Departamento de Filosofia, Ciencias e Letras de Ribeirão Preto - FFCLRP
Mestrado em Física Aplicada à Medicina e Biologia

Mapas de melanina e oxiemoglobina para auxiliar redes neurais na classificação de imagens de retina

André Riccieri Albinati Silva Vitor

Ribeirão Preto - SP, 2023

André Riccieri Albinati Silva Vitor

Mapas de melanina e oxiemoglobina para auxiliar redes neurais na classificação de imagens de retina

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da Universidade de São Paulo, como parte das exigências para a obtenção do título de Mestre em Ciências.

Área:	Física Aplicada à Medicina e Biologia
Orientador:	Prof. Dr. George C. Cardoso

Agosto de 2023

Vitor, André Riccieri Albinati Silva

Mapas de melanina e oxiemoglobina para auxiliar redes neurais na classificação de imagens de retina / André Riccieri Albinati Silva Vitor; Orientador: George Cunha Cardoso. Ribeirão Preto, 2022: 60p.

Dissertação (Mestrado - Programa de Pós-Graduação em Física Aplicada à Medicina e Biologia) - Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da Universidade de São Paulo.

1. Redes Neurais Convolucionais; 2. Melanina; 3. Oxiemoglobina; 4. Fundo de Olho

FFCLRP - Departamento de Filosofia, Ciencias e Letras de Ribeirão Preto
Universidade de São Paulo

Dissertação apresentada ao Departamento de Física da Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da Universidade de São Paulo, intitulado ***Mapas de melanina e oxiemoglobina para auxiliar redes neurais na classificação de imagens de retina*** de autoria de André Riccieri Albinati Silva Vitor, como parte das exigências para a obtenção do título de Mestre em Ciências.

Ribeirão Preto, 22 de Agosto de 2022

AGRADECIMENTOS

Aos meus pais, pelas oportunidades que me deram.

Ao prof. George Cunha Cardoso, que me orientou neste trabalho.

Aos meus amigos que sempre me acompanharam: Renata Pazzini, Gabriela Arcos e Gabriela Francisco.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

RESUMO

Vitor A., R. **Mapas de melanina e oxiemoglobina para auxiliar redes neurais na classificação de imagens de retina**. Dissertação (Mestrado - Programa de Pós-Graduação em Física Aplicada à Medicina e Biologia) - Faculdade de Filosofia Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto - SP, 2022.

A maioria das doenças oculares podem ser prevenidas e tratadas de maneira eficaz se detectadas precocemente. As redes neurais podem acelerar os processos de triagem das patologias oculares ao identificar previamente a presença de alguma doença. Uma vez que várias doenças estão relacionadas com o nível de oxigenação dos olhos, esta pode ser uma característica importante a ser analisada, visando o diagnóstico e tratamento. Neste trabalho utilizamos os mapas de densidade de oxiemoglobina e melanina em imagens de retinografia combinados de duas formas: 3 canais (tons de cinza, melanina e oxiemoglobina) ou 5 canais (vermelho, verde, azul, melanina e oxiemoglobina) e testamos essas combinações em 3 redes neurais convolucionais sendo uma delas um modelo pré-treinado.

Palavras-chave: Redes Neurais Convolucionais, Melanina, Oxiemoglobina, Fundo de Olho.

ABSTRACT

Vitor A., R. **Melanin and oxyhemoglobin maps to improve neural networks in eye fundus images classification**. Dissertação (Mestrado - Programa de Pós-Graduação em Física Aplicada à Medicina e Biologia) - Faculdade de Filosofia Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto - SP, 2022

Several eye diseases can be prevented and treated effectively if detected early. Neural networks can speed up the processes of screening for eye diseases by identifying the presence of some disease in advance. Since several diseases are related to the oxygenation level of the eyes, this can be an important feature to be analyzed. In this work we use oxyhemoglobin and melanin density maps as new channels in convolutional neural networks and compare the results with pre-trained models.

Key-words: Convolutional Neural Networks, Melanin, Oxyhemoglobin, Eye Fundus

LISTA DE FIGURAS

LISTA DE TABELAS

Conteúdo

1 Introdução

O ser humano possui cinco sentidos: visão, paladar, olfato, audição e tato, que é como se mantém em contato com o meio ambiente. No entanto, nosso sentido da visão, que tem os olhos como órgãos sensoriais, é um dos mais importantes, pois a maioria das informações que percebemos chega até nós por meio dele. Por esta razão, manter uma boa saúde ocular é essencial em todos os aspectos e atividades de nossas vidas diárias.

A oftalmologia é a parte da medicina que se ocupa do estudo do olho, do diagnóstico e tratamento das diferentes doenças que este pode apresentar. Segundo dados da Organização Mundial da Saúde (OMS), quase 285 milhões de pessoas apresentam problemas oculares no mundo, dos quais entre 60% a 80% dos casos podem ser evitados e tratados [1]. No Brasil, o Censo Demográfico de 2010 identificou mais de 35 milhões de pessoas com algum grau de dificuldade visual [1]. "As doenças oculares mais comuns e que podem ocasionar cegueira irreversível são glaucoma, retinopatia diabética e degeneração macular relacionada à idade, enquanto que uma das principais doenças oculares que podem causar cegueira reversível é a catarata.[1, 2].

Para o estudo de diferentes doenças, a oftalmologia lida muito com a análise de imagens de exames clínicos. Mas muitas vezes os pacientes com doenças oculares não percebem o agravamento dos casos assintomáticos e a doença é detectada quando já está em um estágio muito avançado. [2, 3].

Procurando uma melhor forma de monitorar e diagnosticar as doenças oculares, tem sido desenvolvidos algoritmos de inteligência artificial (IA). Avanços recentes na computação, como também a disponibilidade de grandes *datasets*, tem o potencial de melhorar os diagnósticos. Aprendizado profundo ou *Deep Learning* (DL) se tornou uma prática comum em inteligência artificial, com performances iguais ou até melhores do que as de humanos como por exemplo, em 2012, um artigo de Krizhevsky et al. descreveu como uma rede neural alcançou resultados superiores aos humanos em uma tarefa de reconhecimento de imagem chamada ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [4]. DL é uma família de métodos computacionais que permitem que um algoritmo aprenda a partir de um grande conjunto de dados que apresentam o comportamento desejado [5].

A maioria das aplicações de DL em imagens de fundo podem ser divididas em classificação e segmentação. A segmentação de imagens é o processo de dividir uma imagem em várias regiões ou segmentos, cada um com características semelhantes. Isso é útil para destacar objetos ou áreas específicas na imagem e para facilitar a análise e processamento posterior. Existem vários métodos de segmentação, incluindo baseado em limiar, cor, textura, forma e aprendizado de máquina. Classificação pode ser ainda dividida em identificação da presença de uma doença ou caracterização do nível de doença, como por exemplo os graus de retinopatia diabética [6, 7].

Algumas técnicas para te obter imagens em oftalmologia são a Tomografia de Coerência Óptica (OCT), Tomografia de Coerência Óptica com Angiografia (OCT-A) e Retinografia. O OCT é um exame não invasivo que utiliza a interferência de luz para produzir imagens detalhadas dos tecidos oculares, como retina e nervo óptico. É amplamente utilizado para o diagnóstico e monitoramento de doenças oculares, como degeneração macular, glaucoma e diabéticas retinopatia. O OCT funciona enviando um raio de luz através do olho e medindo a intensidade de luz que é refletida de volta. Essa medida é usada para produzir imagens tridimensionais dos tecidos oculares, permitindo a visualização de estruturas internas com alta resolução. É um exame seguro, rápido e indolor, que ajuda a detectar e monitorar problemas oculares de forma precisa e eficiente.

O OCT-A é uma técnica de exame que combina as características do OCT com as da angiografia. Ela permite a visualização detalhada da microcirculação sanguínea na retina e na coroide, bem como a estrutura da retina. O OCT-A funciona ao medir a variação na velocidade de fluxo de sangue nos vasos sanguíneos da retina e coroide. Isso é feito através de um escaneamento laser que gera imagens da vascularização dos tecidos oculares em diferentes níveis de profundidade. A Retinografia é uma técnica de exame que consiste em observar e registrar fotografias da retina, do nervo óptico e do fundo do olho.

No presente trabalho vamos lidar com classificação de imagens, uma vez que queremos identificar em uma imagem se pode existir a presença de alguma patologia.

2 Referencial teórico

2.1 Principais doenças oculares

Nosso olho é o órgão sensorial mais importante de todos, porque graças a ele conseguimos realizar grande parte de nossas atividades diárias. É um órgão tão complexo quanto sensível. Existem muitas doenças que podem causar sérias consequências à nossa visão se não forem detectadas e tratadas a tempo. Elas vão desde doenças oculares que não causam cegueira, como miopia, opacidades vítreas e estrabismo, até doenças que causam cegueira reversível e cegueira irreversível como glaucoma, degeneração macular relacionada à idade e retinopatia diabética.

2.1.1 Glaucoma

O glaucoma é um problema mundial de saúde atingindo cerca de 60 milhões de pessoas em todo mundo, das quais 8,4 milhões ficaram cegas [8]. O seu diagnóstico precoce resulta em melhores formas de tratamento e maiores chances de evitar a cegueira.

Glaucoma é um conjunto de doenças que resulta em danos no nervo óptico e retina e causa perda da visão. Alguns tipos de glaucoma se desenvolvem lentamente e sem a presença de dor. Pouco a pouco a visão periférica diminui e eventualmente a visão central também, caso não seja tratada. Uma vez que a perda de visão ocorre, é impossível reverter o caso, ou seja, a perda de visão é permanente. Um dos principais indicativos de glaucoma é a alta pressão intraocular [9]. Glaucoma é mundialmente a segunda doença que mais causa cegueira, atrás apenas da degeneração macular para cegueira irreversível, ou catarata, para cegueira reversível. [10]

2.1.2 Degeneração macular relacionada à idade

A degeneração macular relacionada à idade (DMRI) vem sendo considerada o principal motivo de cegueira em pessoas acima de 40 anos [11]. Hoje, sabemos que fatores genéticos têm forte influencia sobre essa condição, mas precisamos levar em consideração estilo de vida, sexo e outros fatores. Embora ainda seja uma incógnita o motivo, sabe-se que a degeneração macular está ligada diretamente com a idade [12]. A DMRI é uma doença crônica progressiva da retina central promovendo degeneração e levando à cegueira. Ela é causada por acúmulo de material extracelular na mácula (região posterior central da retina) isso aliada a fatores como cardiopatias, pacientes fumantes e outros. Há duas décadas a DMRI era considerada sem tratamento[13], mas hoje, fármacos que promovem a supressão do fator de crescimento endotelial vascular (VEGF) mudou substancialmente a forma de tratar a doença [14].

A imagem de retina é habitualmente utilizada para detectar drusas (massas amareladas), onde são verificadas a quantidade e a qualidade morfológica para classificação da DMRI. Assim, procurando diagnosticar, temos [15]:

- exames não-invasivos, como OCT e autofluorescência;
- exames invasivos, como angiografia.

Porém, esses exames precisam ser feitos por profissionais específicos para um diagnóstico correto. Além disso, são exames que podem custar bem caro e não estão disponíveis em todas as regiões [15] fazendo com que boa parte da população não tenha acesso aos exames, aumentando assim a dificuldade no tratamento.

2.1.3 Retinopatia diabética

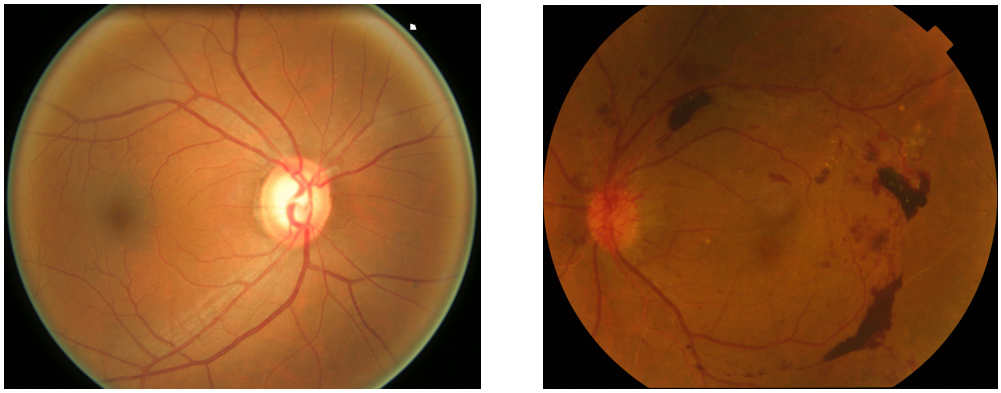
O diabetes mellitus é uma condição multifatorial e complexa que pode levar a graves complicações em vários órgãos e sistemas, incluindo o sistema ocular. Em particular, a disfunção vascular na retina pode causar danos e eventual perda da visão.[16]. O diabetes é uma doença metabólica definida por aumento de glicose no sangue e descontrole de insulina (hormônio produzido no pâncreas) causando picos desordenados de glicose interferindo no armazenamento de energia dos indivíduos. Sedentarismo, genética e envelhecimento são alguns dos fatores de risco para o desenvolvimento desta doença [17]. Estudos mostram que a gravidade da diabetes e a disfunção na retina estão ligadas diretamente: quanto maior o tempo que o paciente for acometido pela doença , maiores as chances de ter uma manifestação da retinopatia diabética [16].

2.2 Técnicas de auxílio ao diagnóstico oftalmológico

2.2.1 Retinografia

A retina é uma camada de tecido encontrada dentro do olho humano que tem um papel muito importante na visão humana, pois capta a luz e a converte em sinais elétricos [18]. A técnica utilizada na medicina para obter imagens coloridas da retina é conhecida como retinografia [19] e é amplamente utilizada para detectar diferentes doenças, como retinopatia diabética (Fig. 2.1), retinopatia hipertensiva, retinite pigmentosa e degeneração macular relacionada à idade [20].

A retinografia é realizada com as chamadas "câmeras de fundo de olho", cujo desenho óptico é baseado na oftalmoscopia indireta monocular, pois os sistemas de observação e iluminação seguem caminhos distintos [22, 23]. Essas câmeras proporcionam uma visão vertical e ampliada do fundo de olho e permitem a visualização da área retiniana de 30 a



(a) Olho saudável **(b) Olho com retinopatia diabética**

Figura 2.1: Retinografia de a) um olho saudável e b) um olho com retinopatia diabética. Imagens retiradas do *dataset* [21].

50°. Além disso, existe a possibilidade de modificações com zoom ou lentes auxiliares ou lente grande angular [23].

A luz de observação em um retinógrafo é focalizada por uma série de lentes através de uma abertura em forma anelar. Este feixe então passa por uma abertura central para formar um anel e depois pela lente objetiva da câmera para passar pela córnea até a retina [23]. A luz refletida da retina passa pelo orifício anelar do sistema de iluminação. Os raios formadores de imagem continuam até a ocular telescópica de baixa potência. Para tirar uma fotografia da retina, um espelho interrompe o caminho do sistema de iluminação e permite que a luz da lâmpada do flash passe para o olho. Nesse momento, um espelho cai na frente do telescópio de observação, e se encarrega de redirecionar a luz para o meio de captura, seja um filme ou câmera digital.

