



**INSTITUTO
FEDERAL**
Ceará

**MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
DEPARTAMENTO DE TELEMÁTICA**

RELATÓRIO DE ATIVIDADE 2024.1

**IMPLEMENTAÇÃO E ANÁLISE DE EXTRATORES DE ATRIBUTOS E
CLASSIFICADORES NO CONTEXTO DE VISÃO COMPUTACIONAL**

**AVALIAÇÃO DA EFICÁCIA E EFICIÊNCIA DOS EXTRATORES E
CLASSIFICADORES UTILIZADOS**

Relatório da Atividade para disciplina de
Visão Computacional sob orientação do
Professor Dr. Reboucas Filho Pedro Pe-
drosa.

Marcelo de Araújo
Campus Fortaleza
Junho / 2024

RESUMO

Este relatório apresenta a implementação de diversas técnicas de extração de características, sendo o processo de identificar e selecionar características relevantes de dados brutos ou de entrada e classificação de imagens utilizando bibliotecas como NumPy, scikit-learn e TensorFlow em Python. Foram utilizadas técnicas de extração de atributos como análise de componentes principais (PCA), análise discriminante linear (LDA), fatoração de matriz não-negativa (NMF), decomposição de valor singular (SVD) e histograma de gradientes orientados (HOG), cada uma responsável por extrair características relevantes das imagens para posterior classificação.

Além disso, o deep learning, uma abordagem poderosa que utiliza redes neurais profundas para aprender representações de alto nível dos dados, também foi explorado neste trabalho. Uma variedade de modelos de aprendizado de máquina, incluindo SVM, Random Forest, Naive Bayes, KNN, MLP e árvores de decisão, foi empregada para a classificação das imagens. Além disso, redes neurais profundas, como as construídas com a biblioteca Keras em cima do TensorFlow, foram utilizadas para explorar ainda mais a capacidade de aprendizado de representações complexas das imagens.

Cada etapa do processo, desde a extração de características até a classificação, é fundamental no processamento de imagens e desempenha um papel crucial na identificação e classificação de padrões visuais. O relatório aborda os conceitos fundamentais dessas técnicas, explorando suas aplicações e importância no campo do processamento de imagens, fornecendo uma visão abrangente das práticas contemporâneas nesse domínio.

1 INTRODUÇÃO

Dentro da área de visão computacional, a extração de atributos refere-se ao processo de identificar e representar características distintivas em uma imagem relevante para uma determinada tarefa. Essas características podem incluir bordas, texturas, formas, padrões e outras propriedades visuais utilizadas para descrever e distinguir objetos ou regiões de interesse em uma cena. Em seu livro seminal sobre visão computacional, (FORSYTH; PONCE, 2003) destacam a importância da extração de atributos, descrevendo-a como "o processo de transformar uma imagem em uma representação mais compacta que é mais fácil de analisar". Essa representação compacta, composta por atributos relevantes, é essencial para reduzir a complexidade computacional e facilitar a interpretação automática de imagens por algoritmos de visão computacional. Já os classificadores referem-se a algoritmos ou modelos estatísticos que atribuem rótulos ou categorias a objetos ou regiões de interesse em uma imagem, analisando as características extraídas das imagens para determinar a classe à qual cada objeto pertence. Segundo (SZELISKI, 2011), "os classificadores são uma ferramenta essencial em visão computacional, permitindo a automação de tarefas como reconhecimento de objetos e detecção de padrões em imagens." A

importância de tais ferramentas reside na capacidade de automatizar tarefas de reconhecimento e categorização, permitindo a análise eficiente de grandes conjuntos de dados visuais.

Este relatório apresenta a utilização de diversas técnicas de processamento de imagens, concentrando-se na extração de características e classificação de imagens utilizando bibliotecas como NumPy, scikit-learn e TensorFlow em Python. Foram exploradas técnicas como análise de componentes principais (PCA), análise discriminante linear (LDA), fatoração de matriz não-negativa (NMF), decomposição de valor singular (SVD) e histograma de gradientes orientados (HOG) para extrair características relevantes das imagens, seguidas por uma variedade de modelos de aprendizado de máquina, incluindo SVM, Random Forest, Naive Bayes, KNN, MLP e árvores de decisão, para a classificação das imagens. Além disso, foram exploradas redes neurais profundas, construídas com a biblioteca Keras sobre o TensorFlow, para aprender representações complexas dos dados. Cada etapa do processo, desde a extração de características até a classificação, desempenha um papel crucial na identificação e classificação de padrões visuais, contribuindo para uma compreensão abrangente das práticas contemporâneas no campo do processamento de imagens, permitindo a visualização e distinção das diferentes regiões em uma imagem rotulada.

