UNIVERSIDADE ESTADUAL DE MARINGÁ - CENTRO DE TECNOLOGIA - CTC
CIÊNCIA DA COMPUTAÇÃO

ARTHUR BELINI PINI

APLICAÇÃO E COMPARAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÕES DE IMAGENS OFTALMOLÓGICAS

MARINGÁ - PR
2024

ARTHUR BELINI PINI

APLICAÇÃO E COMPARAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÕES DE IMAGENS OFTALMOLÓGICAS

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação do Universidade Estadual de Maringá - Centro de Tecnologia - CTC para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Igor da Penha Natal

Coorientador: Prof. Wagner Igarashi

MARINGÁ - PR
2024

ARTHUR BELINI PINI

APLICAÇÃO E COMPARAÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÕES DE IMAGENS OFTALMOLÓGICAS

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação do Universidade Estadual de Maringá - Centro de Tecnologia - CTC para a obtenção do título de Bacharel em Ciência da Computação.

Aprovado em: xx/ 04/ 2024 pela banca examinadora:

Prof. Dr. Igor da Penha Natal - UEM (Orientador)

Prof. Wagner Igarashi - UEM (Coorientador)

Profª. Dra. Josiane Melchiori Pinheiro - UEM

Prof. Maurilio Martins Campano Junior - UEM

RESUMO

O trabalho utiliza conceitos de Inteligência Artificial (IA) e técnicas de Aprendizado de Máquina (AM) para fazer classificações de imagens oftalmológicas de fundoscopia em relação à natureza do olho (saudável ou com alguma doença, como catarata, glaucoma, diabetes, etc.) através de reconhecimento de padrões. A partir da utilização das técnicas de aprendizagem profunda (traduzido do inglês *deep learning*) e algoritmos clássicos de AM, este trabalho visa analisar o estado da arte atual de métodos de classificação.

Palavras-chave: Aprendizado de Máquina. Aprendizagem Profunda. Reconhecimento de Padrões.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

A principal ideia por trás do desenvolvimento deste trabalho está em uma desconfiança em relação ao estado da arte do AM, que pode ser sumarizado em responder a seguinte pergunta: "A maior utilização de aprendizagem profunda sobre outras tecnologias de AM está relacionada somente ao seu desempenho superior, ou existe uma superestimação de sua superioridade por se tratar de uma tecnologia relativamente recente em que há grande entusiasmo em sua utilização?".

Com a impossibilidade de se fazer um estudo prático deste tema sobre todas as aplicações possíveis de AM, foi selecionado um tema onde a aprendizagem profunda é predominantemente utilizada. Sendo este o estado atual, espera-se que o resultado seja favorável à utilização de aprendizagem profunda, o que justifica de fato sua ampla utilização.

1.2 Visão geral

A tecnologia de AM vem tido um grande crescimento nas últimas décadas, com aplicações de sucesso em áreas como automação de automóveis, como os produzidos pela Tesla[1], e *chatbots onlines*, como o ChatGPT[2]. Dentre as aplicações de AM, está o reconhecimento de padrões (BISHOP; NASRABADI, 2006), que pode ser aplicado em diversas áreas. Uma das áreas significativas para esse tipo de aplicação é o processamento de imagens com o objetivo de classificar objetos, que é o foco deste trabalho.

Além disso, um tipo específico de AM, aprendizagem profunda, vem tido destaque em sua utilização pela sua popularidade e realizações (AHMED *et al.*, 2023). Por esse motivo, o trabalho partirá de algoritmos dessa natureza e explorará ainda outras técnicas tradicionais de AM com o objetivo de compará-los na solução do problema de classificação de doenças em imagens oculares. Com a tecnologia de aprendizagem profunda sendo amplamente utilizada e popularizada nos últimos anos, torna-se relevante fazer uma investigação da eficácia de aplicações desse tipo em comparação com outras que utilizam algoritmos clássicos em seu processamento.

Este tipo de tecnologia em específico (aprendizagem profunda) é relevante em vários campos, como para a locomoção de veículos autônomos, reconhecimento de objetos em imagens[3], análise do solo em agricultura (LIAKOS *et al.*, 2018), arte gerada por IA[4], controle de qualidade de produtos (KADAM *et al.*, 2021), entre outros. Nosso foco será sua aplicação em análise médica, onde essa tecnologia também é relevante (TONG *et al.*, 2020).

O trabalho desenvolvido, portanto, se baseia em analisar imagens de olhos humanos para determinar se são saudáveis ou possuem alguma doença, como catarata, glaucoma, miopia,

[1] <https://www.tesla.com/AI>, acessado em 18 de fevereiro de 2024
[2] <https://openai.com/research/gpt-4>, acessado em 18 de fevereiro de 2024
[3] <https://learn.ml5js.org/#/reference/image-classifier>, acessado em 18 de fevereiro de 2024
[4] <https://openai.com/research/dall-e>, acessado em 18 de fevereiro de 2024

etc., dependendo de quais dessas são disponibilizadas pelo banco de imagens (*dataset*) utilizado. Para isso, serão utilizadas algumas tecnologias clássicas de AM. Os resultados da utilização desses algoritmos então serão comparados com a utilização de aprendizagem profunda, que é um conjunto de algoritmos que obteve um recente crescimento e domina os demais em diversos campos.

Com isso, o objetivo final após a comparação dos resultados dos algoritmos clássicos com o que utiliza aprendizagem profunda será avaliar se este possui uma vantagem significativa sobre os demais no problema com o banco de imagens utilizado, e portanto buscar justificar ou não a grande utilização e substituição de algoritmos clássicos. Dependendo dos resultados obtidos, novas possibilidades de utilização de algoritmos que não são tão utilizados nos tempos atuais poderiam surgir.

Para cumprir o objetivo proposto, serão utilizados como metodologia: pesquisas para revisões bibliográficas, bancos de dados e ferramentas disponíveis para dar suporte ao desenvolvimento, criação de códigos que apliquem os algoritmos, treinamento das IAs desenvolvidas, aplicação destas sobre um banco de imagens e comparação dos resultados.

A organização dos tópicos deste trabalho é a seguinte:

Materiais e Métodos: onde serão apresentadas as ferramentes utilizadas no desenvolvimento do trabalho, como softwares e hardwares;

Fundamentação Teórica: onde será feita uma revisão bibliográfica apresentando o estado da arte do AM aplicado a imagens médicas oculares, além do embasamento teórica necessário para se desenvolver o restante do trabalho, como detalhes das doenças oftalmológicas classificadas, métricas de performance de AM e algoritmos.;

Desenvolvimento: onde será descrito de fato o que foi desenvolvido a partir da fundamentação teórica, materiais, métodos e motivações. Isso inclui a sumarização de *datasets* encontrados e planejamento da *baseline*;

Cronograma: onde será definida a sequência de atividades que serão desenvolvidas, levando-se em conta a dependência entre as mesmas para se atingir o objetivo e o prazo de execução destas;

Conclusão: onde são colocados os resultados (como o que foi alcançado dos objetivos propostos além de outras informações significantes que eventualmente foram encontradas) do que foi desenvolvido, incluindo desafios enfrentados e limitações;

Referências Bibliográficas: onde serão compiladas as citações feitas durante o trabalho.

1.3 Objetivo geral

Como apresentado anteriormente, o objetivo geral do trabalho é fazer uma comparação entre algoritmos clássicos, que já foram muito utilizados no passado, e um algoritmo moderno, que está sendo amplamente utilizado, para o problema proposto. Além disso, o estudo das diferenças essenciais entre os métodos utilizados e técnicas para aumento de performance geral dos algoritmos.

1.4 Objetivos específicos

1. **Desenvolvimento da fundamentação teórica (seção 3)**: pesquisar pelos conceitos utilizados durante o desenvolvimento dos modelos, para dar suporte ao desenvolvimento eficaz destes e na comparação dos resultados, com o levantamento de pontos relevantes;

2. **Definição das técnicas de AM (seção 3.7**): pesquisar por métodos clássicos de AM que se enquadram ao problema, tentando buscar um conjunto de técnicas (com diferenças significativas entre si, para representarem uma ampla gama de metodologias) que tenham o melhor resultado possível e alcancem potencialmente o desempenho da aprendizagem profunda;

3. **Definição do conjunto de imagens (seção 4.1)**: buscar por imagens de olhos saudáveis e doentes de bancos de dados disponibilizados gratuitamente e fazer um compilado que seja adequado para o processamento, com uma quantia total de imagens adequada equilibrando a qualidade dos resultados e o poder de processamento necessário para se atingir o resultado no prazo determinado;

4. **Desenvolvimento da *baseline* (seção 4.2)**: sendo definida, brevemente, como um protótipo da aplicação final, o desenvolvimento de uma *baseline* serve como ponto de partida para próximas etapas, fazendo surgir temas importantes;

5. **Pré-processamento das imagens:** definir o pré-processamento que será utilizado nas imagens, adequando o tamanho e cores para que fiquem em um padrão bem definido para melhorar ao máximo os modelos;

6. **Configuração dos modelos**: definir os modelos que serão utilizados, incluindo configurações dos algoritmos clássicos de AM e camadas da rede neural da aprendizagem profunda;

7. **Desenvolvimento das técnicas de AM:** desenvolver códigos das técnicas definidas para aplicar de fato sobre o *dataset*;

8. **Execução dos modelos e coleta dos resultados:** fazer a coleta dos resultados de métricas de performance, como acurácia, precisão, *recall* e F1-score;

9. **Comparação dos resultados (seção 4.3):** apontar as vantagens e desvantagens de cada método aplicado. Destacar pontos onde o algoritmo de aprendizagem profunda se sobressai aos demais, ou o contrário, e ainda chegar a uma conclusão geral de comparação entre os dois. Discutir ainda a aplicabilidade dos algoritmos utilizados para a área médica e oftalmológica;

10. **Melhorias e pesquisas futuras (seção 6.3):** fazer uma análise de possíveis ajustes que podem ser feitos para melhorar o desempenho dos algoritmos, identificando ainda áreas de desenvolvimentos futuros onde os elementos testados podem ter alguma relevância.