2.2.2 OCT e OCTA

A tomografia de coerência óptica (OCT) é uma técnica diagnóstica não invasiva que gera imagens transversais das distintas camadas da retina. Além de ser uma técnica não invasiva, ela fornece imagens de alta resolução, rápidas, sem dor, sem efeitos adversos e não precisa de uma preparação prévia. A OCT é útil para o tratamento e a detecção precoce de patologias que afetam a retina, como, por exemplo, a diabetes ocular, edema macular, miopia e degeneração macular relacionada à idade. Também tem sido amplamente usada em tratamentos de glaucoma uma vez que a OCT é sensível na detecção da progressão do glaucoma no seu estágio inicial, o que pode facilitar seu monitoramento [24, 25].

A obtenção da imagem em OCT é baseada na interferometria de baixa coerência (Fig. 2.2). Uma luz é gerada a partir de um diodo infravermelho de baixa coerência ou uma fonte de laser de femtossegundos, que penetra nos tecidos oculares. O feixe é

dividido uniformemente em dois caminhos ópticos, um dos quais penetra na retina e o outro reflete em um espelho de referência. A interferência ocorre apenas quando as ondas refletidas de volta coincidem dentro do comprimento de coerência da fonte, e seu atraso ou diferença de caminho é medido para saber a profundidade no olho em que a reflexão ocorreu [24, 26].

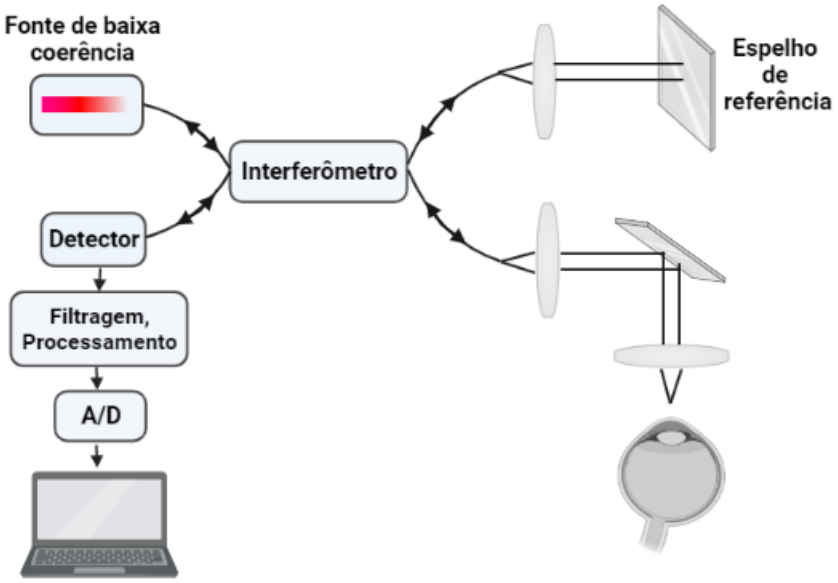


Figura 2.2: Esquema da tomografia de coerência óptica (OCT).

A angiotomografia de coerência óptica (OCTA) é um método recente e complementar baseado na OCT. A OCTA incorpora o contraste gerado pelo movimento das células sanguíneas nos vasos da retina contra o tecido estático circundante. A luz refletida pela superfície dos glóbulos vermelhos em movimento permitem a visualização em alta resolução de imagens angiográficas volumétricas da retina de uma forma não invasiva e sem a necessidade de usar algum elemento de contraste (Fig. 2.3) [27, 28].

2.3 Inteligência Artificial

Muitas vezes há confusão com os termos de inteligência artificial, aprendizado de máquina e aprendizado profundo. Esses são termos relacionados mas diferentes (Fig. 2.4).

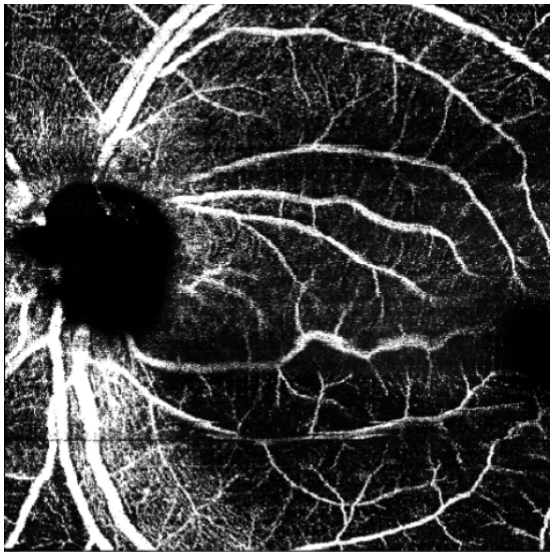


Figura 2.3: Recorte de uma imagem OCTA. Quanto mais clara a imagem maior o movimento dos glóbulos vermelhos.

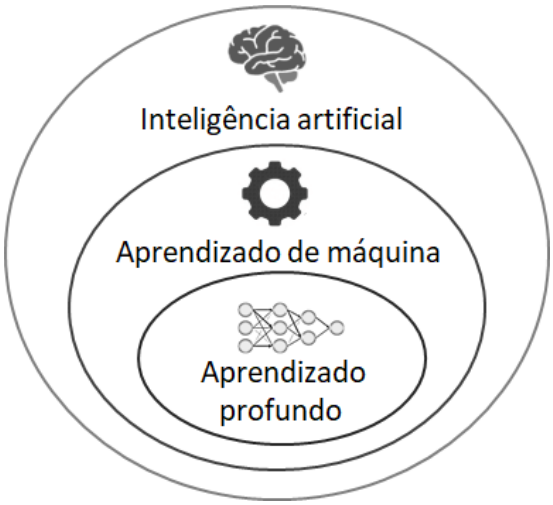


Figura 2.4: Relação entre inteligência artificial, aprendizado de máquina e aprendizado profundo. Adaptado de Chollet 2021 [29].

Inteligência artificial é uma parte das ciências da computação que, através de dispositivos eletrônicos ou sistemas informáticos, busca a automação de tarefas que normalmente são realizadas por humanos. Dentro da IA existe uma sub-área onde os algoritmos "aprendem", isto é, utilizam dados para aumentar a performance, ou eficácia, com a qual realizam alguma tarefa. Essa subárea é chamada de aprendizado de máquina (*Machine Learning*, ML), onde um sistema é treinado ao invés de ser programado explicitamente. A máquina analisa os inputs e procura um padrão de regras de modo a automatizar a tarefa [30].

Uma sub-área de ML é o aprendizado profundo (*Deep Learning*, DL). DL é um conjunto de algoritmos que fazem uso de redes neurais (com duas ou mais camadas ocultas)

para criar e processar dados de entrada e gerar saídas sem a necessidade de inserir regras manualmente [30, 31].

2.4 Redes neurais artificiais

As Redes Neurais Artificiais (Artificial neural networks, ANNs) são sistemas adaptativos inspirados nos processos de funcionamento do cérebro humano. As ANNs são compostas de nós interconectados que replicam o papel dos neurônios biológicos no envio de sinais de uns para os outros [32, 33].

Nas redes neurais, os nós são chamados de neurônios artificias ou simplesmente neurônios e cada um deles atua como uma unidade de processamento. A figura 2.5 mostra o modelo de funcionamento de um neurônio artificial. Nesse modelo, um ou mais sinais de entrada (x_i) chegam a cada neurônio e cada entrada tem um peso associado (W_i) que é equivalente a sua importância. Uma soma ponderada dos sinais é feita e então uma função de ativação (f) é aplicada para emitir um sinal de saída (y_i) para os outros neurônios. A função de ativação é uma função não linear que pode ser uma função degrau, sigmoide, Tanh, Rectified Linear Unit, etc. [33, 34]. As funções de ativação não lineares são usadas em redes neurais para permitir que o modelo aprenda relações não lineares entre as entradas e as saídas. Se a função de ativação fosse linear, a combinação linear de saídas de várias camadas intermediárias de neurônios também seria linear, o que limitaria a capacidade do modelo de aprender relações complexas e não lineares. Além disso, as funções lineares não podem modelar problemas não lineares, como XOR, e não podem capturar o comportamento não linear de muitos sistemas e processos do mundo real.

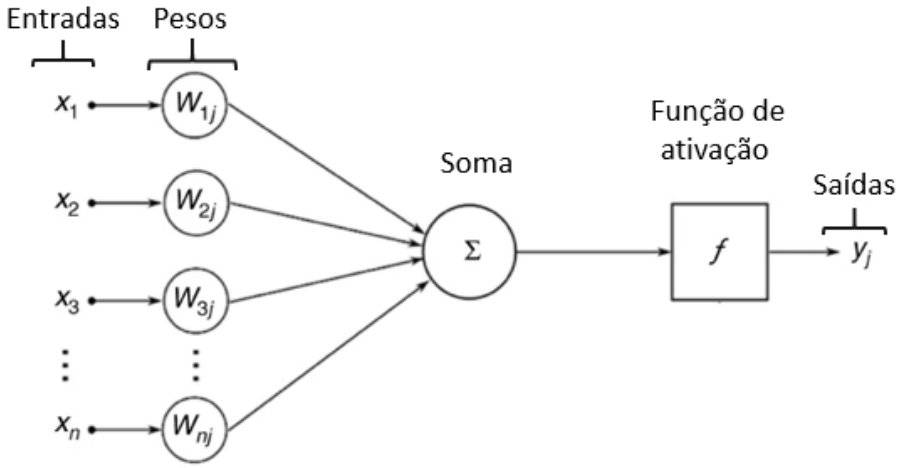


Figura 2.5: Diagrama esquemático de neurônio artificial. Adaptado de Zhang 2019 [34].

As redes neurais são sistemas capazes de ajustar sua estrutura interna, no caso os pesos de suas conexões, de modo a otimizar um resultado. São tipicamente utilizadas

para resolver problemas não lineares, capazes de reconstruir regras que governam a solução ótima para esses problemas [35]. A estrutura de uma rede neural geralmente está organizada por camadas: camada de entrada, camadas ocultas ou intermediárias e camada de saída. Cada uma delas contém neurônios que podem estar conetados aos neurônios das outras camadas (Fig. 2.6) [34, 35].

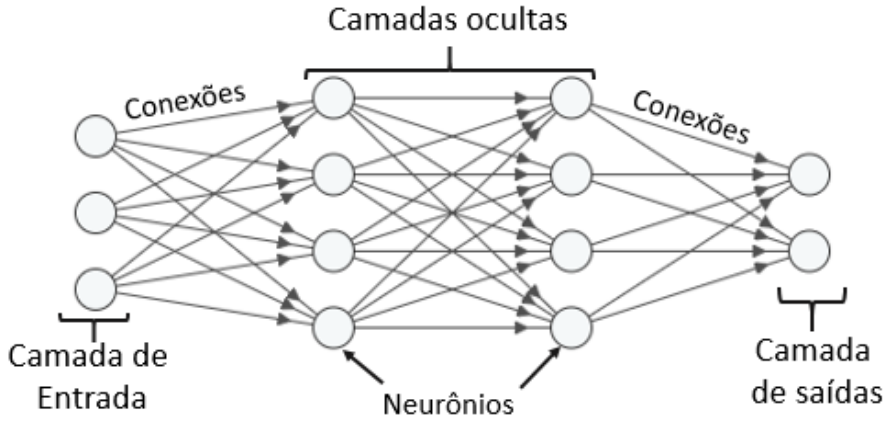


Figura 2.6: Esquema geral da estrutura de uma rede neural.

Um algoritmo de aprendizado de máquina é um algoritmo que é capaz de aprender a partir de dados, e podem ser classificados de acordo com o tipo de supervisão utilizada durante o treino. As quatro principais categorias são: aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi-supervisionado e aprendizado por reforço [36, 37].

Aprendizado supervisionado é um paradigma de aprendizado no qual apresentamos os dados de entrada e também os dados de saída. Aprendizado não-supervisionado só apresentamos os dados de entrada. Aprendizado semi-supervisionado lida com alguns dados de entrada, rotulados com a resposta de saída, e também alguns dados de entrada sem a saída. No aprendizado por reforço, o algoritmo enfrenta uma situação e procura a resposta para o problema através de tentativa e erro, por um sistema de recompensa.

2.4.1 Overfitting

O objetivo de um modelo de aprendizado supervisionado é fazer previsões corretas em dados novos, que não foram usados para treinamento. Dizemos que um modelo que faz as previsões corretas é capaz de generalizar as previsões, do conjunto de treino para o conjunto de teste e dados novos. Porém, se um modelo acerta no conjunto de treino, mas erra no conjunto de teste, ocorre o que chamamos de *overfitting*, ou, sobreajuste, onde o modelo não consegue generalizar as previsões. Alguns motivos são, quando utilizamos um modelo muito complexo, que contém mais parâmetros do que a quantidade de dados, ou quando ocorre um *data leakage* (vazamento de dados). *Data leakage* é uma situação

em que informações de teste ou validação são incorporadas de alguma forma ao modelo durante o treinamento. Isso pode acontecer, por exemplo, quando os dados de teste são usados acidentalmente para treinar o modelo ou quando informações que não estariam disponíveis no momento da previsão são usadas para treinar o modelo. [36, 38]

2.4.2 Hiperparâmetros de rede neural

Os processos de aprendizagem usam diferentes algoritmos para resolver as tarefas. Em algoritmos de aprendizado, certos parâmetros podem ser variados para otimizar o desempenho da tarefa. Esses parâmetros são chamados de hiperparâmetros e devem ser definidos antes do treinamento [36]. Alguns dos principais hiperparâmetros utilizados em redes neurais são: o número de camadas ocultas, o número de neurônios, a função de ativação, o tamanho do lote ou de *Batch*, as épocas (Epoch) e taxa de aprendizado (Learnig Rate) serão apresentados a seguir. A escolha de um hiperparâmetro adequado é importante, pois a eficiência ou o desempenho do algoritmo depende deles, por exemplo, para evitar o overfitting.

Número de camadas ocultas

O número de camadas ocultas de uma rede neural depende da complexidade do problema a ser resolvido. Para problemas simples como a aproximação de uma equação quadrática ou exponencial, pode ser usada uma camada. Já uma rede neural com duas camadas pode modelar até as funções mais complexas e terá um bom desempenho [39]. No entanto, se uma tarefa mais complexa como a classificação de imagens estiver envolvida, uma grande quantidade de dados será tratada e um grande número de camadas terá de ser utilizado [36, 39]. As redes neurais com mais de uma camada oculta são chamadas redes neurais profundas e são utilizadas em DL.

Função de ativação

A função de ativação que tem permitido um melhor treinamento em redes neurais profundas e que será utilizada neste projeto, é a função de unidade linear retificada (ReLU). A função é definida como:

$$f(x) = max(0, x). \tag{2.1}$$

A função ReLU retorna 0 se receber qualquer entrada negativa, e para qualquer valor positivo x ela retorna o próprio valor x. Na prática, funciona muito bem e é rápida de se calcular [36, 40].

Época (Epoch)

É o número de vezes que o algoritmo de aprendizado é executado. Em cada época ou ciclo, todos os dados de treinamento passam pela rede neural para ela aprender as características importantes sobre os dados. [37].