2 OBJETIVO GERAL

O objetivo principal deste estudo é investigar a eficácia de diferentes técnicas de extração de atributos combinadas com vários algoritmos de classificação no reconhecimento de números de placas de veículos em imagens. A análise é realizada utilizando um conjunto de dados rotulados. Este estudo avalia a precisão e a robustez de diversas combinações de técnicas de extração de características e classificadores, incluindo modelos de aprendizado profundo, com o intuito de identificar a melhor abordagem para a tarefa de reconhecimento óptico de caracteres (OCR), sendo uma área crucial em sistemas de visão computacional, com inúmeras aplicações práticas, pois, conforme (S, 2016) exemplifica, o reconhecimento óptico de caracteres (OCR) é uma tecnologia que permite a conversão mecânica ou eletrônica de diferentes tipos de documentos em dados codificado por máquina editáveis e pesquisáveis. A eficácia dos sistemas OCR depende significativamente da qualidade das características extraídas das imagens e da eficiência dos algoritmos de classificação utilizados. Portanto, uma avaliação sistemática das técnicas de extração de características e dos classificadores é essencial para o desenvolvimento de sistemas robustos e precisos.

Neste estudo, são comparadas várias técnicas de extração de características, como análise de componentes principais (PCA), análise discriminante linear (LDA), fatoração de matriz não-negativa (NMF), decomposição de valor singular (SVD) e histograma de gradientes orientados (HOG) para extrair características relevantes, seguidas por uma variedade de modelos classificadores, incluindo SVM, Random Forest, Naive Bayes, KNN, MLP e árvores de decisão, para a classificação das amostras. A combinação dessas técnicas e algoritmos é analisada

em termos de precisão, robustez e desempenho computacional. Os resultados obtidos visam fornecer percepções sobre as melhores práticas na combinação de técnicas de extração de características e algoritmos de classificação para OCR, contribuindo para o avanço na área de visão computacional aplicada a reconhecimento de caracteres.

3 OBJETIVOS ESPECÍFICOS

A proposta deste trabalho é utilizar extratores de atributos, modelos classificadores e uma rede neural para o reconhecimento óptico de caracteres, tais métodos e classes estão disponíveis no **scikit-learn**, para leitura dos dados advindo de um arquivo .txt **Numpy** e para a construção do requisito proposto no trabalho de deep learning é utilizado métodos e classes do **tensorflow**. O trabalho oferece o desenvolvimento de um algoritmo funcional para aplicação de reconhecimento óptico de caracteres.

Descrição das Funcionalidades:

- Carregamento dos dados: implementação de um conjunto de instruções para leitura dos dados a partir de um arquivo .txt utilizando Numpy, transformando os dados em imagens de 35x35 píxeis.
- Extração de atributos: aplicação de várias técnicas de extração de atributos, incluindo PCA, LDA, NMF, SVD e HOG, para reduzir a dimensionalidade dos dados e extrair características relevantes das imagens.
- Divisão em treino e teste: divisão dos dados extraídos em conjuntos de treino e teste, mantendo uma proporção de 80% para treino e 20% para teste, garantindo uma avaliação justa dos modelos.
- Modelos Classificadores: Utilização de vários classificadores do scikit-learn, incluindo SVM, Random Forest, Decision Tree, Naive Bayes, Gaussian, KNN e MLP, para treinar e avaliar os dados.
- Modelo de Deep Learning: Criação de um modelo de rede neural usando Keras, composto por camadas de entrada, flatten, duas camadas dense (uma com ReLU e outra com softmax), compilado com o otimizador Adam e uma função de perda.
- Integração do modelo Keras com scikit-learn: implementação de uma classe para integrar modelos Keras com a API do scikit-learn, permitindo o uso de funcionalidades do scikit-learn.

Além disso, o processo de desenvolvimento do algoritmo de proposto foi cuidadosamente planejado e implementado visando a eficiência e a precisão. A integração harmoniosa

entre as diferentes bibliotecas, como NumPy para manipulação de dados, scikit-learn para técnicas de aprendizado de máquina e TensorFlow para deep learning, proporciona uma abordagem abrangente e escalável para lidar com problemas de reconhecimento de caracteres.

4 METODOLOGIA

A metodologia adotada fundamenta-se na utilização e aplicação de algoritmos extratores e classificadores, seguindo uma abordagem sistemática para investigar a aplicação de técnicas avançadas de aprendizado de máquina instrumentalizadas para reconhecimento óptico de caracteres. A construção do estudo foi organizada em múltiplas fases metodológicas, uma prática consolidada e bem fundamentada na pesquisa científica, especialmente nas áreas de aprendizado de máquina e análise de dados. Esta abordagem sistemática assegura uma organização clara e sequencial do processo de pesquisa, garantindo que cada fase seja meticulosamente planejada e executada. Conforme discutido por (CRESWELL, 2014), uma estrutura metodológica bem definida permite maior rigor e reprodutibilidade na pesquisa. Adicionalmente, (YIN, 2018) enfatiza que a segmentação do estudo em etapas distintas facilita a análise e a interpretação dos resultados. Além disso, tal estrutura facilita a replicação do estudo e a verificação dos resultados por parte de outros pesquisadores, contribuindo para a robustez e a transparência científica.