2 MATERIAIS E MÉTODOS

Os recursos computacionais e metodológicos utilizados para o desenvolvimento do trabalho são: em relação ao hardware, um processador "*AMD Ryzen 53600 6-Core Processor (12 CPUs),* ∼3.6 GHz", uma placa de vídeo "*Radeon Rx 590 Series,* 8 GB" e 16 GB de memória. Em relação ao software, são utilizados a linguagem de programação Python (incluindo bibliotecas desta para aprendizado máquina e manipulação de imagens) e sistema operacional Windows 11. As bibliotecas Python utilizadas podem ser vistas abaixo:

scikit-learn[1]**:** que possui funcionalidades de distribuição dos dados, classificação e avaliação de performance;

OpenCV[2]**:** que possui funcionalidades de pré-processamento das imagens;

NumPy[3]**:** que possui funcionalidades de aprimoramento das estruturas de dados, como matrizes das imagens;

pandas[4]**:** que possui funcionalidades estatísticas, como para ler dados de arquivos CSV;

TensorFlow[5]**:** que implementa algoritmos de aprendizagem profunda;

Keras[6]**:** interface para o TensorFlow;

imbalanced-learn[7]**:** implementa funcionalidades de "reamostragem" para desbalanceamento do *dataset*;

Matplotlib[8]**:** implementa geração de gráficos a partir dos resultados de performance, para facilitar a visualização.

[1] <https://scikit-learn.org/>, acessado em 18 de fevereiro de 2024.
[2] <https://opencv.org/>, acessado em 18 de fevereiro de 2024.
[3] <https://numpy.org/>, acessado em 18 de fevereiro de 2024.
[4] <https://pandas.pydata.org/>, acessado em 18 de fevereiro de 2024.
[5] <https://www.tensorflow.org/>, acessado em 18 de fevereiro de 2024.
[6] <https://keras.io/>, acessado em 18 de fevereiro de 2024.
[7] <https://imbalanced-learn.org/>, acessado em 18 de fevereiro de 2024.
[8] <https://matplotlib.org/>, acessado em 18 de fevereiro de 2024.

3 FUNDAMENTAÇÃO TEÓRICA

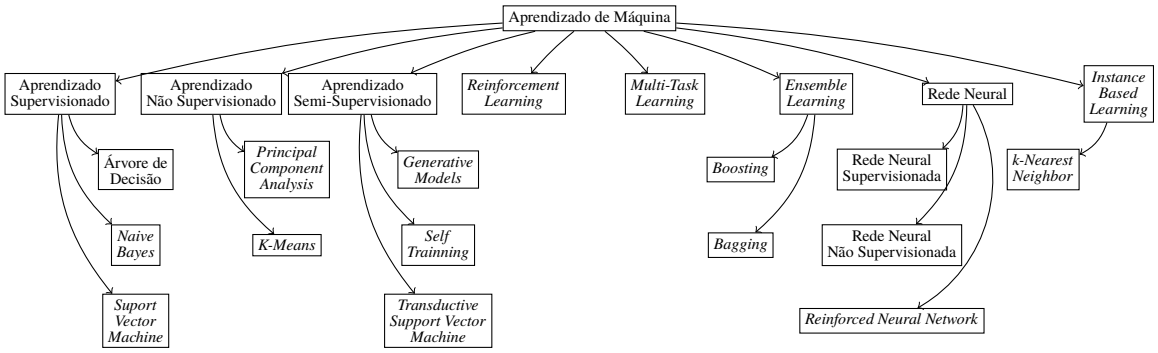
3.1 Aprendizado de máquina

Segundo Mahesh (2020), no qual esta revisão é baseada, Arthur Samuel, um dos pioneiros nos estudos da área, define o AM como a capacidade de uma máquina realizar alguma ação para qual não foi explicitamente programada. Ou seja, parte de seu "conhecimento" advém do programador e a outra parte, que a adapta a realizar suas tarefas, é ajustada posteriormente, simulando aprendizado. Isso é vantajoso em muitos casos, pois, através de um algoritmo geral, a IA pode se adaptar a problemas específicos através de absorção de informações de um *dataset*. O desenvolvedor não precisa se envolver nem compreender o que foi "aprendido" pela IA nessa segunda fase, somente o algoritmo geral.

Portanto, outra utilização vantajosa do AM é sua capacidade de extrair informações de grandes conjuntos de dados complexos demais para a interpretação humana. Com o surgimento de conjuntos de dados cada vez maiores na atualidade, torna-se cada vez maior a demanda por esta tecnologia.

Existem diversas variações de algoritmos de AM, cada um sendo adequado a determinados tipos de problemas. Essas variações podem ser observadas na representação de uma árvore (figura 1) que tem como raiz o AM de forma geral e diversos ramos que se especializam em algoritmos específicos, como redes neurais, k-Nearest Neighbor (k-NN), Naive Bayes, etc..

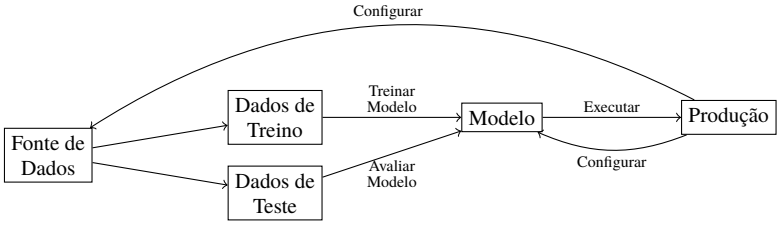
Figura 1 – Árvore de algoritmos de AM.



Fonte: baseado em Mahesh (2020).

Os algoritmos de AM se dividem principalmente em supervisionados e não supervisionados. No primeiro caso, supervisionado, são disponibilizados à máquina vários dados rotulados de acordo com o grupo do qual fazem parte. A máquina então extrai informações específicas de cada grupo para ficar pronta para classificar novos dados como pertencentes a algum destes. A máquina então é testada e um modelo é criado para ser utilizado de fato com dados que se deseja classificar (esse processo pode ser visto na figura 2).

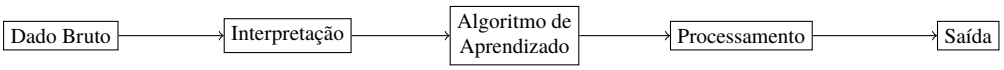
Figura 2 – Processo de aprendizado supervisionado.



Fonte: baseado em Mahesh (2020).

No caso não supervisionado, as classificações dos dados em grupos são feitas de acordo com as características já aprendidas de dados anteriores pelo próprio algoritmo, sem intervenção de uma rotulação manual (esse processo pode ser visto na figura 3).

Figura 3 – Processo de aprendizado não supervisionado.



Fonte: baseado em Mahesh (2020).

O autor conclui dizendo que os algoritmos supervisionados terão, em geral, uma melhor performance na utilização de bancos de dados menores, devido a maior facilidade de rotular os dados. Em casos onde o conjunto de dados é maior é aconselhado utilizar algoritmos não supervisionados, pois, apesar da falta de rotulação dos dados, há uma grande quantidade para o aprendizado do modelo. Ele ainda aponta que para muitas quantidades de dados é vantajoso utilizar técnicas de aprendizagem profunda.

3.2 Aprendizagem profunda

Esta introdução à aprendizagem profunda se baseia no artigo (LECUN; BENGIO; HINTON, 2015), que faz uma revisão sobre o assunto e, apesar de relativamente antigo, se mantém relevante, sendo amplamente citado, por tratar em grande parte de conceitos matemáticos que pouco variam ao longo do tempo.

Os autores comparam a aprendizagem profunda com algoritmos clássicos de AM dizendo que estes últimos possuem grande dificuldade de processamento de dados brutos, como *pixels* de uma imagem, para transformação em informações úteis.