Tamanho de lote ou batch (Batch Size)

O tamanho de *batch* (lote) é o número de amostras que será propagada pela rede [37]. Então, vamos supor que temos 1.050 dados de treino e escolhemos um *batch*, ou *mini-batch*, de tamanho 100. O algoritmo vai usar as 100 primeiras amostras para treinar a rede. Em seguida, pega mais 100 amostras e assim por diante, sendo 11 lotes (10 com 100 e 1 com 50). As vantagens de se treinar em lotes ao invés de treinar na rede inteira é que consome menos memória, ainda mais se o *dataset* inteiro não couber na memória e também as redes são treinadas mais rapidamente usando lotes. A cada lote os pesos são modificados, então nesse exemplo atualizamos os parâmetros 11 vezes ao invés de uma, caso fosse treinado no *dataset* inteiro. Nesse exemplo, uma época está completa depois de treinar em todos os 11 lotes.

Função perda ou Loss (Loss Function) - Uma maneira de determinar se o algoritmo está funcionando corretamente é a *Loss Function*, ou função perda. Ela determina o quão próximo o output atual do algoritmo está do output esperado. Tal medida é usada como feedback para ajustar os parâmetros do algoritmo. Essa etapa de ajuste é o que chamamos de aprendizado [29]. Pode ser separado em dois grupos, um para classificação e outro para regressão, onde na classificação são usados valores discretos e na regressão são usados valores contínuos. Uma das funções *loss* mais utilizadas é a *cross-entropy*.

Cross-Entropy - Na Teoria da Informação, que é o estudo da quantificação, armazenagem e comunicação da informação, a função *cross-entropy* está relacionada com o cálculo de entropia e da diferença entre duas distribuições de probabilidades, a probabilidade verdadeira (p) e a probabilidade da previsão do modelo (q) (Eq. 2.2). A entropia quantifica a quantidade de incerteza envolvida no valor de uma variável ou processo aleatório. Pode ser interpretada também como informação que quantifica o número de bits necessários para codificar e transmitir um evento. Eventos de baixa probabilidade carregam mais informação, enquanto que eventos de alta probabilidade carregam menos informação. Por exemplo, identificar o resultado de jogar uma moeda (probabilidade de 1/2) nos dá menos informação do que identificar o resultado de um dado (probabilidade de 1/6).

$$H(p, q) = -\sum_{x} p(x) \log q(x) \tag{2.2}$$

Learning Rate

Learning rate (LR) ou taxa de aprendizado é um parâmetro que determina o tamanho do passo a cada iteração, conforme o algoritmo procura o mínimo da função Loss. De certa forma representa a velocidade com que o modelo aprende. É importante notar que se o LR for pequeno, o modelo demora muito para chegar no ponto ótimo, mas se for muito grande, pode falhar em convergir e acabar divergindo. [36]

2.4.3 Treinamento de redes neurais profundas

As redes neurais profundas, compostas por muitas camadas, são consideradas um dos modelos mais poderosos de aprendizado de máquina. No entanto, esses modelos podem apresentar desafios significativos durante o treinamento, como uma convergência lenta do processo e problemas de gradiente instável, em que os gradientes podem se tornar muito pequenos ou muito grandes em camadas profundas.

Para superar essas dificuldades, algoritmos de otimização foram desenvolvidos para acelerar o processo de treinamento e evitar mudanças abruptas no gradiente, permitindo que a rede neural atinja uma solução mais rapidamente e com maior precisão. Esses algoritmos incluem técnicas de descida de gradiente estocástica, como Momentum, RMSprop e Adam, que ajustam a taxa de aprendizado adaptativamente e lidam com gradientes instáveis, evitando que o processo de treinamento fique preso em mínimos locais.

Gradiente Descendente

O gradiente descendente é um algoritmo de otimização amplamente utilizado para treinar modelos de redes neurais [41]. A ideia do gradiente descendente é ajustar os parâmetros iterativamente de modo a minimizar a função custo [36, 41]. Este algoritmo permite atualizar os parâmetros na direção oposta do gradiente (inclinação ou derivada) da função de custo e um mínimo é encontrado quando o gradiente é zero ou bem próximo de zero [41].

O processo começa com um valor inicial aleatório dos parâmetros, que será gradualmente melhorado tomando pequenos passos para diminuir a função custo até o algoritmo convergir para o mínimo [36]. O tamanho dos passos é determinado pela taxa de aprendizado ou *learnig rate* e influencia diretamente no tempo de iteração e na divergência [36, 41, 42].

Existem diferentes tipos de algoritmos ou versões de aprendizado de gradiente descendente: gradiente descendente em lote ou batch, gradiente descendente estocástico e gradiente descendente em minilote [36, 41, 42].

Gradiente Descendente Isso é feito calculando o gradiente em relação a cada parâmetro para todo o conjunto de dados de treinamento; ou seja, todos os dados disponíveis são inseridos de uma só vez [41, 42]. Esse processo pode ser muito lento e inviável para conjuntos de dados que não cabem na memória, pois o gradiente sempre será calculado usando todas as amostras [41].

Gradiente Descendente Estocástico - consiste em escolher uma instância aleatória no conjunto de treinamento em cada etapa e computa os gradientes com base apenas nessa única instância [36, 42]. Isso torna o algoritmo mais rápido, pois tem poucos dados a serem manipulados em cada iteração. [42].

Gradiente Descendente em mini-lotes ou mini-batch - A cada iteração calcula os gradientes para pequenos subconjuntos com n amostras, subconjuntos selecionadas aleatoriamente e denominadas mini-lotes. Desta forma, obtém-se um algoritmo mais rápido que os anteriores [42, 41].

2.4.4 Algoritmos de otimização do gradiente descendente

O otimizador usado neste projeto é o otimizador Adam, que será mostrado a seguir, este é baseado na propagação de raiz quadrada ou *RMSProp* e na optimização de momento.

Adam - Adam é um algoritmo de otimização estocástico que é usado para adaptar a taxa de aprendizado de cada peso da rede neural [43, 44]. Seu nome é derivado da estimativa de momento adaptativo, pois esse método calcula taxas de aprendizado adaptativo individual para diferentes parâmetros a partir das estimativas do primeiro e segundo momentos dos gradientes [36, 43].

Adam é um dos algoritmos de otimização mais eficientes até hoje, pois possui poucos requisitos de memória e é adequado para problemas com grandes conjuntos de dados ou grande número de parâmetros [45]. Isso porque, para aumentar sua eficiência, Adam combina ideias da RMSProp e da otimização de momento [36, 44].

Este método calcula a taxa de aprendizado para cada parâmetro como no RMSProp, o que faz com que o algoritmo tenha um bom desempenho em problemas online e não estacionários [44], mas além disso mantém uma média exponencialmente decrescente dos gradientes pré-quadrados, semelhante à otimização de momentos [36, 44].

Propagação de raiz quadrada, RMSProp - Neste método a taxa de aprendizagem é adaptada para cada parâmetro. *RMSProp* inclui uma média móvel exponencial do

gradiente quadrado, ou seja, *RMSprop* divide a taxa de aprendizado por uma média exponencialmente decrescente de gradientes quadrados [41]. Portanto, a taxa de aprendizado não reduz agressivamente ao mínimo nas primeiras iterações [46].

Optimização de momento ou Momentum - *Momentum* é um método que ajuda a acelerar o gradiente descendente estocástico na direção relevante e amortece a oscilação que o gradiente apresenta quando se aproxima do vale e não alcança a convergência [41]. Para isso, essa otimização atualiza os parâmetros da rede adicionando um termo adicional, que está relacionado ao parâmetro de atualização anterior e é multiplicado por uma constante chamada coeficiente de momento. O termo de momento aumenta para dimensões cujos gradientes apontam nas mesmas direções e reduz atualizações para dimensões cujos gradientes mudam de direção, resultando em convergência mais rápida e oscilação reduzida [41, 46].

2.5 Redes neurais convolucionais - CNN

Redes neurais convolucionais ou redes convolucionais (Convolutional Neural Networks, CNN ou convnets) são um tipo especializado de redes neurais para processar dados que tem uma estrutura topológica do tipo grade (*grid-like topology*) [47, 37]. A topologia do tipo grade é um tipo de topologia de rede na qual cada nó da rede está conectado com dois nós vizinhos ao longo de uma ou mais dimensões [48]. Por exemplo, imagens podem ser pensadas como sendo uma grade 2D de pixels.

As redes neurais convolucionais consistem em várias camadas de filtros convolucionais de uma ou mais dimensões. Após cada camada, geralmente é adicionada uma função para realizar o mapeamento causal não linear [49].

Convolução é uma operação matemática, definida como

$$s(t) = \int x(a)w(t-a)da \tag{2.3}$$

ou, de maneira discreta,

$$s(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.4}$$

no contexto de convnets, a função x é chamado de input, a função w é chamado de kernel ou filtro e o resultado s é referido como mapa de atributos (*feature map*).

A principal diferença entre uma camada densamente conectada e uma camada convolucional é que as camadas densas extraem padrões globais no input, por exemplo,

analisando todos os pixels, enquanto que uma camada convolucional extrai padrões locais. No caso de imagens, padrões em pequenas janelas 2D dos inputs [37].

Como qualquer rede utilizada para classificação, no início essas redes possuem uma fase de extração de características, composta por neurônios convolucionais, depois há uma redução por amostragem e ao final teremos neurônios mais simples para realizar a classificação final sobre as características extraídas.

Os padrões que a rede aprende são invariantes na translação. Ao aprender um padrão em um local da imagem, uma convnet consegue identificar esse padrão em todo o restante da imagem. Convnets aprendem padrões espaciais hierárquicos. A primeira camada convolucional aprende padrões locais, como bordas. Uma segunda camada aprende padrões baseados nesses primeiros, e assim por diante, cada camada sendo mais complexa que a anterior [50].

Convoluções operam em tensores de ordem 3, com as dimensões sendo a altura (A), largura (L) e profundidade (P), que, no caso de imagens, representa o número de canais [49]. Para uma imagem em tons de cinza, a profundidade é 1, para imagens RGB (Red, Green, Blue) a profundidade é 3. No presente trabalho em um dos estudos adicionamos mais dois canais, portanto a profundidade é 5.

Uma CNN é normalmente composta de três tipos de camadas: convolução, *pooling* e camadas totalmente conectadas. Os dois primeiros realizam a extração de características, enquanto o terceiro mapeia as características extraídas no resultado final, como a classificação [50, 51].

2.5.1 Camada de convolução

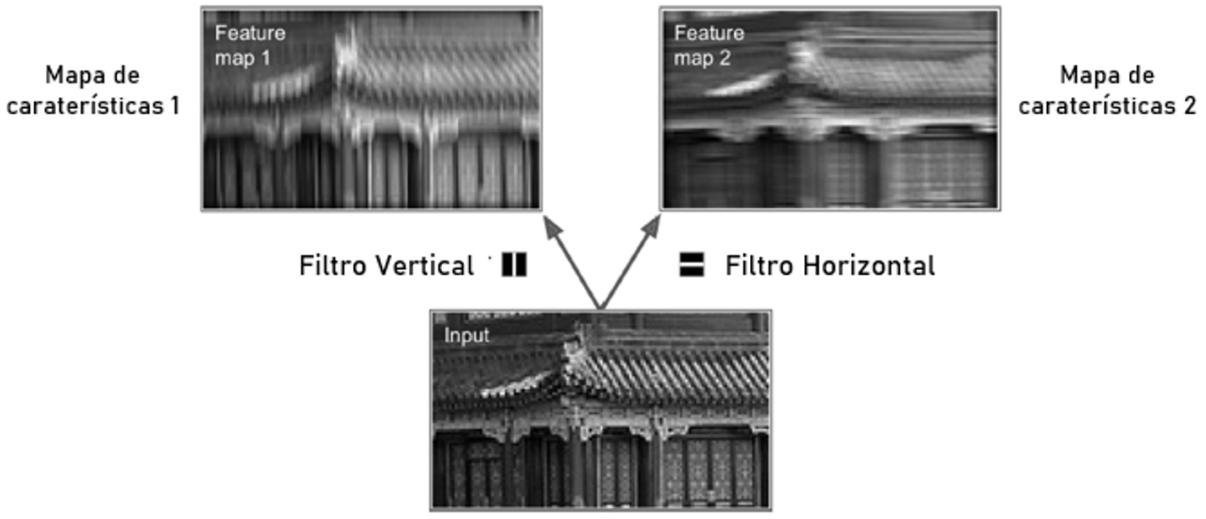


Figura 2.7: Dois filtros diferentes geram mapas de características diferentes. Adaptado de Aurélien [36].

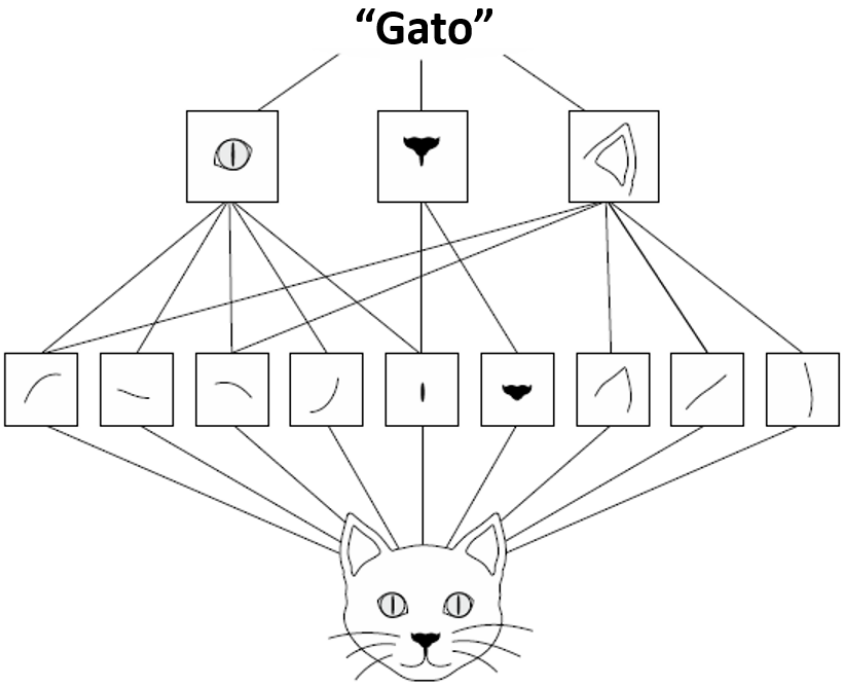


Figura 2.8: Linhas e texturas simples se combinam para formar objetos complexos, como olhos e orelhas e estes se combinam para formar objetos ainda mais complexos, como um gato. Adaptado de Chollet [29].

A camada de convolução extrai pedaços do mapa de atributos e aplica uma transformação, gerando um output de mapa de atributos. Esse output ainda é um tensor de ordem 3, a altura e a largura podem variar um pouco do tamanho original e a profundidade agora é a quantidade de filtros que a camada gera. Para que a altura e a largura fiquem do mesmo tamanho, coloca-se zeros ao redor do input, isso é chamado de preenchimento de zeros (*zero padding*) [51].

Em redes tradicionais, cada unidade de output interage com cada input, mas convnets tem interações esparsas, ao fazer os kernels (filtros) serem menores que os inputs. Por exemplo, uma imagem com centenas de milhares de pixels, podemos fazer um kernel que identifica bordas com apenas dezenas de pixels. Dessa forma usamos menos parâmetros, o que reduz a quantidade de memória utilizada e torna os cálculos mais rápidos [50, 49].

Os filtros, ou kernels de convolução, não são escolhidos manualmente. O processo de treinamento de uma CNN em relação à camada de convolução é identificar os kernels mais úteis para uma determinada tarefa com base em um determinado conjunto de dados de treinamento. O tamanho dos kernels, número de kernels e *padding* são hiperparâmetros que precisam ser definidos antes do início do processo de treinamento [50].