4.1 Pré processamento dos dados

O estudo iniciou-se com o carregamento e pré processamento dos dados a partir de um arquivo de texto contendo 3352 linhas. Cada linha do arquivo representa um conjunto de atributos de caracteres numéricos, seguido por seu respectivo rótulo. Utilizou-se a função `np.loadtxt` da biblioteca NumPy para a leitura dos dados, eliminando os espaços entre os caracteres de cada linha. O último caractere de cada linha foi identificado como o rótulo do objeto correspondente. Após o carregamento dos dados, procedeu-se à separação dos atributos e dos rótulos. A matriz de atributos foi redimensionada para formar imagens de 35 x 35 píxeis, resultando em um tensor tridimensional. Os rótulos foram extraídos como o último elemento de cada linha dos dados carregados.

4.2 Extração de Atributos

Neste estudo, diversos métodos de extração de atributos e técnicas de redução de dimensionalidade foram utilizados para a análise de dados obtidos a partir das imagens resultantes no pré processamento. A extração de atributos é uma etapa crucial nessa análise, pois permite transformar dados brutos em representações adequadas para a tarefa de classificação. Neste trabalho, empregaram-se duas abordagens principais para a extração de atributos: Histogram of Oriented Gradients (HOG) e transformações baseadas em decomposições de matrizes.

Segundo (LI *et al.*, 2021), a decomposição de matrizes é uma ferramenta versátil

com diversas aplicações em áreas como otimização, controle de sistemas, processamento de sinais e aprendizado de máquina. Ao explorar as propriedades matemáticas das matrizes e suas decomposições, consigo extrair informações valiosas sobre os dados subjacentes, revelando padrões e características essenciais para a análise.

O HOG é uma técnica que utiliza informações sobre os gradientes de intensidade e a orientação dos píxeis na imagem. Para cada imagem I_i , foi computado um vetor de características. Para reduzir a dimensionalidade dos dados e melhorar potencialmente o desempenho dos classificadores, aplicaram-se quatro técnicas de redução de dimensionalidade: a análise de componentes principais (PCA), estratégia que projeta os dados em um espaço de menor dimensionalidade, maximizando a variância retida, a análise discriminante linear (LDA), mecanismo supervisionado que maximiza a separação entre as classes, fatoração em matrizes não negativas (NMF), técnica que fatoriza a matriz de dados em duas matrizes de menor dimensão com valores não-negativos e a decomposição em valores singulares (SVD), procedimento que reduz a dimensionalidade dos dados ao decompor a matriz original em componentes principais.

4.3 Treino e Teste

Primeiramente, destaco que a divisão entre treino e teste visa avaliar fundamentalmente a capacidade de generalização do modelo, isto é, sua habilidade de realizar previsões precisas em dados não observados anteriormente. Nesse sentido, é crucial seguir as orientações de (GONZALEZ; WOODS; EDDINS, 2008) para uma avaliação apropriada dos algoritmos de visão computacional, que enfatizam a necessidade de dividir os dados em conjuntos de treinamento e teste, assegurando a capacidade do modelo de generalizar para novos exemplos. Para isso, adotei a prática comum de particionar o conjunto de dados em dois subconjuntos mutuamente exclusivos: o conjunto de treinamento, utilizado para treinar o modelo, e o conjunto de teste, empregado para avaliar o desempenho do modelo em dados não observados.

Nesse contexto, cada conjunto de atributos derivados das técnicas de extração de características (PCA, LDA, HOG, NMF e SVD) foi separadamente dividido em conjuntos de treinamento e teste, seguindo uma proporção convencional de 80% para treino e 20% para teste. Além disso, é imprescindível ressaltar que a divisão foi realizada de forma estratificada, garantindo a manutenção da distribuição das classes-alvo em ambos os conjuntos. Tal prática é especialmente relevante em problemas de classificação, onde é essencial garantir que o modelo seja treinado e testado em dados representativos de todas as classes.

Adicionalmente, a escolha do parâmetro **random_state = 42** foi feita para garantir a reprodutibilidade dos resultados, assegurando que a mesma divisão entre treino e teste seja obtida em diferentes execuções do código. Essa medida é fundamental para assegurar a consistência dos resultados e facilitar a comparação entre diferentes modelos e abordagens.