Os algoritmos de aprendizagem profunda podem ser classificados como *representational learning*, que é um conjunto de métodos capazes de transformar dados brutos em uma representação pronta para ser analisada e extrair informações.

A aprendizagem profunda possui múltiplos níveis de representação dos dados, com níveis de abstração cada vez mais altos. Com o primeiro nível de representação podendo ser, no exemplo onde os dados brutos são *pixels* de uma imagem, *pixels* que formam bordas nessa imagem e um nível mais acima podendo ser objetos nessa imagem e assim por diante, com informações cada vez mais concretas.

Essa tecnologia conquistou vários feitos importantes que algoritmos anteriores de IA não conseguiam nos campos de reconhecimento de imagens, reconhecimento de fala, aplicações na ciência e principalmente processamento em linguagem natural.

3.3 Identificação de doenças em imagens oculares com AM

Este estudo sobre o estado atual do uso de AM para identificação de doenças em imagens oculares é baseado no artigo (TONG *et al.*, 2020), que faz exatamente isso. Os autores afirmam que o uso de IA e AM são relevantes para análise de imagens oculares e médicas no geral.

Os três métodos principais de coleta de imagens oculares são: fundoscopia, tomografia de coerência óptica (OCT do termo em inglês *ocular coherence tomography*) e imagem de lâmpada de fenda. A fundoscopia utiliza câmeras ópticas para obter imagens aumentadas de tecidos da retina. Foi utilizada em vários estudos para diagnosticar e monitorar doenças. O OCT é um tipo não invasivo de diagnóstico por imagem que produz uma grande quantidade informações sobre a morfologia da retina. A imagem de lâmpada de fenda é um método onde é incidido sobre o olho uma grande quantidade de luz para se examinar as partes anteriores e posteriores deste.

Os autores ainda afirmam que existem diversos métodos de AM que podem ser utilizados para análise de imagens médicas, como árvore de decisões, algoritmo de Naive Bayes, floresta aleatória, *Support Vector Machine* (SVM) e k-NN. Estes, apesar de obterem um bom resultado quando utilizados em um conjunto de dados reduzido, apresentam problemas para conjuntos maiores. É nesse cenário que a aprendizagem profunda ganha destaque, conseguindo obter mais informações dos dados brutos através de camadas escondidas sem intervenção humana. Por esse motivo, a aprendizagem profunda é o algoritmo mais utilizado nesses casos de análise de imagens.

Apesar de resultados muito promissores, ainda existem grandes desafios para a área, como a qualidade das imagens, que podem variar de acordo com como são coletadas, o que pode afetar o desempenho da IA. Além disso, há variações das características de olhos dependendo do paciente, o que também pode prejudicar o reconhecimento de padrões da IA. Ainda existem diversos outros problemas, como enviesamento do treinamento dos modelos, falta de imagens o suficiente para treinar os modelos adequadamente, principalmente para doenças raras, privacidade de dados médicos e dificuldade de se criar um único modelo capaz de identificar diversas doenças.

3.4 Rótulos utilizados

Como o trabalho se desenvolve em torno do tema oftalmologia, torna-se interessante fazer uma pequena introdução aos estados de saúde dos olhos que comporão os *datasets* utlizados e que servirão como rótulos para a classificação dos modelos AM. Portanto, segue uma lista das doenças com uma breve descrição — baseadas em Kanski e Bowling (2011) — e seus respectivos rótulos definidos por letras maiúsculas:

Normal (N): este rótulo caracteriza a ausência das doenças que serão classificadas pelos modelos;

Retinopatia diabética (D): metade das pessoas com diabetes desenvolvem retinopatia com o tempo, sendo o edema macular diabético umas das causas mais comuns da perca de visão;

Glaucoma (G): termo utilizado para descrever um grupo de condições. Todas têm em comum a perda de visão conforme a doença progride;

Catarata (C): opacidade da lente natural do olho;

Degeneração macular relacionada à idade (A): doença degenerativa que atinge a macula do olho que, em estágios posteriores, pode comprometer a visão;

Hipertensão (H): doença caracterizada por pressão intraocular (PIO) acima do normal;

Miopia (M): doença causada por fatores genéticos e ambientais que se caracteriza pelo alongamento ocular e que pode causar o comprometimento da visão;

Outros (O): outras anormalidades não identificadas.

3.5 Métricas de performance

Como fundamentação às análises de desempenho dos métodos utilizados, é importante definir diferentes métricas de performance. Essa revisão é baseada em Müller e Guido (2016). São elas: acurácia, precisão, *recall* e *F1-score*. Para isso, é necessário definir alguns termos utilizados nas métricas, que podem ser vistos a seguir:

TP**:** se refere a *true positive* ou “verdadeiro positivo”, ou seja, classificações feitas pelo modelo como positivas que realmente são positivas;

TN**:** se refere a *true negative* ou “verdadeiro negativo”, ou seja, classificações feitas pelo modelo como negativas que realmente são negativas;

FP**:** se refere a *false positive* ou “falso positivo”, ou seja, classificações feitas pelo modelo como positivas que na realidade são negativas;

***FN*:** se refere a *false negative* ou "falso negativo", ou seja, classificações feitas pelo modelo como negativas que na realidade são positivas.

Com esses termos, podemos definir a **matriz de confusão**, que sumariza essas informações a respeito de um modelo de AM, como a matriz abaixo:

$$\begin{array}{c} \\ \text{classe negativa} \\ \text{classe positiva} \end{array} \begin{array}{c} \begin{array}{cc} \text{predição negativa} & \text{predição positiva} \end{array} \\ \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix} \end{array}$$

Dessa forma, existem algumas métricas derivadas dessa matriz que são capazes de produzir informações mais específicas a respeito dos dados vistos TP, TN, FP e FN. Essas métricas são definidas, em relação aos seus significados e equações para calculá-las, abaixo:

Acurácia: proporção entre classificações corretas do AM e quantidade total de classificações — é capaz de informar o grau de acerto do modelo;

Precisão: proporção entre classificações positivas corretas e classificações totais positivas (falsas e verdadeiras) — é capaz de informar o grau de acerto do modelo somente em relação às classificações positivas feitas;

***Recall* ou revocação:** proporção entre classificações positivas corretas e quantidade total de amostras positivas — é capaz de informar o grau de acerto do modelo em relação às classificações feitas somente a respeito de amostras positivas;

***F1-score*:** para dar uma visão mais geral podemos unir precisão e *recall* com uma média harmônica entre os dois — é capaz de informar o grau de acerto do modelo mesclando as informações fornecidas por ambos, precisão e *recall*.

As equações que calculam essas métricas são as seguintes:

$$\text{Acurácia} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precisão} = \frac{TP}{TP+FP}$$

$$\textit{Recall} = \frac{TP}{TP+FN}$$

$$\text{f1-score} = 2 \times \frac{\text{Precisão} \times \textit{Recall}}{\text{Precisão} + \textit{Recall}}$$

É importante ressaltar que dificilmente uma métrica apenas possa dar o contexto geral dos resultados. Geralmente, uma métrica explica algo pontual sobre a performance. Podemos

ter, por exemplo, uma alta precisão ou revocação e ainda termos um modelo ruim, pois essas duas métricas são complementares em certo sentido, mostrando erros que a outra não calcula. Portanto, o ideal é ter um equilíbrio entre essas métricas, de forma que sejam próximas.

Um modelo pode ter uma precisão alta por raramente prever positivo, mas ter acertado essas predições, sabendo que há um desbalanceamento no *dataset* e existem menos positivos que negativos; da mesma forma, um modelo pode ter uma revocação alta por quase sempre prever positivo, pelo conjunto de dados ter uma quantidade maior de positivos. Dessa forma, a precisão e acurácia nunca poderão ser altas ao mesmo tempo para positivo e negativo se algo assim acontece. Portanto, a métrica que mais se aproxima a dar a performance considerando o contexto geral é o f1-score, que mescla ambos as métricas, sendo sensível a diferenças de valores entre elas.

Modelos que vão além de uma classificação binária (valores "verdadeiro" e "falso"), como o descrito pelas definições das métricas, estendem essas definições para calcular a performance das classificações com diversos *labels*, como é feito neste trabalho, com classificações de diversas doenças. Os cálculos dessas métricas podem ser realizados com a biblioteca do Python ***scikit-learn***, que será utilizada para o desenvolvimento dos modelos.

3.6 Pré-processamento

A etapa de pré-processamento em AM se refere à fase após a extração das imagens que serão utilizadas para treinar o modelo. Há, então, duas preocupações a respeito das imagens extraídas: seu tamanho, em *bits*, o que vai afetar na carga de processamento do classificador, e as características que estas apresentam e o quanto são acentuadas para facilitar a extração destas pelo classificador. Mais informações sobre pré-processamento de imagens podem ser encontradas em Gonzalez e Woods (2000).

Algumas dessas características são perdidas na própria extração das imagens. Portanto, deve-se levar em consideração também o método da coleta dessas, para que mantenham a maior quantidade de informações possível do objeto real analisado. Porém, após as imagens já estarem extraídas, torna-se responsabilidade do pré-processamento acentuar as características das imagens coletadas, enquanto que, para isso, utiliza a menor quantidade de informações possível em relação ao tamanho da imagem para não sobrecarregar o processamento.