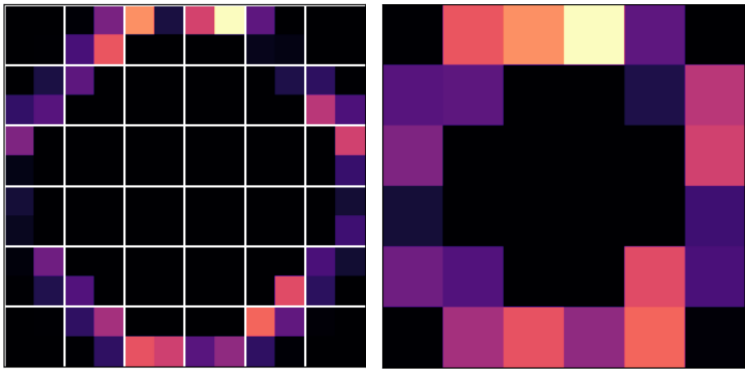
Depois, as saídas de uma operação linear, como a convolução, são passadas por uma função de ativação não linear. No nosso caso, a função não linear utilizada é a função ReLU 2.4.2.

2.5.2 Pooling

Um dos problemas das convnets é que é necessário muita memória. O objetivo da camada de pooling é diminuir o tamanho da imagem, reduzindo assim a carga computacional, uso de memória e número de parâmetros, o que também diminui o risco de *overfitting* [51].

A forma mais popular de operação de *pooling* é *max pooling*, ela particiona a imagem em sub-regiões de quadrados e retorna apenas o valor máximo do interior dessa sub-região [49, 52]. Normalmente a camada de *pooling* é aplicada independentemente em cada um dos canais, então a profundidade do output é a mesma que o input.

Pooling ajuda a representação ser aproximadamente invariante a pequenas translações do input [50]. Isto é, se fizermos uma pequena translação no input, o output muda pouco. É uma propriedade útil se queremos saber se um objeto está presente mas não tanto exatamente onde. Se a rede identificar um objeto no canto superior direito da imagem, por exemplo, a rede consegue identificar esse mesmo objeto em outros cantos da imagem.



(a) Original **(b) Max pool**

Figura 2.9: Exemplo de max pool, separamos a imagem em pedaços 2x2 e a saída é o maior valor dentro de cada um dos pedaços. Note que diminuímos em 75% as dimensões da imagem, mas a estrutura, o formato da imagem se manteve o mesmo. Adaptado de [53].

2.5.3 Camada totalmente conectada

Os mapas de atributos de saída final são geralmente transformados em uma matriz de números unidimensional (1D) e conectados a uma ou mais camadas totalmente conectadas, também conhecidas como camadas densas, onde cada entrada é conectada a cada saída por um peso [50]. A camada totalmente conectada contém neurônios que estão diretamente conectados a neurônios em camadas adjacentes, de forma análoga às formas tradicionais de redes neurais feedforward [51].

2.6 Dropout

Dropout é uma técnica de regularização em *Deep Learning*, proposto por Geoff Hinton [54] e melhorada por Nitish Srivastava et al. [55], que tem como objetivo prevenir *overfitting* em modelos de aprendizado profundo. O *dropout* funciona "desligando"aleatoriamente uma fração dos neurônios em cada camada do modelo durante cada iteração de treinamento. Isso evita que os neurônios fiquem muito dependentes uns dos outros e forçá-los a aprender características mais robustas dos dados. Durante a avaliação, todos os neurônios são mantidos ativos e as saídas são combinadas através de médias ponderadas.

A taxa de *dropout* é geralmente escolhida por experimentação e validação cruzada e pode variar de acordo com o tipo de problema e o modelo em questão. Valores típicos variam de 0,1 a 0,5, o que significa que entre 10% e 50% dos neurônios são desligados em cada iteração de treinamento.

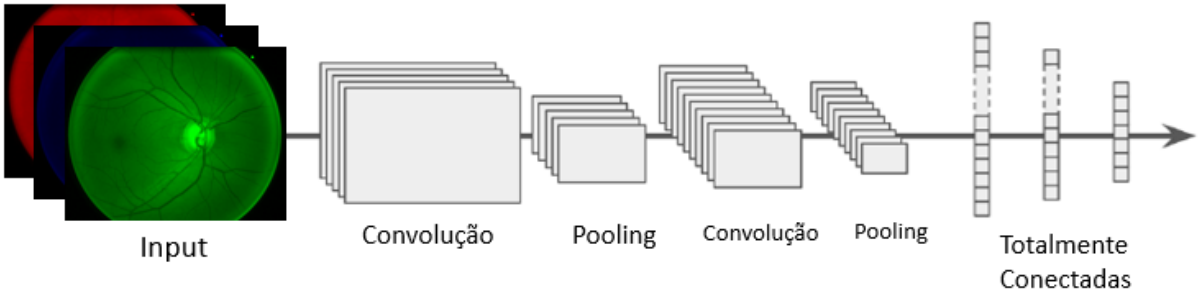


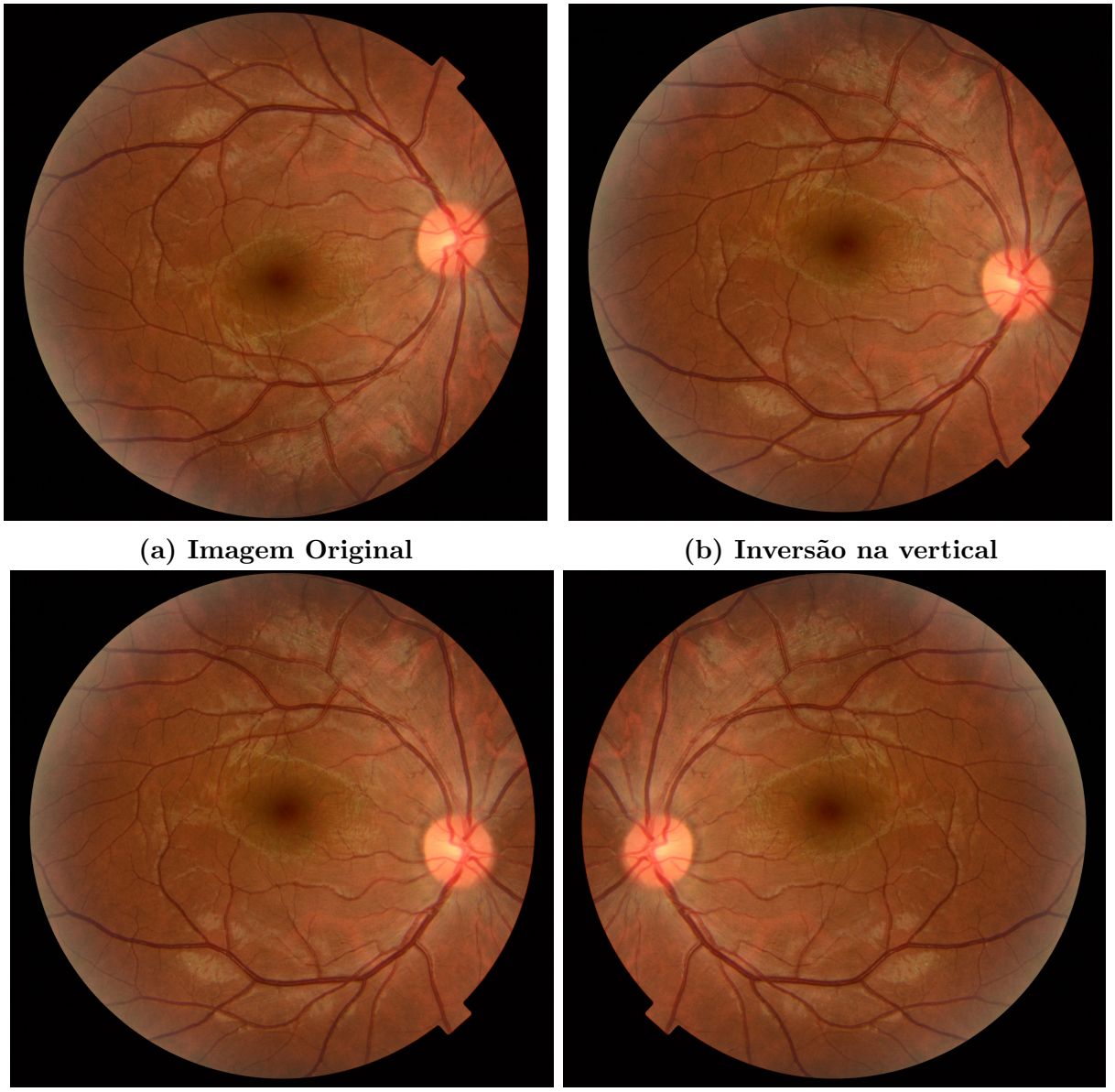
Figura 2.10: Arquitetura típica da rede neural convolucional. A primeira camada são os inputs, no nosso caso uma imagem RGB, seguido de conjuntos de camadas de convolução e *pooling*, finalizando com camadas totalmente conectadas. Adaptado de Aurélien [36]

2.6.1 Data Augmentation

O *data augmentation* (aumento de dados) aumenta o tamanho do conjunto de treino ao gerar variantes de cada instância de treino, como girar, transladar e mudar o tamanho das imagens. Isso ajuda a diminuir o sobreajuste, uma vez que o modelo é forçado a ser mais robusto para pequenas variações nas imagens de treino [36].

2.6.2 Aplicações de CNN

Algumas análises médicas, como radiografias, tomografias, ressonâncias magnéticas, retinografias, ultrassons, etc., geram imagens que precisam ser analisadas por um médico para fazer um diagnóstico. O objetivo da análise de imagens médicas é extrair informações



(a) Imagem Original (b) Inversão na vertical

(c) Inversão na horizontal (d) Inversão na vertical e horizontal

Figura 2.11: Possíveis formas de *data augmentation*: Inversão na vertical, horizontal, ou ambas.

de forma eficaz e eficiente usando técnicas computacionais para melhorar o diagnóstico clínico.

Entre as técnicas de DL, redes convolucionais profundas ou CNNs são usadas ativamente para tal análise [56]. Por exemplo, elas têm sido usados em estudos de segmentação, detecção de anormalidades como tumores, classificação de doenças como câncer de pele [57] e diagnóstico de pneumonia em tomografia computadorizada de tórax [58]. Em oftalmologia as principais linhas de pesquisa estão associadas com retinopatia diabética [59], degeneração macular relacionada à idade [60], catarata [61] e glaucoma [62]. O uso de *deep learning* em glaucoma chegou em altos níveis de sensitividade, 97.60% e 85% de especificidade em um desafio internacional [63]. Para esses diferentes estudos tem sido usado fotos coloridas de fundo de olho e, em menor quantidade, scans de tomografia de coerência óptica (OCT).

Atualmente, as CNNs também têm sido usadas no diagnóstico do COVID-19. O trabalho de *T. Goel* (2020) [64] propõe uma rede neural convolucional otimizada (OptCoNet) composta por otimizadores de classificação e extração de recursos para o diagnóstico automático de COVID-19 a partir de imagens de radiografia de tórax. Outro estudo usou dez CNNs para classificar pacientes com COVID-19 e pacientes com pneumonia [65], onde um dos tipos de rede utilizados foi a rede Xception, que também será a utilizada neste trabalho. Pelo que descrevemos, DL e CNNs possuem uma ampla gama de aplicações na área da medicina, que são aperfeiçoadas ao longo do tempo e facilitarão muito o trabalho dos médicos.

2.6.3 Transferência de aprendizado e redes pré treinadas

Para poder treinar CNNs e aplicá-las a problemas mais reais, é necessário ter uma grande quantidade de dados, o que é um problema devido à dificuldade de obtenção de tais quantidades [66]. Uma possível solução ou forma de melhorar o resultado é fazer uso de uma estratégia chamada Transferência de Aprendizado ou Transfer Learning (TL) que é utilizada para reduzir o tamanho dos dados de treinamento. A transferência de aprendizado consiste em usar uma CNN pré-treinada com uma grande quantidade de dados utilizada para resolver uma tarefa relacionada e usar esse conhecimento para resolver uma nova tarefa que tem poucos dados de treinamento [66, 67].

Por exemplo, modelos treinados em imagens aprendem características similares como linhas, bordas, cantos, gradientes, formas, entre outros...,a partir de diferentes conjuntos de dados de imagens, de modo que estas características podem ser reutilizadas para resolver outras tarefas de reconhecimento ou classificação de imagens [66]. Por exemplo, a rede CheXNet treinada em 112.000 imagens de Raios-X de tórax foi usada para melhorar o aprendizado de uma CNN aplicada a um conjunto de dados de mamografia, que tem uma

quantidade menor de dados, 10.420 imagens [68]; também, a transferência de aprendizado têm sido usada na classificação de histopatologias onde é feita uma comparação usando redes com e sem TL [69]. Em ambos os casos, observou-se uma melhora na classificação das imagens com o uso de TL.

Existem diversas arquiteturas ou redes pré-treinadas como por exemplo: Xception, VGG-19, ResNet, InceptionV3, MobileNet, DenseNet, NASNet [67, 69], entre outras. No presente trabalho utilizamos a rede Xception.

Xception

Xception é uma rede neural proposta por François Chollet em 2017 enquanto trabalhava para a Google [70]. Xception é inspirado na rede *Inception*, que consiste em módulos de *inception* que calculam várias transformações diferentes na mesma entrada e, por fim, vinculam seus resultados para gerar a saída [70, 71].

No Xception, cada módulo *inception* é substituído por convoluções separáveis em profundidade. Este operador realiza uma convolução separada em cada canal de entrada e, em seguida, aplica uma convolução 1×1 para projetar os canais de saída gerados em um novo espaço de canal [36, 70].

Na escolha da rede neural para o problema de classificação de imagens, a rede Xception foi selecionada devido às suas performance satisfatórias em comparação com outras redes pré-treinadas. No entanto, trabalhos futuros podem utilizar outras redes pré-treinadas para avaliar a efetividade de cada rede.

2.7 Métricas

As métricas de avaliação refletem a qualidade de um modelo e devem ser escolhidas de acordo com o tipo de modelo, pois a avaliação depende da tarefa a ser realizada. Por exemplo, para a tarefa de classificação, o modelo é avaliado medindo a taxa na qual uma categoria prevista corresponde à categoria real. E, para o agrupamento, a avaliação se baseia no grau de proximidade entre os itens agrupados e o grau de separação entre os agrupamentos [72].

Em nosso caso, serão usadas diferentes métricas para avaliar a classificação. Uma forma de visualizar o desempenho de um modelo de classificação é através de uma matriz de confusão [36, 73]. A matriz de confusão permite visualizar facilmente quantos exemplos foram classificados corretamente e erroneamente em cada grupo: falso positivo (FP, false positive), falso negativo (FN, false negative), verdadeiro positivo (TP, true positive) e verdadeiro negativo (TN, true negative), como pode ser visto na Figura 2.12. Mas esses

valores não devem ser avaliados de forma independente, mas sim em conjunto com os demais.

