 1999. Disponível em <https://www.teses.usp.br/teses/disponiveis/18/18135/tde-14062024-161019/publico/Tronco_Mario_tese.pdf>. Citado na página 32.
- VARMA, M.; ZISSERMAN, A. A statistical approach to texture classification from single images. In: **International Journal of Computer Vision**. [S.l.: s.n.], 2004. v. 62, p. 61–81. Citado na página 61.
- WATKINS, C. J.; DAYAN, P. Q-learning. **Machine Learning**, v. 8, p. 279–292, 1992. Citado na página 26.
- ZHANG, J. et al. Robust shape matching using the level sets method. **IEEE Transactions on Image Processing**, v. 24, n. 5, p. 1472–1483, 2015. Citado na página 61.
- ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. Semi-supervised learning using gaussian fields and harmonic functions. **Proceedings of the 20th International Conference on Machine Learning (ICML)**, p. 912–919, 2003. Citado na página 29.