Em relação a diminuir o tamanho da imagem, em geral, são utilizadas algumas técnicas: redimensionamento e dessaturação da imagem. A primeira, redimensionar a imagem para uma resolução inferior, é feita até certo ponto onde pouca informação da imagem original relevante é perdida enquanto que diminui o tamanho da imagem, de forma que haja um equilíbrio entre as duas coisas. No caso da dessaturação da imagem, tornando-a "preto e branco", é feita quando a informação da matiz dos *pixels* da imagem não são (ou são pouco) relevantes para as características que estão sendo extraídas, dessa forma é feita a diminuição do tamanho da imagem através da remoção dos canais de cores desta.

Ainda existem dois métodos de pré-processamento mais avançados, que, diferentes dos outros métodos que apenas alteram as imagens para deixarem mais adequadas para o algoritmo, criam novas imagens sinteticamente com base nas originais. Ambos os métodos, "aumento de dados"[1] e "reamostragem"[2], têm como objetivo aumentar a variedade de imagens no *dataset*, porém servem propósitos específicos diferentes. O primeiro, "aumento de dados", tem como objetivo aumentar a quantidade de dados quando esta é pequena, fazendo transformações aleatórias nas imagens originais. O segundo, "reamostragem", tem como objetivo aumentar (classes minoritárias) ou diminuir (classes majoritárias) a quantidade de imagens de classes buscando balanceá-las.

Apesar da importância dos objetivos desses métodos criação sintética de imagens e do grande potencial destes para o aumento da performance do modelo, existem alguns riscos quando utilizados exageradamente. Quando muitos dados são criados dessa forma artificial, corre-se o risco de aumentar a probabilidade de ocorrer *overfitting*[3], ou seja, adaptação extrema do modelo para os dados sintéticos, que podem não representar fielmente os dados reais e levar a uma acurácia de validação dos dados de teste baixa. Portanto, é preciso tomar o cuidado de manter dados reais em uma proporção significativa dos dados utilizados. Vale ainda notar que gerar dados fictícios somente faz sentido para o *dataset* de treino, uma vez que o modelo precisa funcionar para dados reais e ser avaliado a partir disso.

3.7 Métodos clássicos de classificação

Os métodos clássicos de classificação foram considerados de forma que cubram uma ampla variedade de tipos de algoritmos, para que as diferenças (vantagens e desvantagens) entre eles possam ficar bem evidentes. Portanto, as famílias de algoritmos e seu representante escolhidos podem ser vistos a seguir:

- ***lazy* (baseado em instância):** essa família de métodos, diferentemente das demais, não ajusta as configurações do modelo de forma genérica com base nos dados de forma prévia, mas possui um processamento "atrasado" e único para cada instância classificada, o que, em alguns casos, pode ser vantajoso (MITCHELL, 1997). O algoritmo escolhido para essa família foi o k-NN;

- **Baseado em função:** essa família de métodos utiliza funções lineares ou não lineares para estimar curvas que separem classes de instâncias da forma mais otimizada possível (BOSER; GUYON; VAPNIK, 1992). O algoritmo escolhido para essa família foi o *Support vector machine* (SVM);

[1] <https://www.tensorflow.org/tutorials/images/data_augmentation>, acessado em 18 de fevereiro de 2024.
[2] <https://imbalanced-learn.org/stable/references/index.html>, acessado em 18 de fevereiro de 2024.
[3] <https://www.tensorflow.org/tutorials/keras/overfit_and_underfit>, acessado em 18 de fevereiro de 2024.

- **Árvores de Decisão:** Algoritmos baseados em árvores de decisão, onde decisões são tomadas seguindo diferentes ramificações de uma árvore onde os nós são valores para variáveis que descrevem o problema (QUINLAN, 1986). O algoritmo escolhido para essa família foi o floresta aleatória;

- **Bayesiano:** Algoritmos que usam como fundamento a teoria de "probabilidade condicional" e Teorema de Bayes introduzidos por Bayes (1763), buscando maximizar as probabilidades de dependência entre as classes e características das instâncias classificadas (RISH *et al.*, 2001). O algoritmo escolhido para essa família foi o Naive Bayes.

Algoritmo k-NN

Segundo Peterson (2009), o *K-nearest-neighbor* (kNN, KNN ou k-NN), que pode ser traduzido como "k-ésimo vizinho mais próximo", é um dos algoritmos mais simples de classificação de dados. É um bom um candidato quando não se conhece a distribuição dos dados. Esse classificador é descrito por Cover e Hart (1967).

De forma geral, o k-NN é um método de classificação de dados em classes que leva em consideração os rótulos de maior frequência de dados próximos (vizinhos) do qual se deseja classificar. Formalmente, a função matemática que representa o classificador pode ser descrita[4] como:

$$h(x) = \text{moda}(y'' : (x'', y'') \in S_x),$$

onde "moda$(\cdot)$", nesse contexto, retorna os valores mais frequentes de y''. Valores de y representam classes ou rótulos de valores de x, que são os dados classificados. S_x representa os k vizinhos mais próximos de x, o que implica que $|S_x| = k$.

Os vizinhos de x em S_x são considerados os mais próximos segundo a seguinte propriedade:

$$\text{dist}(x, x') \geq \underset{(x'', y'') \in S_x}{\arg\max} \left(\text{dist}(x, x'')\right),$$

onde (x', y') e S_x são disjuntos, "max$(\cdot)$", nesse contexto, retorna o valor máximo do conjunto dist(x, x'') e "dist$(\cdot)$" é a métrica de distância utilizada que define a proximidade dos vizinhos. Ou seja, não há dados fora de S_x que estão mais próximos de x do que algum em S_x.

Existe mais de uma métrica de distância possível de ser utilizada, sendo melhor aquela que prever uma maior similaridade de rótulos entre dois dados para valores mais próximos. A métrica mais comumente utilizada é a **distância de Minkowski**, uma generalização da distância euclidiana (quando o parâmetro $p = 2$), que pode ser dada como:

$$\text{dist}(x, z) = \left(\sum_{r=1}^{d} |x_r - z_r|^p \right)^{1/p}.$$

[4] <https://www.cs.cornell.edu/courses/cs4780/2017sp/lectures/lecturenote02_kNN>, visitado em 27 de novembro de 2023

Algoritmo SVM

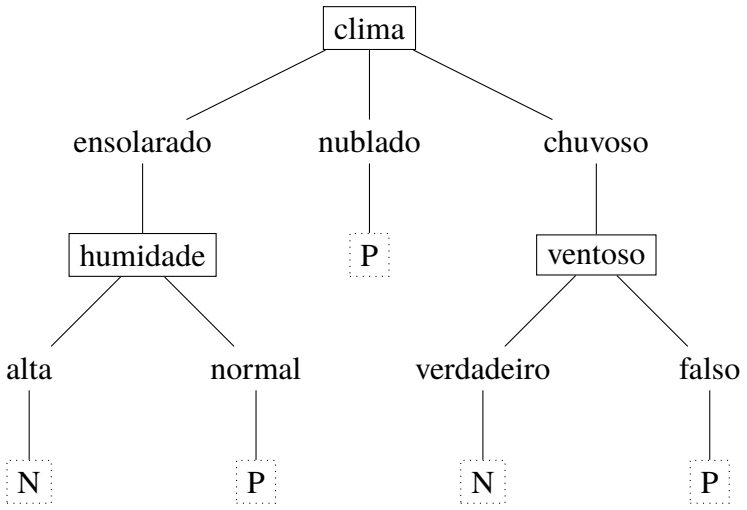
O método de classificação *Support vector machine* (*SVM*), que pode ser traduzido como "Máquina de vetor de suporte", gera uma função linear ou não linear que tenta dividir, através das regiões acima e abaixo da linha ou curva, as classes dos dados, que estão distribuídos em pontos do espaço[5]. O algoritmo foi introduzido por Boser, Guyon e Vapnik (1992).

Algoritmo floresta aleatória

O método de classificação "floresta aleatória", descrito por Breiman (2001), é uma fusão (de *ensemble*) de classificadores "árvore de decisão", descrito por sua vez por Quinlan (1986). Segundo Breiman, aumentos significativos na acurácia de classificação foram alcançados através da fusão de árvores de decisão. Portanto, este foi um dos métodos de classificação escolhidos justamente por ser um bom representante da "família" de algoritmos baseados em árvores de decisão.

Na figura 4, é possível observar uma árvore de decisão simples para uma classificação binária de um problema onde se deseja decidir se o clima está ou não adequado para uma atividade ao ar livre qualquer. O problema tem como a decisão não adequado o valor "N" (negativo) e adequado o valor "P" (positivo). A partir dos valores dos atributos "clima", "humidade" e "ventoso", a árvore se ramifica de diferentes formas até chegar em um nó folha que contém uma decisão N ou P (QUINLAN, 1986). A partir desse contexto, o algoritmo de floresta aleatória utiliza a votação de vários resultados de árvores de decisão, que frequentemente possuem seus crescimentos baseados em vetores gerados aleatoriamente (BREIMAN, 2001).