4.4 Classificadores

Neste estudo, concentro a análise na aplicação de técnicas de aprendizado de máquina para a classificação de dados, utilizando uma variedade de algoritmos de classificação e métodos de redução de dimensionalidade. O objetivo principal é comparar o desempenho dos classificadores em diferentes cenários, a fim de identificar a abordagem mais eficaz para a tarefa em questão. Cada um desses classificadores foi avaliado quanto ao seu desempenho em cenários variados, permitindo uma análise comparativa que visa identificar no estudo proposto.

Os modelos de classificação selecionados para este estudo foram escolhidos com base em sua ampla utilização em problemas de classificação e sua capacidade comprovada de lidar com conjuntos de dados complexos. Cada modelo foi avaliado quanto ao seu desempenho em diferentes cenários para permitir uma análise comparativa significativa.

- **Support Vector Machine (SVM):** O SVM é um classificador robusto que busca encontrar o hiperplano que melhor separa as classes. Sua capacidade de lidar com conjuntos de dados de alta dimensionalidade e sua robustez em relação a outliers foram consideradas vantagens importantes para nossa análise.
- **Random Forest:** O Random Forest é um conjunto de árvores de decisão que utiliza o método de bagging para melhorar a precisão do modelo. Sua capacidade de lidar com conjuntos de dados grandes e complexos, além de sua resistência ao overfitting, foram fatores determinantes para sua inclusão neste estudo.
- **Decision Tree:** As Decision Trees são classificadores baseados em regras que segmentam os dados de forma iterativa. Sua interpretabilidade e capacidade de lidar com conjuntos de dados heterogêneos foram consideradas vantagens significativas para nossa análise.
- **Naive Bayes Gaussian:** O Naive Bayes Gaussian é baseado no teorema de Bayes e assume a independência entre as características. Sua simplicidade e eficácia em conjuntos de dados de alta dimensionalidade foram consideradas vantagens para sua inclusão neste estudo.
- **K-Nearest Neighbors (KNN):** O KNN classifica os dados com base na proximidade dos pontos de dados. Sua simplicidade e robustez em relação a ruídos foram consideradas vantagens importantes para nossa análise.
- **Multi-layer Perceptron (MLP):** O MLP é uma rede neural artificial com uma ou mais camadas ocultas. Sua capacidade de aprender representações complexas dos dados e sua flexibilidade foram consideradas vantagens importantes para a análise.

A diversidade de modelos escolhidos oferece uma ampla gama de abordagens para a tarefa de classificação de dados, cada uma trazendo consigo suas próprias vantagens e limitações. Conforme mencionado por (GEETHA; SENDHILKUMAR, 2023), a construção de modelos

classificadores é uma arte delicada, onde cada variável e sua interação devem ser consideradas com cuidado, como peças em um quebra-cabeça, para formar um todo coeso e preditivo. Assim, a análise comparativa desses modelos em diferentes cenários se torna crucial para determinar a estratégia mais eficaz para a aplicação em questão.

4.5 Deep Learning

Este estudo apresenta a implementação de um modelo de Deep Learning baseado na arquitetura conhecida como Rede Neural Artificial (RNA), conforme descrito por (BISHOP, 2016), em um contexto cada vez mais interconectado, as redes neurais artificiais surgem como uma ferramenta crucial para a análise e interpretação de conjuntos de dados complexos, possibilitando a identificação de padrões ocultos e a tomada de decisões inteligentes. Especificamente, este estudo adota uma abordagem utilizando uma rede neural densa, uma estrutura amplamente empregada em tarefas de classificação. Nesse tipo de rede, os neurônios são organizados em camadas, com cada neurônio recebendo entradas, realizando operações matemáticas sobre elas e transmitindo os resultados para a próxima camada.

A arquitetura do modelo começa com uma camada de entrada, que recebe os dados de entrada de uma forma especificada pelo parâmetro **input_shape**. Em seguida, os dados são achatados para uma única dimensão através da camada **Flatten()**. Isso é necessário para que os dados possam ser processados por camadas densamente conectadas. Após isso, há uma camada densa com 128 neurônios ativados pela função de ativação **ReLU** (Rectified Linear Activation), que ajuda na aprendizagem de representações complexas dos dados. Finalmente, há uma camada de saída com 10 neurônios, ativados pela função de ativação **softmax**, que produz uma distribuição de probabilidade sobre as classes de saída.

Depois que o modelo é definido, ele é compilado usando o otimizador **adam**. O algoritmo de otimização Adam é uma variação do gradiente descendente estocástico que ajusta as taxas de aprendizado para cada parâmetro individualmente, o que geralmente resulta em uma convergência mais rápida. A função de perda escolhida é a **sparse_categorical_crossentropy**, adequada para problemas de classificação com várias classes. Esta função calcula a perda entre as classes verdadeiras e as probabilidades previstas pelo modelo. Além disso, a métrica de desempenho **accuracy** é especificada para monitorar a precisão do modelo durante o treinamento.