		PREDITO	
		Positivo	**Negativo**
REAL	**Positivo**	**TP** Verdadeiro Positivo	**FN** Falso Negativo
	Negativo	**FP** Falso Positivo	**TN** Verdadeiro Negativo

Figura 2.12: Matriz de confusões mostrando os diferentes grupos ou classes, TP, TN, FP e FN. Adaptado de [36]

Relacionando os diferentes grupos ou classes, podemos definir várias métricas. Há nomes variados para representar as mesmas abstrações, portanto vamos explicitar algumas delas e definir quais serão usadas neste trabalho.

Sensibilidade ou Taxa de verdadeiro positivo (TPR): Mede a proporção de positivos reais que são corretamente identificados como positivos [36, 72].

$$TPR = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.5}$$

Especificidade ou Taxa de verdadeiro negativo (TNR): Mede a proporção de negativos reais que são corretamente identificado negativos [36, 72].

$$TNR = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{2.6}$$

Precisão ou Valor preditivo positivo (PPV): A proporção de observações positivas que são verdadeiros positivos [36, 72].

$$PPV = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.7}$$

Valor preditivo negativo (NPV): A proporção de observações negativas que são verdadeiros positivos [36, 72].

$$NPV = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} \tag{2.8}$$

Acurácia (ACC): Avalia a proporção do número de previsões corretas para o número total de amostras. Acurácia não é uma boa métrica para dados muito desbalanceados

[36, 72]. Suponha uma amostra com 95 casos positivos e 5 negativos. Um modelo pode classificar todos os casos como positivos e terá uma acurácia de 0,95.

$$ACC = \frac{\text{TP+TN}}{\text{TN + TP + FN + FP}} \tag{2.9}$$

Curva Característica de Operação do Receptor (Curva ROC): ou, do inglês Receiver Operating Characteristic Curve (ROC curve), é um gráfico que representa o desempenho de um classificador binário [36, 72]. A curva ROC é criada ao plotar a TPR contra a TNR em várias configurações de limiar. Cada ponto da curva representa um par sensitividade/especificidade correspondente a um limiar.

A curva ROC é uma ferramenta que nos ajuda a selecionar o melhor modelo para um dado problema. Pode-se medir a performance de um teste ou a acurácia de um teste para distinguir casos de doenças dos casos normais [74, 75]. Pode ser também usada para comprar a performance de dois ou mais testes de laboratório [76].

Um teste com uma distinção perfeita tem uma curva que passa sobre o canto superior esquerdo do gráfico, alta especificidade e alta sensibilidade. Portanto, quanto mais próxima do canto superior esquerdo melhor a acurácia geral do teste.

Área Sob a Curva ROC (AUC): É uma medida de quão bem um parâmetro consegue distinguir entre dois grupos [36]. AUC é robusto em relação a classes desbalanceadas. Por exemplo, caso um classificador retorne apenas uma classe, a AUC será 0,5, enquanto que acurácia seria um valor alto.

Nesse trabalho serão utilizadas as métricas acurácia (ACC) e a área sob a curva ROC (AUC) para analisar o desempenho dos modelos utilizados.

2.8 Objetivos

O propósito da presente pesquisa de mestrado é investigar o impacto da adição de informações físicas, em forma de cromóforos, sobre o desempenho de redes neurais convolucionais na tarefa de classificação da presença de doenças oculares, particularmente no que se refere à acurácia e AUC.

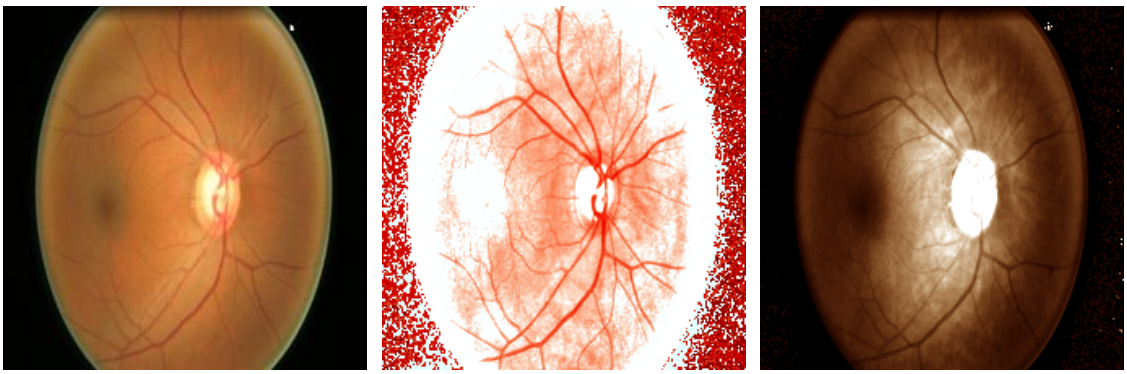
Além disso, objetiva-se examinar a influência da quantidade de dados disponíveis sobre os modelos e métricas escolhidos. Embora seja sabido que a inclusão de um volume maior de dados pode melhorar os resultados, há situações em que a disponibilidade de instâncias é limitada. Nesse contexto, busca-se averiguar se o pré-processamento de melanina e oxiemoglobina pode aprimorar a capacidade de generalização das redes neurais convolucionais.

Neste trabalho queremos responder as seguintes perguntas:

- Como se comportam as curvas de treino?
- Em uma rede pré-treinada, é melhor treinar apenas as camadas modificadas ou todas as camadas?
- Para as imagens RGB, quais dos modelos é melhor, um pequeno, um médio ou um grande?
- Fazer o pré-processamento da melanina e oxiemoglobina e utilizar esses mapas como canais das imagens, os resultados são melhores do que utilizar apenas o RGB?

3 Metodologia

O método proposto consiste em estudar diferentes imagens de fundo de olho, com e sem a presença de patologias. Faremos isso em dois passos: o primeiro é rodar um algoritmo de separação de melanina e hemoglobina das imagens para obter as intensidades dos cromóforos. O segundo passo é treinar duas redes neurais, uma com os dados originais e outra com as concentrações dos cromóforos, para então comparar os resultados. Juntamos as imagens de forma que os três primeiros canais fossem RGB e os outros dois HbO2 e melanina, essa nova "imagem"será referida como "5 canais". Um cromóforo é a parte da molécula que é responsável por sua cor. O algoritmo em questão foi aplicado no estudo de [77] para quantificar a concentração dos cromóforos principais da pele, incluindo a hemoglobina oxigenada (HbO2), hemoglobina desoxigenada e melanina Figura 3.1. Neste trabalho, vamos utilizar somente as concentrações de HbO2 e melanina.



(a) RGB **(b) HbO2** **(c) Melanina**

Figura 3.1: Um olho normal em RGB e as concentrações de seus cromóforos: b) HbO2 e c) melanina.

Podemos notar na figura que a extração não é perfeita, há alguns artefatos, como por exemplo as bordas da hemoglobina oxigenada estão marcadas com vermelho, embora não exista hemoglobina nessas áreas. Portanto, embora não seja um bom resultado para se quantificar a quantidade de hemoglobina no olho, ainda podemos usar como auxílio para redes neurais, uma vez que todas as imagens de HbO2 estarão com o mesmo artefato.

3.1 Separação dos cromóforos

O método de separação dos cromóforos está mais detalhado no Apêndice A.

3.2 Software utilizado

Foi utilizado a linguagem de programação Python, com a biblioteca Tensorflow em conjunto com o Keras.

Foram utilizados duas plataformas para rodar os códigos, o Google Colab e o Kaggle. Google Colab é uma ferramenta gratuita do Google que permite criar e executar códigos em Python em um ambiente de Jupyter Notebook no navegador. Ele tem recursos de processamento em nuvem, GPU e TPU gratuitos, o que torna mais fácil para os usuários treinar modelos de aprendizado de máquina sem se preocupar com a configuração do hardware. Além disso, o Colab permite a colaboração em tempo real, permitindo que múltiplos usuários editem um mesmo notebook ao mesmo tempo. Kaggle é uma plataforma de aprendizado de máquina e data science, propriedade do Google, que fornece recursos de competições, *datasets* e ferramentas de desenvolvimento para os usuários. É uma comunidade global de cientistas de dados, desenvolvedores e analistas que trabalham juntos para solucionar os maiores desafios de aprendizado de máquina. No Kaggle, os usuários podem participar de competições de aprendizado de máquina para resolver problemas reais de negócios, explorar grandes conjuntos de dados, treinar e avaliar seus modelos em um ambiente seguro, e aprimorar suas habilidades em ciência de dados. Além disso, o Kaggle também oferece recursos de treinamento e educação, incluindo tutoriais interativos, cursos e palestras. O Google Colab usa as GPUs Nvidia Tesla K80, Nvidia T4, Nvidia P100 e Nvidia V100, mas não se pode escolher qual, depende da quantidade de pessoas usando a plataforma e se o plano é gratuito ou pago [78]. A Kaggle utiliza a GPU NVidia K80 .

As versões do Python, Tensorflow e Keras são 3.8, 2.9.2 e 2.9.0 no Colab e 3.7, 2.6.4 e 2.6.0 no Kaggle.

3.3 Descrição dos datasets

Os dados foram obtidos de dois *datasets* públicos, o "Glaucoma Detection"com 650 imagens de olhos com glaucoma e normais (saudáveis) [79] (sendo 482 normais e 168 com glaucoma) e "1000 Fundus images with 39 categories", com 1000 imagens de fundo de olho, sendo 38 olhos normais e 982 olhos com diversos tipos de patologias diferentes [21]. No total temos 1650 imagens, sendo 520 de olhos normais e 1130 de olhos com alguma patologia, uma proporção de 1:2,17.

O *dataset* foi dividido em três conjuntos: treino, validação e teste. O conjunto de treino é utilizado para treinar as redes, onde elas procuram os melhores pesos e filtros. O conjunto de validação é usado durante o ajuste do modelo para avaliar a função *loss* e quaisquer outras métricas de interesse, para avaliar o quão bem o modelo está generalizando para dados não vistos. O conjunto de teste não é usado durante a fase de treinamento, só é usado no final para avaliar o quão bem o modelo generaliza para novos dados. O número total de imagens é 1650 e foi separado 80% para treino e 20% para teste. Do conjunto de treino, 20% foi separado para a validação, de forma que a proporção final foi de 64% das imagens para treino, 16% para validação e 20% para teste (Fig. 3.2). A

divisão foi feita de maneira estratificada, isto é, a proporção entre controle e patologia foi mantida nos conjuntos de treino, teste e validação.

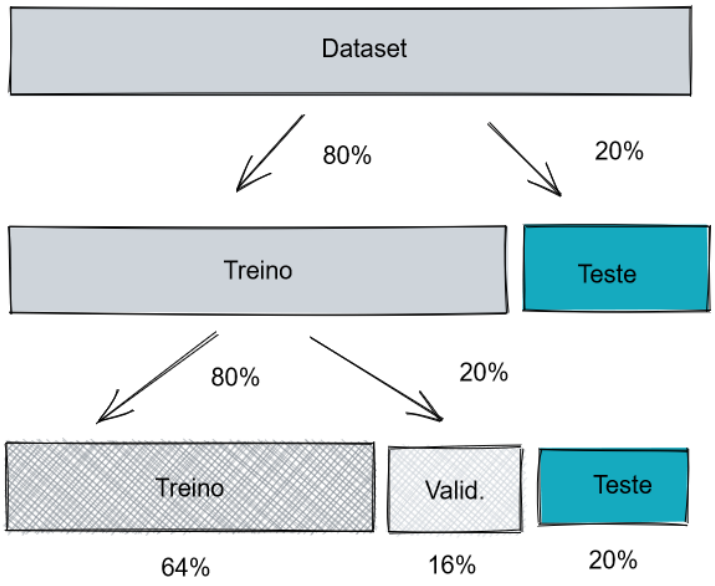
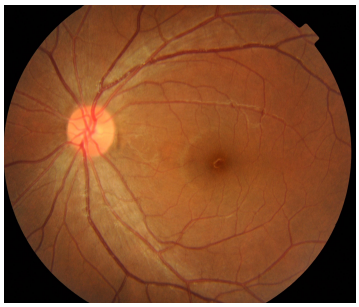


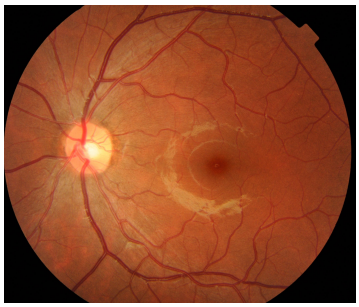
Figura 3.2: Divisão do conjunto de dados em treino, teste e validação.

3.4 Pré-processamento



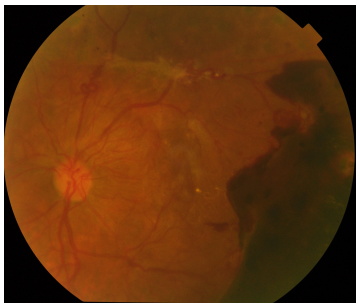
(a) Olho Normal

(b) Olho Normal

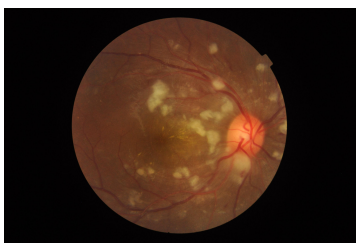


(c) Olho Normal

(d) Glaucoma



(e) Retinopatia diabética



(f) Retinopatia hipertensiva

Figura 3.3: Exemplos de imagens de retinografia dos datasets, de diversos pacientes, sendo (a),(b) e (c) de olhos normais e (d) (e) e (f) de olhos com alguma doença.

Como dito anteriormente, o primeiro passo é utilizar o algoritmo para isolar as componentes de hemoglobina e melanina das imagens.

Essas redes tem um tamanho de input fixo esperado, por exemplo 224x224x3 [80, 81], portanto redimensionamos o tamanho das imagens, que no geral tem um tamanho de 3000x3000 para 224x224. Um exemplo de algumas imagens utilizadas durante o treinamento pode ser visto na Figura 3.3. Outro motivo para se diminuir o tamanho das imagens é poder ter um treino mais eficiente: treinar uma rede neural com imagens grandes requer muitos recursos computacionais, incluindo tempo de processamento e memória. Usar imagens menores ajuda a reduzir esse custo e também aumenta a velocidade de treinamento. Treinar uma rede com imagens menores ajuda a prevenir o *overfitting*, uma vez que essas imagens possuem menos informação do que imagens maiores. Isso significa que as redes neurais precisam aprender representações mais genéricas e abstraídas dos dados, ao invés de se ajustar excessivamente aos dados de treinamento. Isso aumenta a capacidade da rede de generalizar para dados desconhecidos, reduzindo assim o risco de *overfitting*.

Outra maneira de diminuir o *overfitting* e aumentar a capacidade das redes de generalizarem o treinamento, é o chamado *data augmentation*, onde pequenas transformações são feitas nas imagens de modo a gerar instâncias sempre um pouco diferentes para as redes. No presente trabalho aplicamos duas transformações: espelhamos as imagens na vertical e na horizontal, como pode ser visto na Figura 2.11.

3.5 Arquitetura dos modelos

Temos 3 modelos, o Baseline, Principal e Xception. Os nomes escolhidos para identificar os modelos, isto é, Baseline e Principal, são meramente designativos e não se baseiam em motivos específicos descritos na literatura científica.