Figura 4 – Exemplo simples de uma árvore de decisão para decidir se o clima está adequado para alguma atividade qualquer.



Fonte: traduzido de Quinlan (1986).

[5] <https://www.cs.cornell.edu/courses/cs4780/2017sp/lectures/lecturenote09>, visitado em 27 de novembro de 2023

Naive Bayes

A família de algoritmos de classificação Naive Bayes se baseia na teoria estatística de "probabilidade condicional", introduzida por Bayes (1763), que define a probabilidade de um evento A dado a ocorrência de B como:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Este método possui a suposição "ingênua" (*naive*) de que as características de uma instância a ser classificada pelo modelo são independentes, para facilitar o cálculo. Apesar da suposição ingênua, que na prática normalmente não ocorre, pelas características geralmente possuírem certas dependências, esse método possui potencial de obter bons resultados. De maneira formal, essa independência é descrita como:

$$P(X \mid C) = \prod_{i=1}^{n} P(X_i \mid C) \tag{3.1}$$

onde $X = (X_1, \cdots, X_n)$ é um vetor de características e C é uma classe (RISH *et al.*, 2001).

Com isso, uma instância pode ser classificada como fazendo parte da classe que maximiza a probabilidade de $P(C = i \mid X = x) = \frac{P(X=x|C=i)P(C=i)}{P(X=x)}$, sendo o denominador $P(X = x)$ igual para toda classe e portanto ignorado. Dessa forma, a classe (i) da instância (x) é calculada pelo Naive Bayes como (RISH *et al.*, 2001):

$$\arg\max_{i}(P(X = x \mid C = i)\, P(C = i))$$

esta, por sua vez, pode ser reduzida, devido à equação 3.1, para:

$$\arg\max_{i}(\prod_{j=i}^{n} P(X_j = x_j \mid C = i)\, P(C = i)).$$

4 DESENVOLVIMENTO

4.1 Pesquisa por *datasets*

Foram pesquisados *datasets* que possuem as seguintes características: coleção pública de imagens oculares que utilizam o método de fundoscopia. Foram utilizadas como fontes de pesquisa a plataforma *online* Kaggle[1] e artigos científicos como Jeong, Hong e Han (2022), que lista vários *datasets* que se enquadram na pesquisa feita.

Portanto, segue a tabela (tabela 1) de *datasets* que cumpriram os requisitos estabelecidos e que contém as respectivas características destes, como nome, fonte, quantidade de imagens e doenças incluídas, representadas pelos *labels* definidos anteriormente:

Tabela 1 – *Datasets* pesquisados.

Dataset	**Quantidade de Imagens**	***Rótulos***
ODIR (Grand Challenge, 2019)	8.000	N, D, G, C, A, H, M, O
DRIVE (Grand Challenge, 2018)	40	N, D
STARE (Adam Hoover, 2023)	397	N, D
DIARETDB1 (An Hung Nguyen, 2020)	89	D
IDRiD (PORWAL *et al.*, 2018)	516	D
DDR (LI *et al.*, 2019)	13.673	N, D
Messidor-2 (ADCIS, 2023)	1.744	N, D
APTOS (Kaggle, 2019)	$\sim$ 13.000	N, D
RFMiD (PACHADE *et al.*, 2021)	3.200	N, D, A, M
JSIEC-1000 (CEN *et al.*, 2021)	1000	N, D, M
SMDG (KIEFER *et al.*, 2023)	7.846 [2]	N, G
Total	41.505	-

Fonte: o autor.

Alguns *datasets* foram encontrados porém não utilizados por motivos específicos. Os *datasets* INSPIRE (Carver College of Medicine, 2023), AREDS (CLEMONS *et al.*, 2003), FIRE (HERNANDEZ-MATAS *et al.*, 2017), ARIA/MuReD (RODRÍGUEZ; ALMARZOUQI; LIATSIS, 2022) e RIGA (ALMAZROA *et al.*, 2018) foram descartados por motivos diversos, como dificuldade de se obter acesso e ausência de *labels* para categorizar as imagens, que são necessários pelo motivo dos modelos a serem desenvolvidos ser do tipo supervisionado.

[1] <https://www.kaggle.com/>, acessado em 18 de fevereiro de 2024

[2] Este *dataset* é uma coleção de outros *datasets* individuais (alguns aparecem na lista), portanto o a quantidade de imagens apresentada representa somente as únicas que não se repetem em outro lugar da lista e não o total deste *dataset* independentemente.

4.2 *Baseline*

Definição

O conceito de *baseline*[3] pode ser descrito como uma simples implementação e uma versão menor do que se tem como objetivo desenvolver no trabalho como um todo. Isso pode ser utilizado posteriormente como referência para medições de performance dos códigos feitos com o passar do tempo conforme a evolução e ajustes destes.

Desenvolvimento

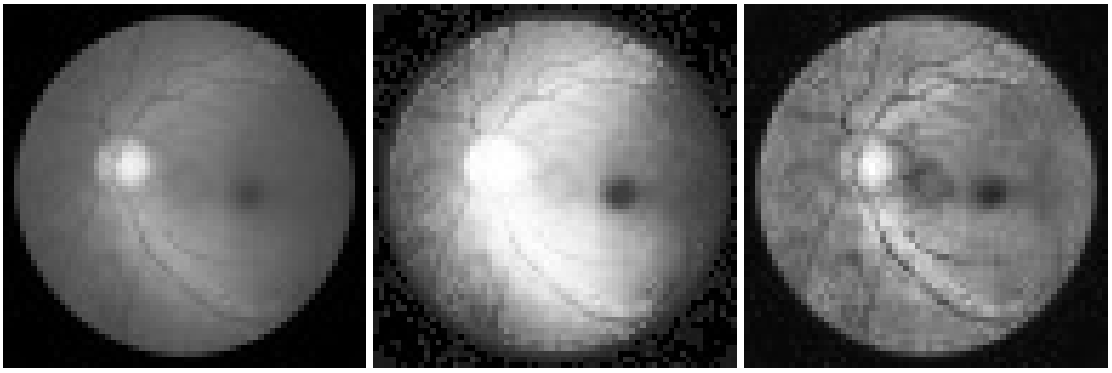
Para o desenvolvimento da *baseline*, foram utilizados dois dos métodos clássicos planejados para serem utilizados no desenvolvimento do trabalho, k-NN e SVM, para simplificar o processo. Além disso, foi utilizado como padrão de pré-processamento, para todos os testes, imagens em tons de cinza e resolução 64×64, isso porque foi observado, em testes prévios, que a informação da cor da imagem original e uma resolução mais alta de imagens não tinham um efeito tão significativo no resultado final de performance para esse problema, enquanto que impactavam significativamente no custo de processamento.

Ainda, foram testados diferentes métodos de redimensionamento da imagem para atingir a resolução próxima a 64×64, buscando não distorcer a imagem, como manter a proporção e eliminar ou adicionar pixels de forma a alterar a proporção original da imagem. Porém, nenhum desses métodos de redimensionamento obteve resultados significativamente superiores em relação ao redimensionamento comum que distorce a imagem.

Além disso, foram testados dois métodos diferentes de pré-processamento adicionais além da imagem redimensionada e em tons de cinza: equalização de histograma comum e CLAHE (*contrast limited adaptive histogram equalization*). Equalizações de histograma, em geral, funcionam através do aumento da diferença entre os tons de cinza próximos e dos extremos claros e escuros da imagem. Com esses métodos, busca-se aumentar o contraste da imagem como uma tentativa de deixar mais claro ao classificador detalhes das imagens que podem ser relevantes na distinção de classes.

É possível visualizar os efeitos visuais desses métodos em imagens de exemplo na figura 5. Aparentemente, através de uma interpretação humana, é possível se chegar na conclusão de que a imagem com equalização de histograma possui detalhes mais exagerados em comparação com a original, devido aos pixels mais escuros da original se tornarem pretos e os mais claros brancos; além disso, através do CLAHE, chega-se à conclusão de que detalhes da imagens, como algumas "linhas" que cruzam esta, ficam mais evidentes. Porém, apesar dessas conclusões humanas, é importante destacar que os classificadores possuem seus próprios meios de funcionamento e tiram diferentes vantagens desses métodos, como será vistos nos resultados encontrados a seguir.