Ao treinar o modelo utilizando essa abordagem, ele aprende a identificar os padrões nos dados de entrada e saída correspondentes, e pode ser usado posteriormente para prever previsões precisas em novos dados.

4.6 Integração do Modelo Keras com Scikit-Learn

Neste estudo, foi necessário projetar e implementar uma classe denominada **KerasClassifierWrapper** para integrar modelos construídos usando a biblioteca Keras com a funci-

onalidade do scikit-learn. A classe **KerasClassifierWrapper** é desenvolvida para se comportar como um estimador scikit-learn, implementando os métodos **fit**, **predict**, **score**, **get_params** e **set_params** essenciais para a integração com a API do scikit-learn. Abaixo descrevo as funcionalidades da classe implementada:

- **construtor**: o construtor da classe aceita parâmetros que definem a arquitetura do modelo Keras (**build_fn**), número de épocas de treinamento, tamanho do lote e verbosidade durante o treinamento.
- **fit**: este método é chamado para treinar o modelo. Ele instancia um modelo Keras usando a função de construção (**build_fn**) fornecida, ajusta o modelo aos dados de entrada (**X**) e rótulos (**y**) usando o método **fit** do Keras.
- **predict**: este método faz previsões usando o modelo treinado. Ele usa o modelo Keras para prever os rótulos das amostras de entrada (**X**) e retorna os rótulos previstos.
- **score**: este método calcula a precisão do modelo. Ele faz previsões usando o método **predict** e compara as previsões com os rótulos verdadeiros (**y**), retornando a precisão média das previsões.
- **get_params** e **set_params**: esses métodos permitem obter e definir os parâmetros do estimador, o que é útil para ajuste de hiper parâmetros e validação cruzada.

Essa integração permite que modelos Keras sejam tratados como estimadores scikit-learn, aproveitando as funcionalidades oferecidas pelo scikit-learn, como validação cruzada, busca por hiper parâmetros e pipelines de pré-processamento, facilitando a incorporação de modelos Keras em pipelines de aprendizado de máquina mais amplos e simplifica a experimentação com diferentes modelos e técnicas de pré-processamento.

5 RESULTADOS E DISCUSSÕES

Nesta seção, apresento os resultados e discussões sobre a utilização de diferentes técnicas de extração de atributos e modelos de classificação aplicados à base de dados proposta. A base de dados consiste em imagens de 35 x 35 píxeis, extraídas a partir de números rotulados de 0 a 9. A análise foi realizada utilizando cinco distintas técnicas de extração de características: Análise de Componentes Principais (PCA), Análise Discriminante Linear (LDA), Histogramas de Gradientes Orientados (HOG), Fatoração Matricial Não Negativa (NMF) e Decomposição em Valores Singulares (SVD).

Os dados foram divididos em conjuntos de treinamento e teste para cada técnica de extração de características. Esses conjuntos de dados foram armazenados em variáveis específicas e iterados utilizando um dicionário denominado **extractors**, onde a chave corresponde ao nome

da técnica de extração de características e o valor é uma tupla contendo os respectivos dados de treinamento e teste.

A partir desses dados preparados, aplico diversos modelos de classificação para avaliar o desempenho de cada combinação de técnica de extração de características e classificador. Os modelos de classificação escolhidos para esta análise incluem Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Naive Bayes, Gaussian, Multi-layer Perceptron (MLP) e um modelo baseado em Deep Learning. A avaliação do desempenho dos classificadores foi realizada utilizando a técnica de validação cruzada K-Fold com 10 iterações, reportando a acurácia média, desvio padrão e matriz de confusão para cada combinação.

5.1 Análise de Componentes Principais (PCA)

Os resultados obtidos a partir da aplicação de diferentes algoritmos de classificação, como Support Vector Machine (SVM), K-Nearest Neighbors (KNN) e Neural Network (Deep Learning), revelam uma notável precisão na tarefa de reconhecimento de padrões, com acurácias médias variando em torno de 99,89%. O uso do PCA (Análise de Componentes Principais) em conjunto com esses classificadores demonstra ser uma abordagem robusta e eficiente. O Random Forest e o MLP apresentaram resultados excelentes, embora ligeiramente inferiores em acurácia média em comparação com o PCA, mas ainda mantendo-se dentro da faixa de alta precisão. A estabilidade desses resultados é evidenciada pelos baixos desvios padrão, sugerindo uma consistência notável na classificação dos padrões.