As arquiteturas dos modelos são as seguintes: temos o modelo Baseline, cujo objetivo é ter uma rede de base para poder comparar com a eficácia de redes mais complexas. Essa rede é composta por uma camada de convolução com 8 filtros, com um kernel de tamanho 3, seguido de 3 camadas de *pooling* (de tamanho 2x2), o que diminui o tamanho das imagens e agiliza o processo de treino representado na figura 3.4. O segundo modelo, que chamamos de Principal, possui 3 camadas de convolução seguidas de *pooling* e termina com duas camadas densas e de *dropout*, representado na figura 3.5. Mais especificamente, o modelo Principal tem a seguinte estrutura: uma camada convolucional com 32 filtros de kernel 7, uma camada *maxpool*, uma camada convolucional com 128 filtros de kernel 3 seguida de uma *maxpool*, uma camada convolucional com 256 filtros de kernel 3 seguida de uma *maxpool*, uma camada densa com 128 neurônios seguida de *dropout*, uma camada densa com 64 neurônios seguida de *dropout* e finalmente uma camada densa com

2 neurônios para se fazer a classificação binária.

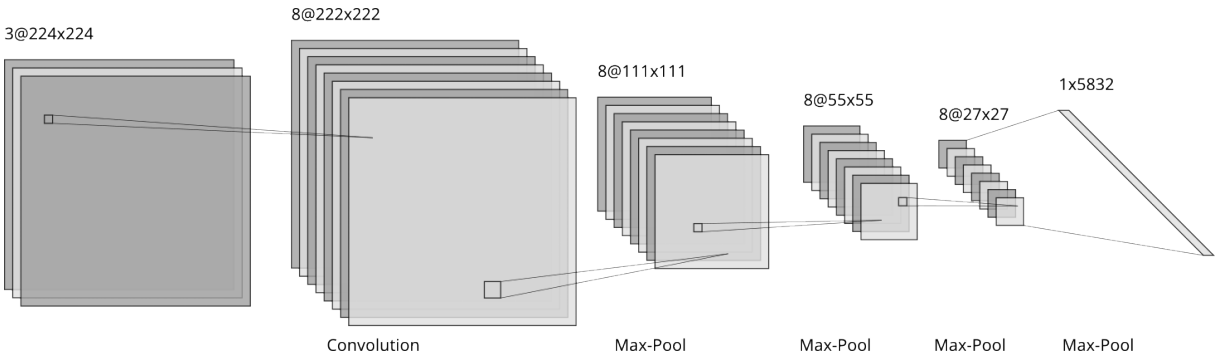


Figura 3.4: Arquitetura da rede Baseline. Começa com uma convolução e finaliza com 3 camadas max-pool.

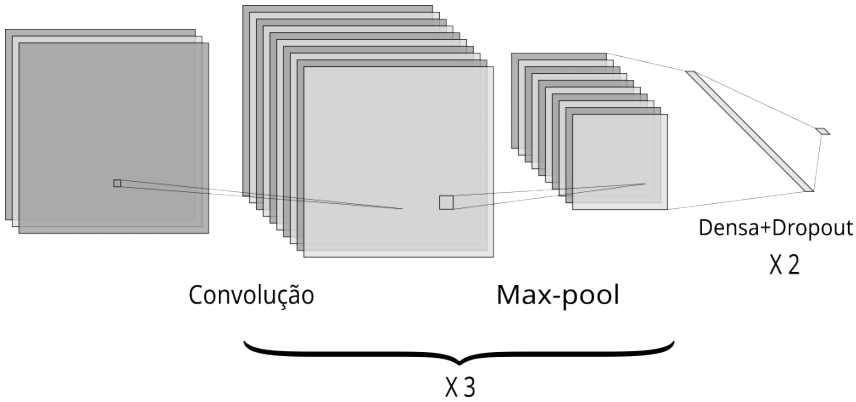
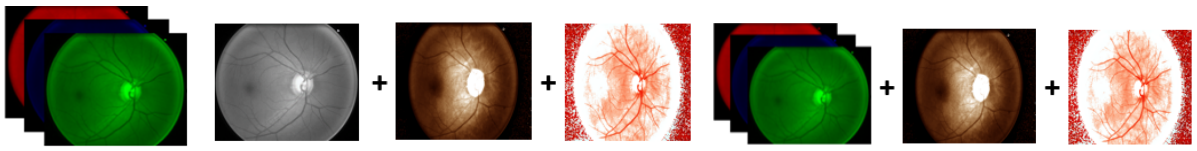


Figura 3.5: Arquitetura da rede principal. Começa com três camadas de convolução seguidas de max-pool e termina com duas camadas densas seguidas de *dropout*.

E finalmente utilizamos um modelo pré-treinado, em imagens RGB, Xception [70]. Não utilizamos a Xception para as imagens de 5 canais uma vez que é um modelo pré-treinado e os inputs são fixos para imagens de 3 canais, como as RGB.

3.6 Treinando as redes neurais

Os parâmetros escolhidos para treinamento foram: um tamanho de *mini-batch* de 32, o otimizador ADAM com uma taxa de aprendizado de 10^{-4}, 1000 épocas, mas com um *early stopping* de 5, ou seja, se a rede não melhorar os valores das métricas escolhidas em 5 épocas, ela para de treinar e utiliza os pesos do melhor resultado. Também configuramos o *early stopping* para uma variação mínima entre o valor de uma métrica em uma época e outra em 10^{-3}, uma vez que a métrica pode melhorar, mas muito pouco. Para o *early stopping* a métrica escolhida foi a acurácia de validação e a função *loss* foi a entropia cruzada.



(a) RGB **(b) 3ch** **(c) 5ch**

Figura 3.6: Exemplo dos conjuntos utilizados. (a) RGB, (b) escala de cinza, HbO2 e Melanina e (c) RGB, HbO2 e Melanina.

3.7 Nomenclatura utilizada

Para testar a eficácia de se utilizar os mapas de HbO2 e melanina, comparamos a performance das redes em 3 conjuntos diferentes, que chamamos de RGB, 3 canais (3ch) e 5 canais (5ch). RGB são as imagens sem pré-processamento (isto é, sem utilizar os mapas), 3ch utilizamos

3.8 Número de repetições

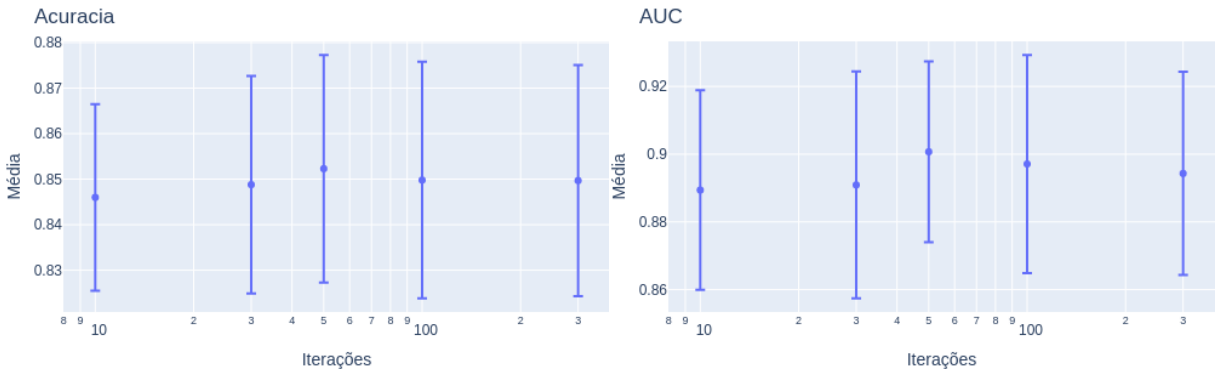


Figura 3.7: Comparação das médias da quantidade de repetições da acurácia e da AUC. Vemos que não há muita diferença entre as médias, portanto os valores são consistentes.

Os pesos das redes, as conexões entre uma camada e outra, são iniciados de maneira aleatória, portanto nem sempre a rede encontra o mínimo global, as vezes o algoritmo fica preso em um mínimo local. Portanto há uma variação nos valores finais. Para determinar o número de repetições no treinamento da rede que fornece um valor estável das métricas de acurácia e AUC treinamos o mesmo modelo várias vezes (10, 30, 50, 100 e 300 repetições) com o modelo principal, canais RGB e 60% dos dados e plotamos as médias e desvios padrão no gráfico. Podemos notar na Figura 3.7 que a diferença é estatisticamente insignificante para a acurácia e AUC entre 30 e 300 repetições, portanto para os próximos gráficos/modelos vamos utilizar 30 repetições do treinamento. Disponibilizamos os códigos em https://github.com/Andre-Vitor/Codigo-Mestrado.

3.9 Limitações

Este trabalho apresenta algumas limitações que devem ser destacadas. Primeiramente, é importante observar que o tamanho de imagem empregado foi de 224x224, o que pode acarretar em perda de informação. Entretanto, não foram realizados experimentos para avaliar tamanhos alternativos de imagem. Outro aspecto relevante é que todas as informações contidas nos mapas de oxiemoglobina e melanina já estão presentes nas imagens RGB utilizadas. Além disso, nem todas as doenças oculares estão relacionadas com a quantidade de hemoglobina presente no olho. Também, não foi realizado um ajuste fino dos hiperparâmetros para as imagens RGB e nem para as imagens submetidas ao pré-processamento de cromóforos. Por fim, é importante mencionar que as imagens foram obtidas a partir de diferentes retinógrafos. A separação entre melanina e HbO2 é influenciada pelas condições da câmera e iluminação utilizadas, o que sugere que a utilização de um único retinógrafo poderia fornecer resultados mais confiáveis.

4 Resultados e Discussão

A cada época de treino, isto é, a cada vez que a rede passa por todos os dados, calculam-se métricas para sabermos como está o andamento do aprendizado. O gráfico da métrica (no nosso caso da acurácia e da *loss*) é chamado de curva de aprendizado. Nas figuras de curva de aprendizado, por exemplo Fig. 4.1, o eixo X representa a época de treinamento e o eixo Y indica o valor da acurácia Fig 4.1 (a) ou *loss* Fig 4.1 (b).

Podemos ver a evolução dos treinos nas figuras Fig. 4.1 para as imagens RGB, Fig. 4.2 para as imagens de 5 canais para o modelo *Baseline* e Fig. 4.3 e 4.4 para o modelo principal.

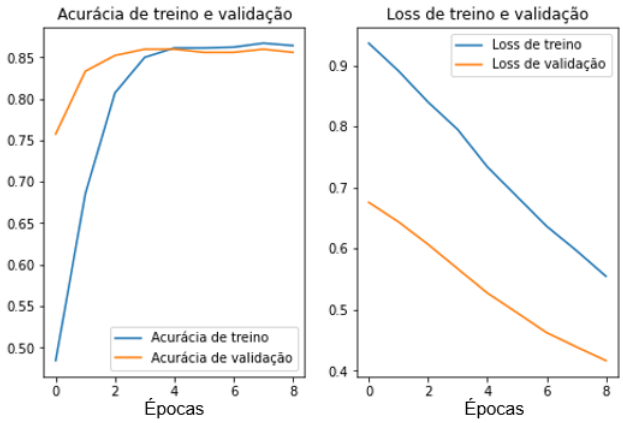


Figura 4.1: Evolução do treino do modelo *Baseline* nas imagens RGB. A acurácia satura com poucas épocas de treino, embora a *loss* continue a diminuir.

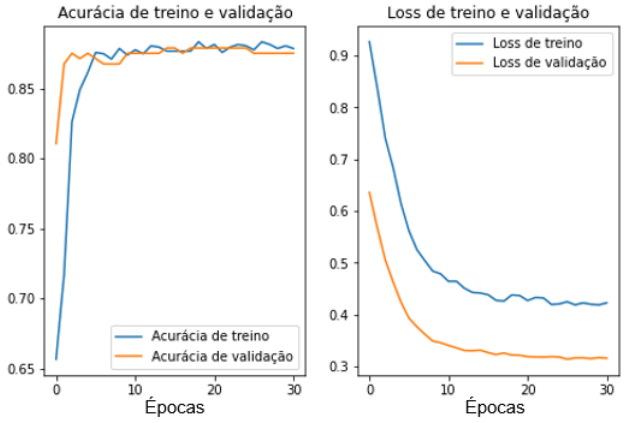


Figura 4.2: Evolução do treino do modelo *Baseline* nas imagens de 5 canais. Tanto as acurácias quanto as loss se saturam.

Podemos notar que no começo do treino a acurácia de validação é maior que a acurácia de treino (Figs. 4.1 - 4.5). Isso ocorre principalmente porque a camada de

dropout está ativa durante o treino, diminuindo a capacidade da rede. Já durante a validação a camada de *dropout* é desativada, portanto todos os neurônios da camada densa estão ativos, o que normalmente melhora os resultados [36]. Lembrando que a camada de *dropout* desativa uma porcentagem (no nosso caso 50%) dos neurônios da camada seguinte, o que força a rede a utilizar neurônios diferentes em cada época, evitando que poucos pesos sejam muito relevantes e outros pesos sejam pouco relevantes. Por conseguinte isso ajuda a aliviar o sobreajuste.

A quantidade de repetições necessárias para a obtenção das médias e desvios padrões é um ponto relevante. Para determinar a quantidade ideal, foram realizados experimentos com diferentes números de repetições (10, 30, 50, 100 e 300), utilizando o modelo Principal e 60% dos dados. Análises estatísticas foram realizadas para cada número de repetições, calculando-se as médias e desvios padrões. Após avaliar os resultados, concluiu-se que não houve uma diferença significante entre os resultados obtidos com 30 e 300 repetições. Portanto, para os resultados descritos neste estudo, utilizou-se o número de 30 repetições.

Na figura 4.1 a acurácia de treino começa baixa e se estabiliza em torno de 0,85. A loss de treino e validação ambas diminuem e com mais épocas poderia ficar menor, mas como a acurácia não aumentou, não há motivos para continuar o treino.

Na figura 4.2 a acurácia de treino e validação se estabilizam em torno de 0,88 enquanto que a loss deixa de diminuir em torno da epoca 15.

A quantidade de épocas de treino muda, por exemplo na figura 4.1 a rede *Baseline* em imagens RGB é treinada por 8 epocas enquanto que o treino da *Baseline* para as imagens 5ch é de 30 épocas. Isso ocorre devido ao *early stopping*, um parâmetro que indica se a rede deve continuar treinando ou não. São duas condições: a primeira é que se a acurácia de validação não melhorar por mais de um determinado número de épocas (no nosso caso 5). Entretanto, pode ocorrer de o modelo sempre melhorar a acurácia (ou outra métrica escolhida), mas muito lentamente. A segunda condição é de que essa melhora seja acima de um certo limiar, no nosso caso 0.001.

Essas medidas são necessárias para evitar o sobreajuste. Poderíamos deixar o modelo treinando por um número arbitrário de épocas, mas eventualmente o modelo iria "decorar"as respostas e a acurácia de validação diminuiria.

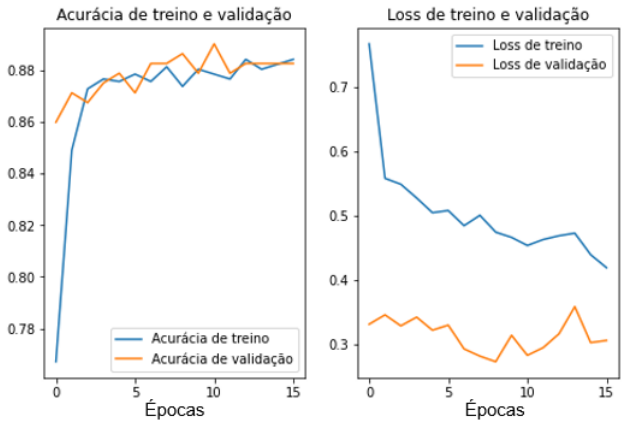


Figura 4.3: Evolução do treino do modelo *Principal* nas imagens RGB.