[3] <https://towardsdatascience.com/baseline-models-your-guide-for-model-building-1ec3aa244b8d>, acessado em 18 de fevereiro de 2024



(a) Original (b) Equalização de histograma (c) *CLAHE*

Figura 5 – Exemplo, sobre uma imagem qualquer (a) do *dataset* utilizado, dos métodos de pré-processamento equalização de histograma (b) e *CLAHE* (c).[4]

Foram utilizadas 1.608 imagens de cada uma das classes N e D do *dataset* ODIR, sendo as imagens de teste 20% dos dados. Este *dataset* foi escolhido devido à grande quantidade de imagens e classes que apresenta, o que aumenta a versatilidade e quantidade de possibilidades de testes ao se trabalhar com ele. Além, disso, foram escolhidas essas duas classes por serem a de maior número de imagens rotuladas do *dataset*, possuindo a classe N 2.873 exemplares e a classe D 1.608 exemplares. Dessa forma, pode ser observado que a quantidade de imagens escolhida de classe se limitou pela classe de menor quantidade de imagens das duas, para manter os dados equilibrados.

Nessa fase de desenvolvimento, foram utilizadas somente duas classes do total do *dataset*. Isso tem como objetivo facilitar essa etapa, uma vez que seu objetivo não é extrair o potencial máximo do *dataset*, e sim comparar as diferentes técnicas de pré-processamento utilizadas, que espera-se que tenham resultados semelhantes ao serem utilizadas em cenários mais genéricos e complexos, e se obter uma base de resultados aos quais se basear para desenvolvimentos de etapas posteriores.[5]

Resultados

Os resultados foram coletados através do cálculo das médias das métricas acurácia, precisão, revocação e f1-score e do desvio padrão da acurácia em dez execuções de distribuições diferentes de dados de treino e teste para cada uma das seis configurações: k-NN e SVM aplicado a cada uma dos três tipos de pré-processamento (original, equalização de histograma e CLAHE, conforme visto na figura 5). Em geral, os resultados entre as métricas precisão e revocação para uma mesma configuração foram similares, assim como a acurácia e seu desvio padrão para diferentes configurações, porém com algumas diferenças que serão estudadas com mais

[4] As imagens originais podem ser visualizadas digitalmente através de <https://drive.google.com/drive/folders/1bCWolYT-vTD2UTWdbiFg17fg-zeoHP5d?usp=sharing>.

[5] O código da *baseline* pode ser visualizado em <https://drive.google.com/file/d/1qjJx1UTSn-6wva0K5fMrIci91QGFLxGg/view?usp=sharing>.

detalhe a seguir. As matrizes de confusão obtidas das execuções das seis configuração podem ser observadas abaixo.[6]

$$\begin{array}{c} \\ \text{D} \\ \text{N} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \text{D} & \text{N} \end{array} \\ \begin{bmatrix} 194 & 123 \\ 140 & 185 \end{bmatrix} \end{array}$$

(a) k-NN, original

$$\begin{array}{c} \\ \text{D} \\ \text{N} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \text{D} & \text{N} \end{array} \\ \begin{bmatrix} 192 & 130 \\ 140 & 180 \end{bmatrix} \end{array}$$

(b) k-NN, equal. hist.

$$\begin{array}{c} \\ \text{D} \\ \text{N} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \text{D} & \text{N} \end{array} \\ \begin{bmatrix} 180 & 139 \\ 130 & 194 \end{bmatrix} \end{array}$$

(c) k-NN, CLAHE

$$\begin{array}{c} \\ \text{D} \\ \text{N} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \text{D} & \text{N} \end{array} \\ \begin{bmatrix} 216 & 101 \\ 137 & 188 \end{bmatrix} \end{array}$$

(d) SVM, original

$$\begin{array}{c} \\ \text{D} \\ \text{N} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \text{D} & \text{N} \end{array} \\ \begin{bmatrix} 226 & 96 \\ 154 & 166 \end{bmatrix} \end{array}$$

(e) SVM, equal. hist.

$$\begin{array}{c} \\ \text{D} \\ \text{N} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \text{D} & \text{N} \end{array} \\ \begin{bmatrix} 216 & 102 \\ 144 & 180 \end{bmatrix} \end{array}$$

(f) SVM, CLAHE

Comparações

De todas as seis configurações, o melhor resultado de acurácia foi do SVM com configuração original de pré-processamento, com aproximadamente 63%; por outro lado, o pior resultado de acurácia foi do k-NN com equalização de histograma, com aproximadamente 58%. Portanto, podemos concluir que a maior variação entre duas configurações (aproximadamente 5%) não foi tão expressiva.

Apesar disso, podemos notar algumas diferenças. A média de acurácia das configurações do k-NN foi de aproximadamente 58%, enquanto que a média de acurácia das configurações do SVM foi de aproximadamente 62%. Com isso, podemos concluir que, em geral, o SVM obteve melhores resultados independentemente das configurações de pré-processamento, uma vez que estas não afetaram expressivamente nenhum dos métodos, que pouco variaram do pior (para as configurações do k-NN) e melhor resultado (para as configurações do SVM).

Por fim, podemos concluir que os métodos utilizados mantiveram uma boa constância de resultados entre as execuções, uma vez que o desvio padrão de acurácia entre as execuções variaram entre aproximadamente 1.5% e 1.9% entre todas as configurações. Além disso, apesar da acurácia maior por parte do SVM, pode ser observado nas matrizes um maior desequilíbrio de classificações por parte deste: a quantidade de classificações corretas deste para imagens da classe D aparenta ser generalizadamente maior para todas as configurações em relação à classificações corretas da classe N; no k-NN, a quantidade de classificações corretas de cada classe aparenta ser mais equilibrada.

Através desses resultados, é possível estabelecer um referencial para outros desenvolvimentos mais complexos, com cuidados maiores para se configurar o modelo. É possível concluir

[6] Os resultados completos podem ser obtidos através de <https://drive.google.com/file/d/1YLVCv2VoOasIrRE1lHdUpGGnXjpaNaUL/view?usp=sharing>.

também que os métodos de pré-processamento pouco alteraram a acurácia, o que indica que a equalização de histograma e o *CLAHE* não foram mais eficazes comparado a somente redimensionar e dessaturar as imagens, apesar de terem aumentado a carga de processamento.

É possível ainda observar que os resultados obtidos das configurações utilizadas na *baseline*, que foram, em média, próximos de 60%. Esses resultados podem ser considerados como ruins, pois pouco passam de 50%, que é o mínimo esperado para classificações binárias, caso onde pode-se dizer que o classificador é inútil, por não acertar mais do que errar. Isso indica que novas técnicas devem ser testadas nas próximas etapas para tentar se obter resultados melhores. O fato de que somente duas classes foram utilizadas na presente etapa também pode indicar que em etapas posteriores, onde serão utilizadas mais classes, pode não ser trivial obter bons resultados.

Embora encontrar as configurações dos modelos que proporcionem as melhores performances possíveis não seja o foco maior do presente estudo, e sim compará-los, é desejável que se aproximem de seus potenciais máximos. Isso deve ser buscado para que a comparação seja justa para cada modelo, representando suas capacidades da melhor forma possível.

Limitações

Vale ressaltar também uma limitação do método simplificador utilizado para separar os dados de treino e de teste. Ou seja, dos dados totais do *dataset* utilizado, a separação foi feita aleatoriamente. Isso permite casos que causam o problema de *data leakage*[7], ou seja, dados indevidos no conjunto de testes que podem enviesar os resultados. Problemas como esse devem ser corrigidos em etapas posteriores, onde se deseja obter resultados mais fiéis ao modelo em produção, onde são utilizados dados novos, não conhecidos pelo modelo.

Como a separação é aleatória, um par de olhos pode ser dividido entre o conjunto de treino e de teste. Isso pode causar viés, uma vez que classificações corretas de instâncias do conjunto de testes podem ser influenciadas por aprendizados de características específicas da pessoa a qual pertence o par de olho "dividido". Esse aprendizado indevido pode trazer um aumento na performance que não revela a capacidade real do modelo.

Contudo, foi feita a escolha por não resolver esse problema na etapa atual pois espera-se, com o *data leakage*, que os resultados sejam "inflados" de forma não muito significativa, uma vez que ocorre somente entre pares de dados, e não de forma muito generalizada. Portanto, para os propósitos de comparação esperados da *baseline*, o resultado obtido, mesmo que não totalmente preciso, é aceitável.