Por outro lado, o Decision Tree, apesar de exibir uma acurácia ligeiramente menor, ainda demonstrou um desempenho respeitável. No entanto, sua maior variação nos resultados, indicada pelo desvio padrão mais elevado, pode ser atribuída à sua natureza menos complexa, tornando-o potencialmente mais sensível a variações nos dados de entrada. Finalmente, o Naive Bayes Gaussian mostrou um bom desempenho, com uma acurácia média próxima aos outros métodos. No entanto, sua matriz de confusão revelou alguns erros de classificação. Apesar disso, sua precisão geral sugere que ainda é uma escolha viável para tarefas de reconhecimento de padrões.

Tabela 1 – Acurácia e Desvio Padrão dos Modelos de Classificação usando PCA

Classificador	Acurácia Média	Desvio Padrão
Support Vector Machine (SVM)	0,9989	0,0017
Random Forest	0,9985	0,0018
Decision Tree	0,9922	0,0063
Naive Bayes Gaussian	0,9981	0,0025
K-Nearest Neighbors (KNN)	0,9989	0,0017
Multi-layer Perceptron (MLP)	0,9985	0,0025
Neural Network (Deep Learning)	0,9989	0,0017

Figura 1 – Matriz de Confusão: SVM, Random Forest, KNN, MLP e Deep Learning

67	0	0	0	0	0	0	0	0	0
0	61	0	0	0	0	0	0	0	0
0	0	58	0	0	0	0	0	0	0
0	0	0	81	0	0	0	0	0	0
0	0	0	0	79	0	0	0	0	0
0	0	0	0	0	62	0	0	0	0
0	0	0	0	0	0	77	0	0	0
0	0	0	0	0	0	0	56	0	0
0	0	0	0	0	0	0	0	55	0
0	0	0	0	0	0	0	0	0	75

Figura 2 – Matriz de Confusão: Decision Tree

67	0	0	0	0	0	0	0	0	0
0	61	0	0	0	0	0	0	0	0
0	0	58	0	0	0	0	0	0	0
0	0	0	81	0	0	0	0	0	0
0	0	0	0	79	0	0	0	0	0
0	0	0	0	0	62	0	0	0	0
0	0	0	0	0	0	77	0	0	0
0	0	1	0	0	0	0	55	0	0
0	0	0	0	0	0	0	0	55	0
0	0	0	0	0	0	0	0	0	75

Figura 3 – Matriz de Confusão: Naive Bayes Gaussian

67	0	0	0	0	0	0	0	0	0
0	61	0	0	0	0	0	0	0	0
0	0	58	0	0	0	0	0	0	0
0	0	0	81	0	0	0	0	0	0
0	0	0	0	79	0	0	0	0	0
0	0	0	0	0	62	0	0	0	0
0	0	0	0	0	0	77	0	0	0
0	0	0	1	0	0	0	55	0	0
1	0	0	0	0	0	0	0	54	0
0	0	0	0	0	0	0	0	0	75

5.2 Análise Discriminante Linear (LDA)

Os resultados demonstram consistentemente altos níveis de acurácia para todos os algoritmos testados quando empregados em conjunto com a técnica de LDA. A acurácia média variou entre 99,67% e 99,89%, com desvios padrão relativamente baixos, variando entre 0,17% e 0,39%. Esses resultados sugerem uma excelente capacidade dos modelos treinados para discriminar entre as diferentes classes de dados.

As matrizes de confusão corroboram esses resultados, revelando que a maioria dos modelos apresentou uma classificação precisa para todas as classes, como evidenciado pelos

altos valores na diagonal principal das matrizes. Além disso, a observação de valores não nulos fora da diagonal principal foi mínima, indicando uma baixa taxa de erros de classificação.

Um aspecto interessante a ser observado é a consistência dos resultados entre os diferentes algoritmos testados. A similaridade nos padrões de classificação observados sugere que a técnica de LDA pode estar capturando efetivamente as informações discriminantes presentes nos dados, independentemente do algoritmo específico utilizado para realizar a classificação.

Esses resultados têm implicações significativas para a aplicação prática da LDA em uma variedade de domínios, incluindo reconhecimento de padrões, processamento de imagem e diagnóstico médico. A capacidade da LDA em produzir modelos de classificação altamente precisos e robustos, independentemente do algoritmo de aprendizado de máquina utilizado, destaca seu potencial como uma ferramenta poderosa para análise de dados em larga escala.