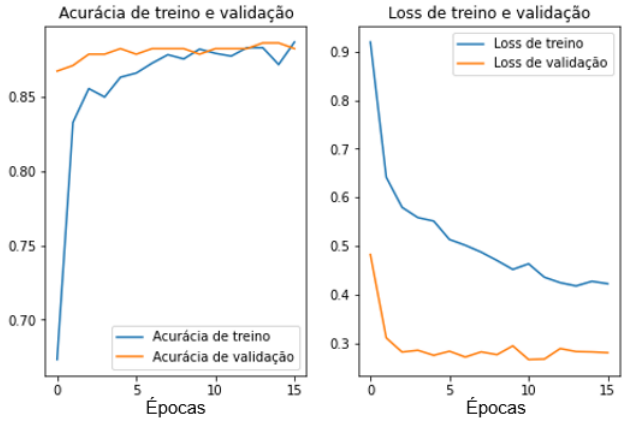


Figura 4.4: Evolução do treino do modelo *Principal* nas imagens de 5 canais.

Na figura 4.3 a acurácia de treino e validação variam um pouco mais que na figura 4.2, os valores em torno de 0,88, enquanto que a não há muita melhora na loss.

Na figura 4.4 a acurácia de treino e validação se estabilizam em torno de 0,88 enquanto que a loss de validação deixa de diminuir em torno da época 3.

4.1 Transferência de aprendizado - rede Xception

A transferência de aprendizado é uma técnica usada quando adaptamos uma rede que já foi treinada e ajustada para uma tarefa usando outros datasets para um outro problema. No presente trabalho utilizamos a rede Xception [70], que faz parte de uma família de modelos de melhor performance no dataset da ImageNet [82]. Para podermos aproveitar as camadas treinadas, precisamos fazer algumas modificações: congelar as camadas inferiores, isto é, as primeiras camadas, mais próximas do input, uma vez que já estão ajustadas, e apenas modificar as últimas camadas. No caso, modificamos as camadas de global average pooling e a camada densa de output. No lugar colocamos a

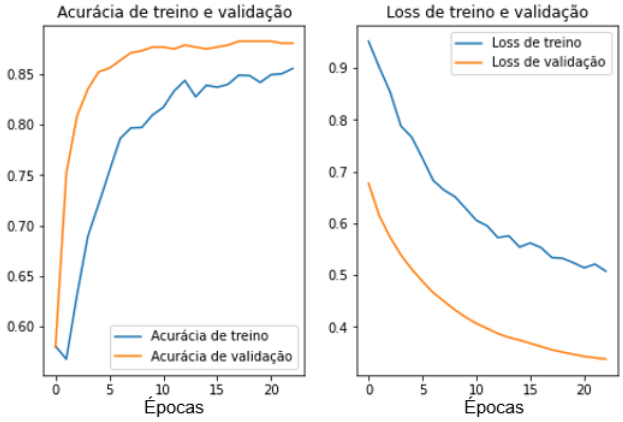


Figura 4.5: Rede Xception treinada com apenas as camadas superiores, para imagens RGB.

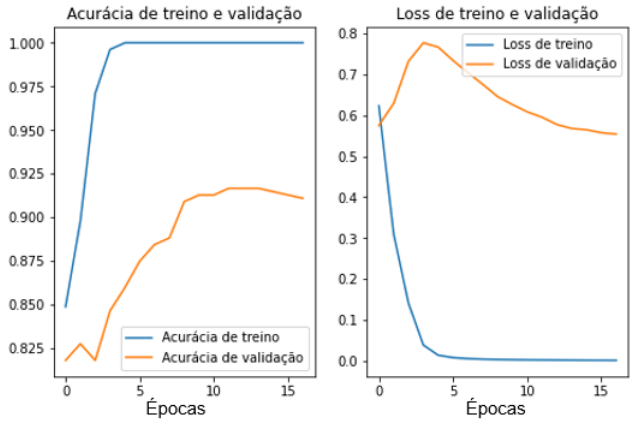


Figura 4.6: Evolução do treino da Xception com todas as camadas, para imagens RGB. Há um overfitting a partir da época 4 de treino, uma vez que a acurácia de treino satura em 1 e a loss vai para zero.

nossa própria camada de pooling seguida da camada de output, usando a sigmoide como função de ativação e para apenas 2 outputs ao invés dos 1000 originais. A evolução de um dos treinos está na Figura 4.5 e os resultados na Tabela 4.1.

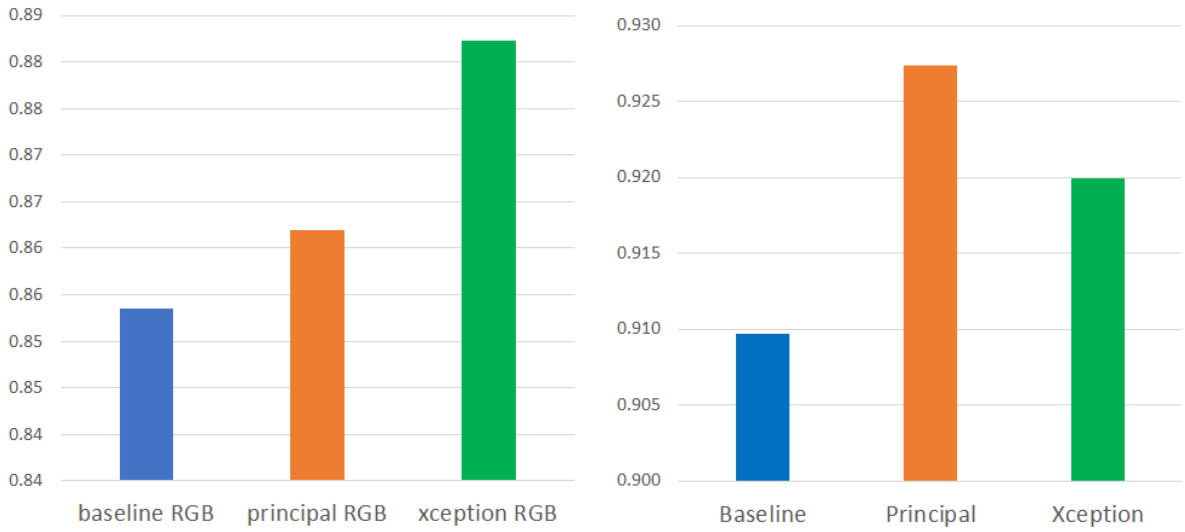
Tabela 4.1: Comparação da Xception usando todas as camadas e apenas as últimas. Um asterisco indica que o p-valor é menor que 0,05 e dois asteriscos indicam valores menores que 10^{-3}.

	Acurácia	P-valor Acurácia	AUC	P-valor AUC
Camadas superiores treinadas	0,859	-	0,914	-
Todas as camadas treinadas	0,882**	3.2E-6	0,920	0,11

Depois de treinar apenas as camadas superiores, descongelamos as outras camadas e treinamos novamente, agora usando todas as camadas da rede. Podemos observar que a acurácia de validação da rede Xception vai rapidamente para 1 e se mantém constante, portanto houve um *overfitting* (os resultados podem ser vistos na Tabela 4.1 e a evolução

do treino na Figura 4.6). Outro ponto que indica overfitting pode ser visto no comportamento na curva de *loss*, onde a *loss* de validação está muito maior que a *loss* de treino e elas não possuem tendências similares, a *loss* de treino diminui e se mantém constante enquanto que a *loss* de validação aumenta no começo e depois diminui. Algumas possíveis soluções para o *overfitting* é, por exemplo, modificar a porcentagem de *dropout*, aumentar a quantidade de dados, já que 1650 é considerado um número pequeno para tarefas de classificação de imagens. Também podemos mais modificações no *data augmentation*, já que no nosso trabalho fizemos apenas duas, espelhamento na vertical e espelhamento na horizontal.

Em relação a usar todas as camadas ou apenas as duas últimas camadas da rede Xception, observamos pela tabela 4.1 que há evidências de melhora na acurácia, em imagens RGB, quando utilizamos todas as camadas. Portanto vamos nos referir aos resultados da Xception como sendo treinados por todas as camadas.



(a) Acurácia **(b) AUC**

Figura 4.7: Comparação entre os modelos para o conjunto RGB, para as métricas (a) acurácia e (b) AUC. Podemos notar que a Xception obteve resultados melhores para a acurácia, e embora a AUC seja 0.007 menor que o modelo Principal, não é uma diferença estatisticamente significante.

4.2 Comparando os modelos

Observamos que a rede Xception melhora a acurácia em 3% em relação ao modelo Baseline. Comparado com o modelo Baseline podemos ver na Tabela 4.2 (e Fig. 4.7 que o modelo Xception melhorou em 3% a acurácia. A diferença na AUC não foi significativa. entre Xception e Baseline. O modelo principal também melhorou a acurácia e obteve um AUC 2% maior quando comparado com o Baseline, como esperado.

Podemos ver na Tabela 4.2 que o modelo principal teve um AUC de 0,927, cerca de 3% melhor que o Baseline .

Tabela 4.2: Resultados para os modelos e comparação com o baseline

	Acurácia	P-valor	AUC	P-valor
Baseline	0,853	-	0,910	-
Principal	0,862*	0,046	0,927*	0,0003
Xception	0,882**	3.2E-6	0,920	0,064

4.3 RGB vs 3ch vs 5ch

Tabela 4.3: Resultados para o modelo baseline. 5ch é o melhor dos 3 mas essa melhora não é significante pra auc

	Acurácia	P-valor	AUC	P-Valor
RGB	0,854	-	0,910	-
3ch	0,855	0,794	0,890*	0,006
5ch	0,864*	0,026	0,910	0,987

Tabela 4.4: Resultados para o modelo principal. RGB, 3ch e 5ch geram resultados equivalentes.

	Acurácia	P-valor	AUC	P-Valor
RGB	0,862	-	0,927	-
3ch	0,862	0,927	0,922	0,402
5ch	0,867	0,220	0,930	0,581

Tabela 4.5: Resultados para o modelo Xception. Os resultados são equivalentes, seja usando 3ch, seja usando RGB.

	Acurácia	P-valor	AUC	P-valor
RGB	0,882	-	0,920	-
3ch	0,878	0,30	0,921	0,85

Até agora analisamos se a rede Xception tem uma performance melhor quando treinamos apenas as últimas camadas ou quando fazemos o treino completo, e se os modelos Principal e Xception performam melhor que o Baseline. Vamos comparar agora se utilizar 3 canais (melanina, oxiemoglobina e tons de cinza) ou 5 canais (vermelho, verde, azul, melanina, oxiemoglobina) é mais eficiente do que utilizar apenas os canais RGB

Os p-valores das tabelas 4.3, 4.4 e 4.5 foram calculados comparando RGB com 3ch e RGB com 5ch.

Quanto a utilizar a melanina e oxi-hemoglobina, para o modelo Baseline não há evidência significante na melhora da acurácia quando utilizamos 3 canais e há evidências significantes para uma pequena melhora na acurácia quando utilizamos 5 canais, Tabela 4.3. Para os modelos Principal e Xception, não há evidências significativas de melhora para nenhuma das métricas.

Os resultados estão representados nas figuras 4.8 e 4.9 onde azul é o modelo Baseline, laranja é o modelo principal e verde é o modelo Xception. As barras de incertezas são os desvios padrões.

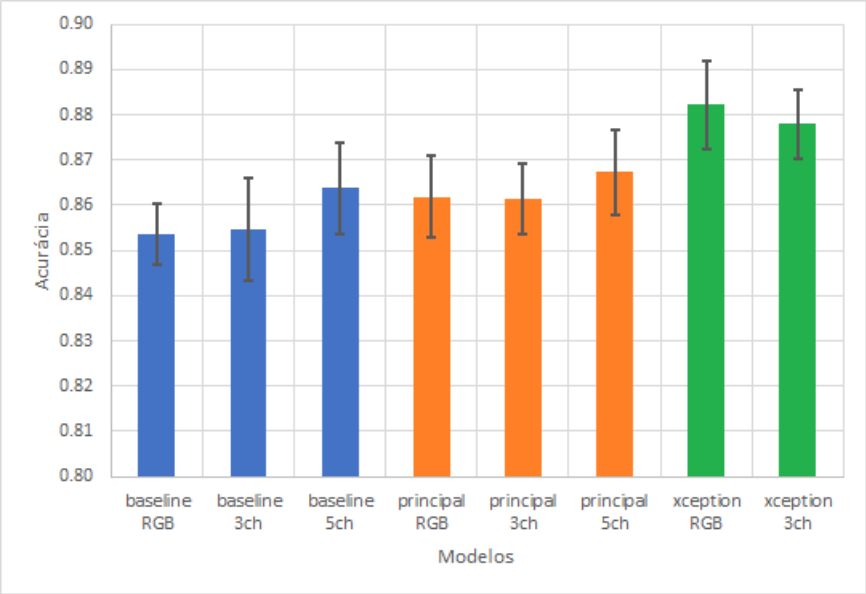


Figura 4.8: Comparação da acurácia dos modelos. Não há evidência de melhora ao modificar os canais. (exceto baseline 5ch com melhora de 1%). As barras de erro são os desvios padrões.

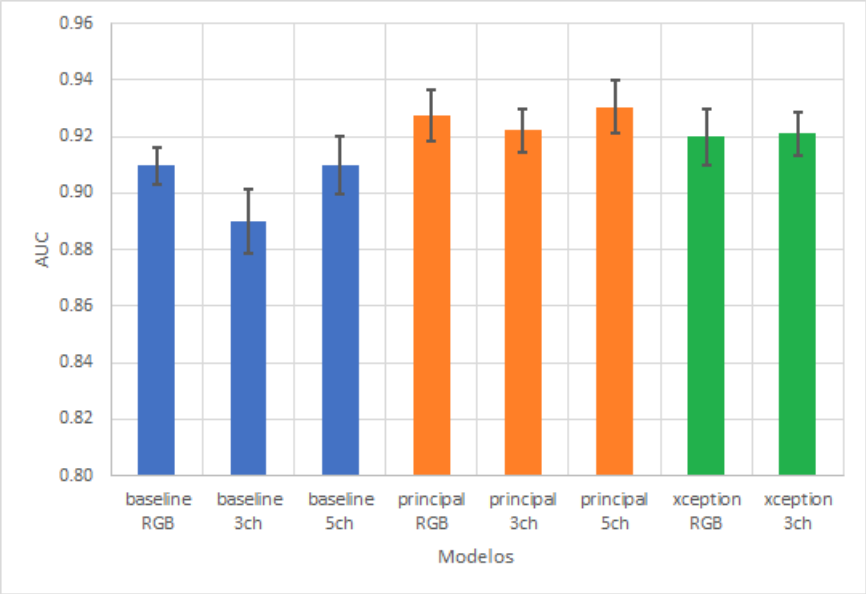
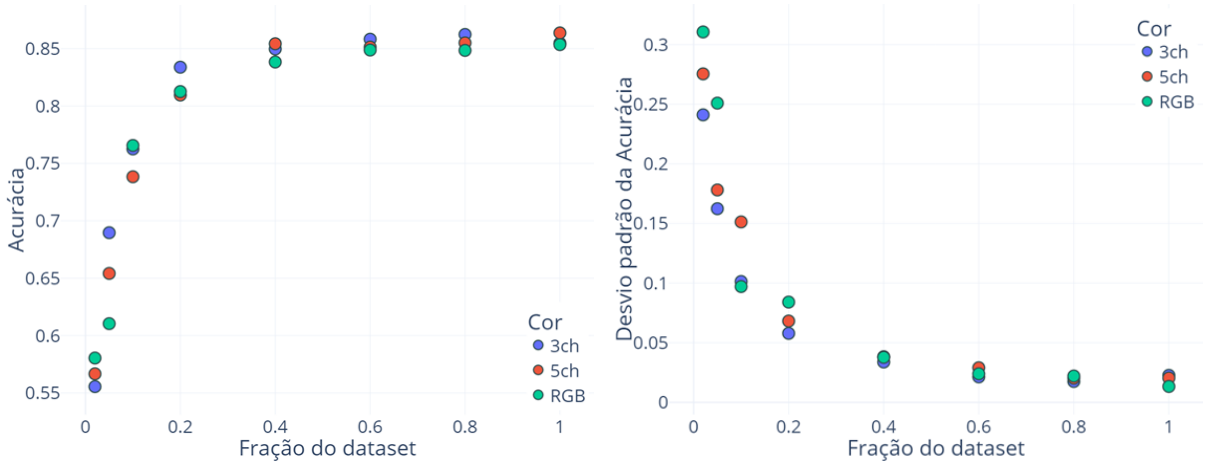


Figura 4.9: Comparação da AUC dos modelos. Não há evidência de melhora ao modificar os canais. As barras de erro são os desvios padrões.