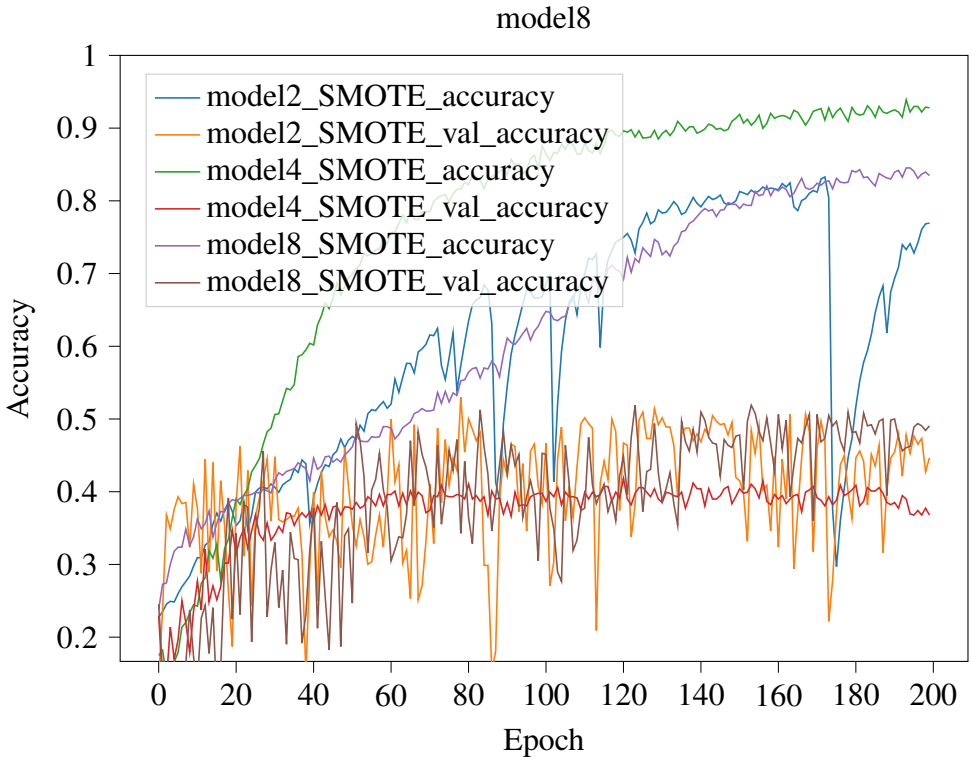
[7] <https://scikit-learn.org/stable/common_pitfalls.html#data-leakage>, acessado em 18 de fevereiro de 2024.

4.3 Resultados e comparações

Continuando com a discussão anterior a respeito das limitações da *baseline*, para se resolver o problema de *data leakage*, foi desenvolvido um algoritmo[8] de separação entre conjunto de teste e treino que tem como restrição não permitir que um par de olhos de uma mesma pessoa esteja dividido entre treino e teste. Dessa forma, o par de olhos sempre estará ou no conjunto de testes ou de treino, evitando assim o *data leakage* mostrado anteriormente.

9 10

Figura 7 – Gráfico.



Fonte: o autor.

Incompleto...

[8] O código fonte do algoritmo pode ser obtido através de <https://drive.google.com/file/d/1DzOVv_ggYOa4wF-c_qYvht2A0C5u9N5o/view?usp=sharing>.

[9] Os resultados da execução dos algoritmos de aprendizagem profunda podem ser obtidos através de <https://drive.google.com/file/d/1LgVN7a7-GLRjxgq_RYUjK1uzVuCzPPic/view?usp=drive_link>

[10] Os resultados da execução dos algoritmos clássicos podem ser obtidos através de <>

5 CRONOGRAMA

O seguinte cronograma (tabela 2) foi desenvolvido de acordo com os prazos de entrega, seguindo os objetivos específicos e gerais e adequando os intervalos de tempo de acordo com a exigência esperada de cada atividade.

Tabela 2 – Cronograma de Atividades

Mês/Ano	**Objetivo(s) Específico(s)**
09/2023	1, 2, 3 e 4 (Pesquisas e desenvolvimento da *baseline*)
10/2023	5 e 6 (Regulagem dos softwares)
11/2023	7 (Desenvolvimento dos modelos)
12/2023	7 (Desenvolvimento dos modelos)
01/2024	7 (Desenvolvimento dos modelos)
02/2024	8, 9 e 10 (Conclusões)

Fonte: o autor.

6 CONCLUSÃO

6.1 Objetivos específicos

Objetivo específico 1 (“Desenvolvimento da fundamentação teórica”)

Foram levantados conceitos que “sustentam” os elementos utilizados durante o desenvolvimento das aplicações, além de darem base às comparações feitas entre os resultados destas. Foi relevante na escolha das técnicas de AM que seriam utilizadas, uma vez que foi possível escolher métodos distintos entre si, o que torna a comparação mais interessante. Além disso, deu base para interpretar os resultados dados pelas métricas de performance, uma vez que foi visto que não há uma métrica única capaz de dar a performance de uma execução, mas sim um conjunto de métricas que demonstram a consistência de um algoritmo.

Objetivo específico 2 (“Definir técnicas de AM”)

Pode-se dizer que o objetivo de definir as técnicas de AM clássicas utilizadas foi cumprido com êxito, uma vez que, da grande variedade de métodos de classificação existentes, que pode ser visto na figura 1, foram reunidos um conjunto de quatro métodos (k-NN, SVM, floresta aleatória e Naive Bayes) que compreendem bem diferentes fundamentos por trás dos classificadores, conforme explorado na seção 3.7. Essa grande variedade de tipos de classificadores é importante para o objetivo geral pois a comparação se torna mais significante, uma vez que resultados diferentes são esperados.

Objetivo específico 3 (“Definição do conjuntos de imagens”)

Pode-se concluir que a pesquisa por *datasets* foi parcialmente bem sucedida, tendo sido encontrado uma grande quantidade de dados, como pôde ser visto na tabela 1. Apesar dessa grande quantidade, pode ser dito que não há um equilíbrio entre as classes desses dados, devido a uma grande quantidade de *datasets* possuírem poucos rótulos ou serem desbalanceados entre estes, sendo a maioria das imagens de rótulos N e D. Uma hipótese para esse desbalanceamento é a raridade de certas doenças, o que dificulta seus registros, como foi visto na seção 3.3.

Além disso, uma dificuldade encontrada foi a inclusão de todos os *datasets* na aplicação, uma vez que os formatos de rotulação dos dados de cada um varia. Isso faz com que a inclusão de todos seja dificultado devido ao tratamento particular requerido para cada um para se extrair os rótulos das instâncias. Incluir imagens adicionais de diferentes *datasets* poderia ser um fator de aumento na performance dos classificadores em geral. Contudo, ainda pode ser adicionado que esse problema pode não ser tão relevante devido ao objetivo geral ser a comparação entre os métodos de classificação, com um aumento de performance geral potencialmente não sendo tão importante.

Objetivo específico 4 (“Desenvolvimento da *baseline*”)

Foi desenvolvida uma *baseline* capaz de dar uma base de performance a ser atingida pelos demais algoritmos, com a esperança de que os resultados posteriores com modelos mais robustos fossem ao menos melhores. Além disso, ainda mais relevante, demonstrou quais técnicas de pré-processamento teriam um efeito mais significativo na performance, possibilitando um desenvolvimento posterior mais pontual sem perca de tempo, uma vez que o processamento da *baseline* é, em geral, menos exigente de recursos computacionais, economizando tempo.

Objetivo específico 5 e 6 (pré-processamento e configurações dos modelos)

Além disso, diferentes técnicas de pré-processamento foram estudadas (seção 3.6), testadas e comparadas (4.2), para se obter o melhor desempenho geral entre os métodos. Ainda, foram analisadas diferentes configurações para cada um dos métodos para se extrair o melhor resultado possível.

Objetivo específico 7 (“Desenvolvimento das técnicas de AM”)

O objetivo específico 7, que diz respeito ao desenvolvimento dos métodos de AM selecionados, além da aprendizagem profunda, foi realizados com sucesso. Através da assistência do módulo Python scikit-learn, pôde ser desenvolvido os classificadores clássicos com facilidade, com a alteração de poucas linhas de código para se utilizar diferentes classificadores[1]; o mesmo pode ser dito da aprendizagem profunda[2], através do módulo Python TensorFlow, que permite ao programador customizar facilmente as camadas de neurônios do modelo, com o módulo se encarregando do restante.

Objetivo específico 8 (“Aplicação dos modelos e coleta dos resultados”)

Além disso, a etapa de coleta dos resultados também foi concluída com sucesso, tendo sido utilizadas diferentes métricas de performance de classificação, estudados na seção (3.5), relevantes para se observar o desempenho dos métodos sob diferentes perspectivas, trazendo várias possibilidades de interpretações realizadas na etapa posterior de comparação dos resultados 4.3.

Objetivo específico 9 (“Comparação dos resultados”)

Sobre o objetivo específico de se comparar os resultados dos diferentes métodos de classificação utilizados, pode-se considerar que este foi atingido, uma vez que se encontrou distinções significativas entre eles, além de vantagens e desvantagens em diferentes casos. Através

[1] O código fonte da implementação dos classificadores podem ser acessados através de <>

[2] O código fonte da aplicação de aprendizagem profunda pode ser acessada através de <>

dessas diferenças, é possível extrair conhecimento dando sentidos a elas, chegando assim em uma conclusão maior, fato que conclui o objetivo geral do trabalho.

6.2 Objetivo geral

A respeito do objetivo geral, resumidamente, o de se chegar a uma conclusão sobre a eficácia e praticidade dos diferentes métodos de classificação utilizados para o problema proposto, foi encontrado...

É interessante notar também que, apesar do objetivo geral maior ser o estudo comparativo, ainda surgiram outras contribuições relevantes do estudo de forma natural, devido às pesquisas que foram feitas como base para o desenvolvimento. Por exemplo, pode-se destacar a fundamentação teórica levantada que resume de certa forma técnicas de classificação de AM; além disso, também é possível apontar o levantamento de *datasets* que foi feito na seção 4.2, que pode servir para outros estudos posteriores, apesar de não ter sido utilizado no presente.