Tabela 2 – Acurácia e Desvio Padrão dos Modelos de Classificação usando LDA

Classificador	Acurácia Média	Desvio Padrão
Support Vector Machine (SVM)	0,9989	0,0017
Random Forest	0,9985	0,0018
Decision Tree	0,9967	0,0039
Naive Bayes Gaussian	0,9989	0,0017
K-Nearest Neighbors (KNN)	0,9989	0,0017
Multi-layer Perceptron (MLP)	0,9989	0,0017
Neural Network (Deep Learning)	0,9989	0,0017

Figura 4 – Matriz de Confusão

$$\begin{bmatrix} 67 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 58 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 81 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 79 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 62 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 77 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 56 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 55 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 75 \end{bmatrix}$$

5.3 Histogramas de Gradientes Orientados (HOG)

Os resultados indicam que a técnica de HOG é eficaz na extração de características discriminantes dos dados, resultando em modelos de classificação com alto desempenho. A acurácia média dos modelos variou entre 99,20% e 99,89%, com desvios padrão relativamente baixos, variando entre 0,17% e 0,48%. Esses resultados demonstram a capacidade do HOG em capturar informações relevantes dos dados, permitindo uma classificação precisa das diferentes classes. Ao analisar as matrizes de confusão, observa-se que a maioria dos modelos apresentou

uma classificação precisa para todas as classes, como evidenciado pelos altos valores na diagonal principal das matrizes. A ocorrência de valores não nulos fora da diagonal principal foi mínima, indicando uma baixa taxa de erros de classificação.

No entanto, é importante observar que alguns algoritmos mostraram uma ligeira variação no desempenho em comparação com outros. O algoritmo Decision Tree apresentou uma acurácia média ligeiramente mais baixa em comparação com os outros algoritmos testados, variando entre 99,25% e 99,35%. Isso pode ser atribuído à sensibilidade desse algoritmo a pequenas variações nos dados ou à natureza específica do problema em questão. Apesar dessas pequenas variações no desempenho entre os algoritmos, os resultados gerais corroboram a eficácia da técnica de HOG na extração de características discriminantes e sua aplicação bem-sucedida na classificação de dados.

Tabela 3 – Acurácia e Desvio Padrão dos Modelos de Classificação usando HOG

Classificador	Acurácia Média	Desvio Padrão
Support Vector Machine (SVM)	0,9989	0,0017
Random Forest	0,9981	0,0025
Decision Tree	0,9925	0,0045
Naive Bayes Gaussian	0,9941	0,0039
K-Nearest Neighbors (KNN)	0,9989	0,0017
Multi-layer Perceptron (MLP)	0,9985	0,0019
Neural Network (Deep Learning)	0,9981	0,0018

Figura 5 – Matriz de Confusão

$$\begin{bmatrix} 67 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 61 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 58 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 81 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 79 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 62 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 77 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 56 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 55 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 75 \end{bmatrix}$$

5.4 Fatoração Matricial Não Negativa (NMF)

Os resultados indicam que a técnica de NMF foi eficaz na extração de características discriminantes dos dados, resultando em modelos de classificação com alto desempenho. A acurácia média dos modelos variou entre 96,98% e 99,89%, com desvios padrões relativamente baixos, variando entre 0,35% e 0,79%. Esses resultados demonstram a capacidade do NMF em capturar informações relevantes dos dados, permitindo uma classificação precisa das diferentes classes. Ao analisar as matrizes de confusão, observa-se que a maioria dos modelos apresentou

Figura 8 – Matriz de Confusão: Naive Bayes Gaussian

67	0	0	0	0	0	0	0	0	0
0	61	0	0	0	0	0	0	0	0
0	0	58	0	0	0	0	0	0	0
0	0	0	81	0	0	0	0	0	0
0	0	0	0	79	0	0	0	0	0
0	0	0	0	0	62	0	0	0	0
0	0	0	0	0	0	77	0	0	0
0	0	1	0	0	0	0	55	0	0
0	0	0	0	0	0	0	0	55	0
0	0	0	0	0	0	0	0	0	75

Figura 9 – Matriz de Confusão: K-Nearest Neighbors (KNN)

62	0	0	0	0	0	2	0	3	0
0	61	0	0	0	0	0	0	0	0
0	0	58	0	0	0	0	0	0	0
0	0	0	81	0	0	0	0	0	0
0	0	0	0	79	0	0	0	0	0
0	0	0	0	0	62	0	0	0	0
2	0	0	0	0	0	72	0	3	0
0	0	1	0	0	0	0	55	0	0
1	0	1	0	0	0	1	1	51	0
0	0	0	0	0	0	0	0	0	75

5.5 Decomposição em Valores Singulares (SVD)

Os resultados demonstram que a aplicação de SVD como método de extração de características resultou em alta precisão nos modelos de classificação. A acurácia média dos modelos variou entre 99,21% e 99,89%, com desvios padrão variando de 0,17% a 0,43%. Estes valores indicam não apenas um alto desempenho dos classificadores, mas também uma baixa variabilidade nos resultados, o que sugere uma consistência na capacidade dos modelos em classificar corretamente os dados. Ao analisar as matrizes de confusão, observa-se que a maioria dos modelos apresentou uma classificação precisa para todas as classes, como evidenciado pelos altos valores na diagonal principal das matrizes, porém, o modelo, decision tree, apresenta pequenas imprecisões na matriz de confusão.