6.3 Melhorias e pesquisas futuras

Apesar de ter sido possível se chegar a diversas conclusões relevantes através dos resultados encontrados com o desenvolvimento deste presente estudo, como foi visto anteriormente, muitas coisas foram deixadas de fora necessariamente devido ao seu escopo, que priorizou certos resultados em detrimento de outros naturalmente. Portanto, torna-se relevante considerar as limitações do que foi estudado, abrindo, assim, possibilidades para pesquisas futuras seguindo o mesmo tema.

A primeira coisa a qual o presente estudo se limitou foi a natureza das imagens utilizadas para serem classificadas. Fazendo isso, parte da interpretação dos resultados encontrados passam a estar ligados a esse fato. Assim, mantendo-se as mesmas configurações dos classificadores utilizadas, porém sobre um diferente conjunto de imagens, poderia resultar em diferentes valores de performance. Dessa forma, é possível concluir que o estudo comparativo dos métodos de classificação utilizados sobre diferentes conjuntos de imagens é interessante como uma pesquisa futura.

Além disso, algumas configurações se mantiveram sem alteração para diferentes métodos, por indução, das melhores performances obtidas na fase da *baseline* (seção 4.2), como o pré-processamento utilizado. Pesquisas futuras poderiam se concentrar na utilização de diferentes configurações e técnicas específicas para cada método buscando elevar o potencial de performance de cada um. Isso ocorre pois os métodos se comportam de formas diferentes dependendo de como são tratados, devido às suas naturezas diferentes no funcionamento de seus algoritmos. Adicionalmente, outros métodos poderiam ser testados para se tentar melhorar a performance para o problema escolhido.

Além de mudanças qualitativas nas configurações dos algoritmos, ainda pode ser dado como sugestão alterações quantitativas, ou seja, aumento da quantidade de imagens utilizadas, que, como pôde ser visto na tabela 1, ainda existem muitas disponíveis que não foram aproveitadas. O aumento da quantidade de imagens para treino e validação pode resultar em performances melhores dos métodos em geral, uma vez que o modelo terá novas características para aprender da grande variedade de dados. Contudo, ainda deve ser levado em conta o custo computacional que terá que ser despendido com o aumento do conjunto de dados.

Por fim, uma sugestão de pesquisa futura seria a utilização de diferentes métricas de performance em relação as utilizadas, buscando encontrar diferentes formas de avaliar o desempenho dos algoritmos. Um tipo de métrica não explorada no presente estudo foi o custo computacional (tempo de execução e memória gastos pelos algoritmos), uma vez que foi considerado somente que a técnica de aprendizagem profunda possui um custo computacional maior no geral, sem que medidas mais precisas fossem realizadas. Através de medições de métricas como essas, novas possibilidades poderiam surgir dos resultados, como comparação do custo computacional entre os métodos clássicos ou identificação de custos computacionais não esperados para determinado método.

Como conclusão geral, podemos afirmar que o presente trabalho estudou apenas uma pequena parte das inúmeras possibilidades oferecidas pelo tema de comparações de algoritmos de AM. Alguns exemplos de sugestões de pesquisas futuras foram dadas acima, porém é importante perceber o fato de que o presente estudo fez comparações a partir de um conjunto limitado de "parâmetros". Sendo assim, uma quantidade quase ilimitada de novas comparações e possíveis conclusões podem ser alcançadas através da utilização de diferentes pontos os quais podem ser levados em consideração, contanto que seja mantido uma consistência e reconhecido suas limitações.

REFERÊNCIAS

Adam Hoover. **STructured Analysis of the Retina**. 2023. <https://cecas.clemson.edu/~ahoover/stare/>. Acessado em 19 de setembro de 2023.

ADCIS. **Messidor-2**. 2023. <https://www.adcis.net/en/third-party/messidor2/>. Acessado em 19 de setembro de 2023.

AHMED, S. F. *et al.* Deep learning modelling techniques: current progress, applications, advantages, and challenges. **Artificial Intelligence Review**, Springer, p. 1–97, 2023.

ALMAZROA, A. *et al.* Retinal fundus images for glaucoma analysis: the riga dataset. In: SPIE. **Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications**. [S.l.], 2018. v. 10579, p. 55–62.

An Hung Nguyen. **DiaRetDB1 V2.1**. 2020. <https://www.kaggle.com/datasets/nguyenhung1903/diaretdb1-v21>. Acessado em 19 de setembro de 2023.

BAYES, T. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. **Philosophical transactions of the Royal Society of London**, The Royal Society London, n. 53, p. 370–418, 1763.

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

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.: s.n.], 1992. p. 144–152.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001.

Carver College of Medicine. **Inspire Datasets**. 2023. <https://medicine.uiowa.edu/eye/inspire-datasets>. Acessado em 20 de setembro de 2023.

CEN, L.-P. *et al.* Automatic detection of 39 fundus diseases and conditions in retinal photographs using deep neural networks. **Nature communications**, Nature Publishing Group UK London, v. 12, n. 1, p. 4828, 2021.

CLEMONS, T. E. *et al.* National eye institute visual function questionnaire in the age-related eye disease study (areds): Areds report no. 10. **Archives of ophthalmology**, American Medical Association, v. 121, n. 2, p. 211–217, 2003.

COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE transactions on information theory**, IEEE, v. 13, n. 1, p. 21–27, 1967.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000.

Grand Challenge. **DRIVE: Digital Retinal Images for Vessel Extraction**. 2018. <https://drive.grand-challenge.org/>. Acessado em 19 de setembro de 2023.

____. **Metrics and scoring: quantifying the quality of predictions**. 2019. <https://odir2019.grand-challenge.org/>. Acessado em 19 de setembro de 2023.

HERNANDEZ-MATAS, C. *et al.* Fire: fundus image registration dataset. **Modeling and Artificial Intelligence in Ophthalmology**, v. 1, n. 4, p. 16–28, 2017.

JEONG, Y.; HONG, Y.-J.; HAN, J.-H. Review of machine learning applications using retinal fundus images. **Diagnostics**, mdpi, v. 12, n. 1, p. 134, 2022.

KADAM, V. *et al.* Enhancing surface fault detection using machine learning for 3d printed products. **Applied System Innovation**, MDPI, v. 4, n. 2, p. 34, 2021.

Kaggle. **APTOS 2019 Blindness Detection**. 2019. <https://www.kaggle.com/competitions/aptos2019-blindness-detection/data>. Acessado em 19 de setembro de 2023.

KANSKI, J. J.; BOWLING, B. **Clinical ophthalmology: a systematic approach**. [S.l.]: Elsevier Health Sciences, 2011.

KIEFER, R.; ABID, M.; ARDALI, M.; STEEN, J.; AMJADIAN, E. Automated fundus image standardization using a dynamic global foreground threshold algorithm. In: **2023 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME); accepted for publication**. [S.l.: s.n.], 2023.

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

LI, T. *et al.* Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening. **Information Sciences**, v. 501, p. 511 – 522, 2019. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025519305377>.

LIAKOS, K. G.; BUSATO, P.; MOSHOU, D.; PEARSON, S.; BOCHTIS, D. Machine learning in agriculture: A review. **Sensors**, Mdpi, v. 18, n. 8, p. 2674, 2018.

MAHESH, B. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR).[Internet]**, v. 9, n. 1, p. 381–386, 2020.

MITCHELL, T. M. **Machine learning**. [S.l.]: McGraw-hill, 1997.

MÜLLER, A. C.; GUIDO, S. **Introduction to machine learning with Python: a guide for data scientists**. [S.l.]: "O'Reilly Media, Inc.", 2016.

PACHADE, S. *et al.* Retinal fundus multi-disease image dataset (rfmid): A dataset for multi-disease detection research. **Data**, MDPI, v. 6, n. 2, p. 14, 2021.

PETERSON, L. E. K-nearest neighbor. **Scholarpedia**, v. 4, n. 2, p. 1883, 2009.

PORWAL, P. *et al.* Indian diabetic retinopathy image dataset (idrid): a database for diabetic retinopathy screening research. **Data**, MDPI, v. 3, n. 3, p. 25, 2018.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, p. 81–106, 1986.

RISH, I. *et al.* An empirical study of the naive bayes classifier. In: **IJCAI 2001 workshop on empirical methods in artificial intelligence**. [S.l.: s.n.], 2001. v. 3, n. 22, p. 41–46.

RODRÍGUEZ, M. A.; ALMARZOUQI, H.; LIATSIS, P. Multi-label retinal disease classification using transformers. **IEEE Journal of Biomedical and Health Informatics**, IEEE, 2022.

TONG, Y.; LU, W.; YU, Y.; SHEN, Y. Application of machine learning in ophthalmic imaging modalities. **Eye and Vision**, BioMed Central, v. 7, n. 1, p. 1–15, 2020.