A aplicação do SVD para a redução de dimensionalidade em conjunto com diversos classificadores demonstrou resultados excepcionais na classificação dos dados de estudo. A ligeira inferioridade da Árvore de Decisão e as nuances observadas no Naive Bayes ressaltam a importância da seleção cuidadosa do modelo e da compreensão das suposições subjacentes de cada algoritmo. No geral, o uso do SVD mostrou-se altamente eficaz para preparar os dados para classificação precisa e eficiente.

5.6 Discussão

Dado o desempenho consistente e a alta precisão observada em todos os métodos de extração de características, os classificadores SVM, KNN e Redes Neurais destacam-se como as principais escolhas. Estes modelos demonstram não apenas um nível elevado de acurácia, mas também estabilidade através dos diferentes métodos de extração de características. Portanto, para tarefas que exigem precisão elevada e estabilidade na classificação, especialmente no contexto dos dados analisados, recomenda-se o uso de SVM, KNN e Redes Neurais.

No que diz respeito aos métodos de extração de características, PCA e LDA surgem como excelentes opções, com HOG sendo uma alternativa viável dependendo dos requisitos específicos e dos recursos computacionais disponíveis. Adicionalmente, o modelo Random Forest apresenta um desempenho robusto, situando-se ligeiramente atrás dos melhores desempenhos observados. Em contrapartida, os modelos Decision Tree e Naive Bayes, embora apresentem um desempenho relativamente inferior e maior variância, podem ser adequados para aplicações mais simples ou menos críticas.

6 CONCLUSÕES

A conclusão deste estudo evidencia a eficácia e a robustez das técnicas de extração de características e dos algoritmos de classificação utilizados no reconhecimento óptico de caracteres (OCR). A análise comparativa dos diferentes modelos de classificação, incluindo Support Vector Machine (SVM), Random Forest, Naive Bayes, K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP) e árvores de decisão, demonstrou a importância da combinação adequada de técnicas de extração de características e algoritmos de classificação para alcançar resultados precisos e confiáveis.

Além disso, a implementação de um modelo de Deep Learning baseado em redes neurais artificiais mostrou-se promissora para lidar com conjuntos de dados complexos e identificar padrões ocultos de forma eficiente. Este estudo destaca a relevância da análise sistemática das técnicas de extração de características e dos classificadores para o desenvolvimento de sistemas de visão computacional robustos e precisos, com potencial para diversas aplicações práticas. Os resultados obtidos contribuem para o avanço na área de OCR e reforçam a importância da pesquisa contínua neste campo em constante evolução.

A utilização de técnicas como Análise de Componentes Principais (PCA), Análise Discriminante Linear (LDA), Fatoração de Matriz Não-Negativa (NMF), Decomposição de Valor Singular (SVD) e Histograma de Gradientes Orientados (HOG) para a extração de características das imagens evidenciou a importância da seleção adequada de métodos de processamento de imagens para aprimorar a precisão e a generalização dos modelos de classificação. A combinação dessas técnicas, juntamente com a aplicação de algoritmos de aprendizado de máquina e redes neurais profundas, permitiu a construção de sistemas capazes de identificar padrões visu-

ais complexos e realizar tarefas de reconhecimento com alto desempenho. Este estudo destaca a relevância da integração de diferentes abordagens para aprimorar a eficácia dos sistemas de visão computacional em cenários práticos e desafiadores.

7 REFERÊNCIAS

BISHOP, C. **Pattern Recognition and Machine Learning**. [S.l.]: Springer New York, 2016. (Information Science and Statistics). ISBN 9781493938438.

CRESWELL, J. W. **Research design: Qualitative, quantitative, and mixed methods approaches**. 4th. ed. [S.l.]: Sage Publications, 2014.

FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. [S.l.]: Prentice Hall, 2003.

GEETHA, T.; SENDHILKUMAR, S. **Machine Learning: Concepts, Techniques and Applications**. Chapman & Hall/CRC Press, 2023. ISBN 9781003290100. Disponível em: <<https://books.google.com.br/books?id=hnCezwEACAAJ>>.

GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L. **Digital image processing using MATLAB**. [S.l.]: Gatesmark Publishing Knoxville, 2008. ISBN 978-0982085400.

LI, Y. *et al.* Revisiting dynamic convolution via matrix decomposition. **CoRR**, abs/2103.08756, 2021. Disponível em: <<https://arxiv.org/abs/2103.08756>>.

S, A. **An overview of Tesseract OCR Engine**. Department of Computer Science and Engineering, 2016. A Seminar Report.

SZELISKI, R. **Computer Vision: Algorithms and Applications**. [S.l.]: Springer Science & Business Media, 2011.

YIN, R. K. **Case study research and applications: Design and methods**. 6th. ed. [S.l.]: Sage Publications, 2018.