4.4 Variação no tamanho do dataset / Regime de poucas imagens

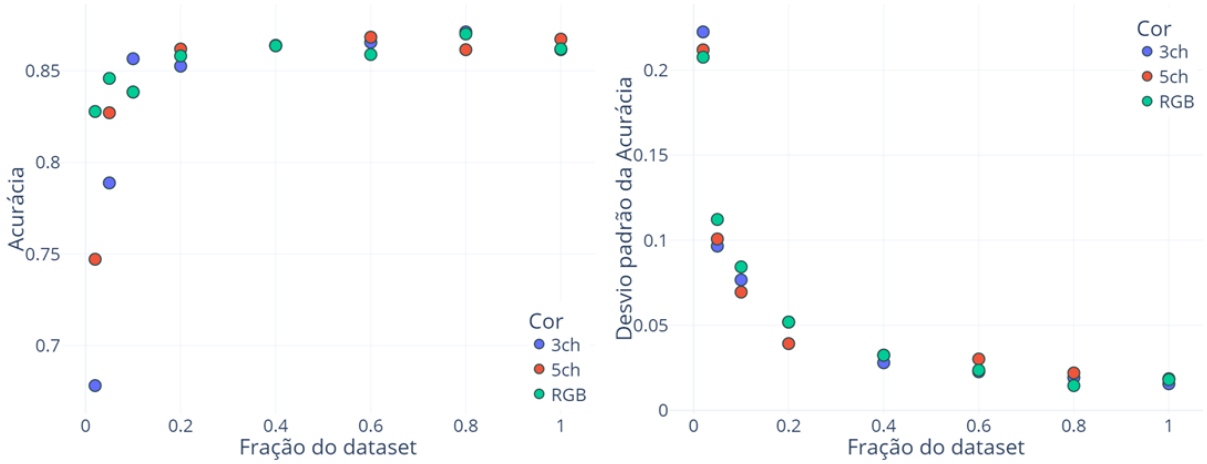


(a) Acurácia **(b) Desvio padrão da acurácia**

Figura 4.10: (a) Acurácia do modelo Baseline e (b) seu desvio padrão

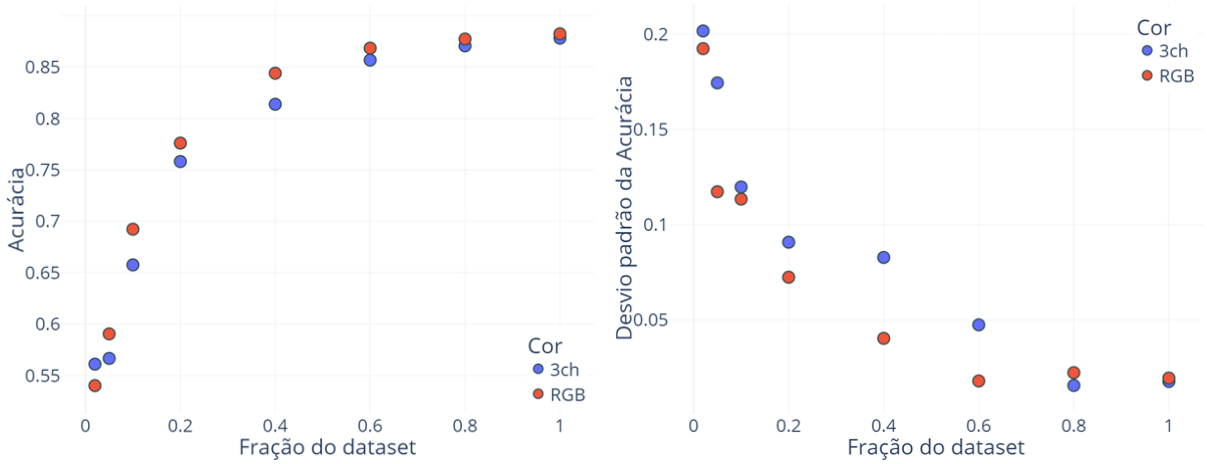
Em estudos de aprendizado de máquina e aprendizado profundo, é conhecido que um grande volume de dados pode conduzir a modelos mais precisos e previsões mais acuradas. Apesar disso, a aquisição de grandes conjuntos de dados nem sempre é factível, especialmente em áreas como a medicina, onde o recrutamento de voluntários pode ser limitado a algumas dezenas de indivíduos.

Diante desse cenário, torna-se relevante investigar o impacto do uso de canais 3ch e 5ch na análise de dados, à medida que se varia a quantidade de informações disponíveis.



(a) Acurácia **(b) Desvio padrão da acurácia**

Figura 4.11: (a) Acurácia do modelo Principal e (b) seu desvio padrão



(a) Acurácia **(b) Desvio padrão da acurácia**

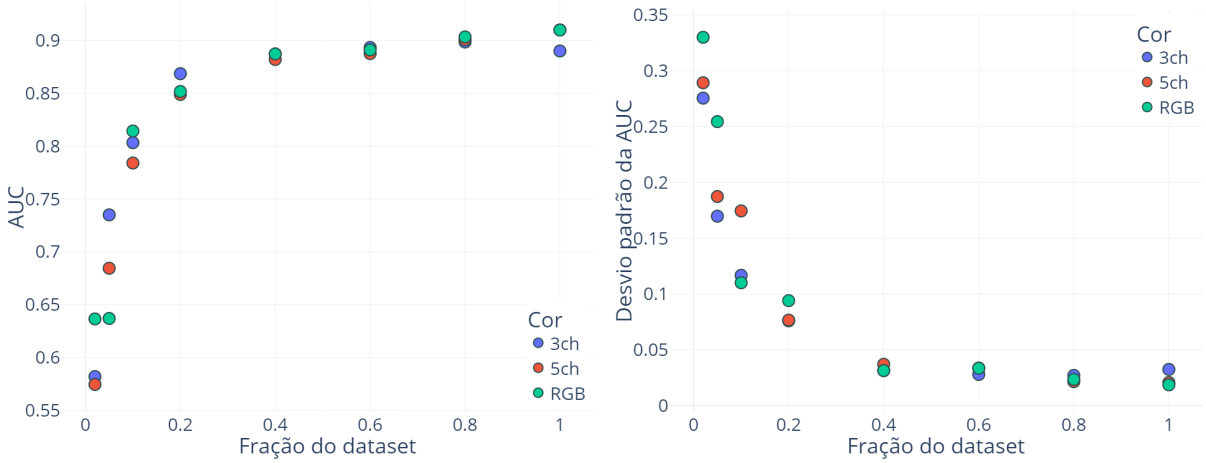
Figura 4.12: (a) Acurácia do modelo Xception e (b) seu desvio padrão

Nesse sentido, espera-se avaliar se a inclusão desses canais pode contribuir para a obtenção de resultados mais robustos, mesmo quando a quantidade de dados disponíveis é reduzida.

Fizemos a divisão dessa forma: primeiro separamos as proporções de 64% para treino, 16% de validação e 20% de teste aleatoriamente, mas mantendo as proporções de controle e patologia. Em seguida amostramos uma certa porcentagem (2%, 5%, 10%, 20%, 40%, 60%, 80% e 100%) desses conjuntos.

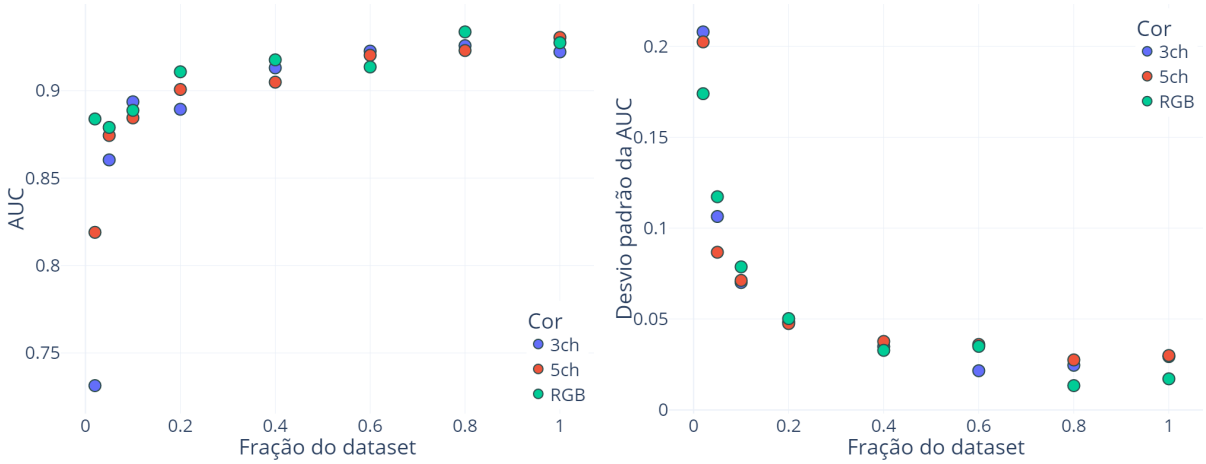
Como temos 1650 imagens e 1130 são de olhos com alguma patologia, um classificador aleatório acertaria 50% das vezes e um classificador que só retornasse "patologia"acertaria 68% das vezes. Ao escolher as frações dos mantemos a mesma proporção das imagens, portanto a acurácia mínima para um modelo ser considerado melhor do que um modelo da um único output é de 68%.

No geral, ao aumentar a quantidade de dados as métricas melhoram, mas não há



(a) AUC **(b) Desvio padrão da AUC**

Figura 4.13: (a) AUC do modelo Baseline e (b) seu desvio padrão

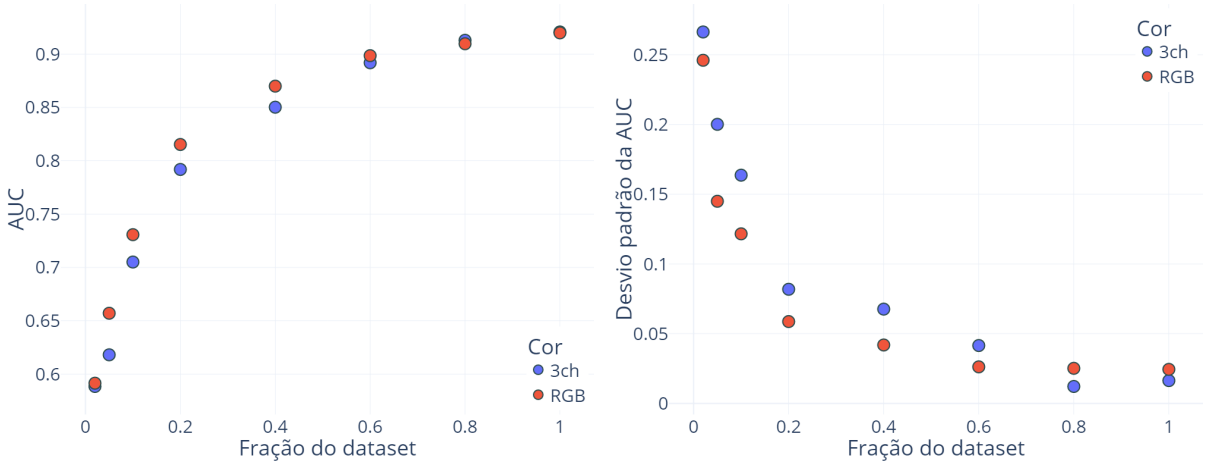


(a) AUC **(b) Desvio padrão da AUC**

Figura 4.14: (a) AUC do modelo Principal e (b) seu desvio padrão

evidências de que uma alteração nos canais mude significativamente os resultados. Por exemplo, vemos em Fig. 4.11 (a) que no regime de poucos dados (fração entre 2% e 20%), os conjuntos 3ch e 5ch tem uma performance pior ou igual à RGB. Os dados completos para gerar as figuras estão no Apêndice. Ambos, o modelo Baseline Fig. 4.10 e Principal Fig. 4.11, começam com valores pequenos de acurácia, $0,55$ para Baseline e entre $0,6$ e $0,85$ para o Principal, mas que aproximam de uma assíntota conforme aumentamos os dados. Os resultados da rede Xception estão representados nas figuras Fig. 4.12 para acurácia e Fig. 4.15 para a AUC. Podemos ver que não há diferenças significativas entre RGB e 3ch. É interessante notar que a Xception não teve bons resultados no regime de poucos dados, talvez devido ao fato de que o *dataset* utilizado seja bem diferente daquele no qual a rede foi originalmente treinada.

Vemos que os modelos Baseline e Principal saturaram a acurácia em torno de 40% dos dados, ou seja, adicionar mais dados não melhorou a acurácia. Entretanto, o modelo



(a) AUC **(b) Desvio padrão da AUC**

Figura 4.15: (a) AUC do modelo Xception e (b) seu desvio padrão

Xception continuou a aumentar a acurácia e não chegou a saturar, então com mais dados a Xception teria uma acurácia ainda maior.

Em relação aos desvios padrões, ao invés de colocar junto com os gráficos representamos em figuras separadas, os itens (b) das Figuras 4.10 - 4.15. Vemos que há uma grande variação quando há uma pouca quantidade de dados, mas que conforme aumentamos os dados essa variação diminui e tende a ficar constante, o que indica uma menor variação nos resultados uma vez que há mais exemplos para as redes treinarem. Isso é válido para todos os modelos, Baseline, Principal e Xception.

Embora no caso deste projeto os pré-processamentos utilizados não tenham melhorando a performance das redes, outros tipos de pré-processamento, como por exemplo "*contrast limited adaptive histogram equalization*"(CLAHE) [83, 84], conseguiram melhorar a performance das redes, em conjunto com outras técnicas.

Após análise dos resultados, foi constatado que as redes neurais são capazes de capturar os padrões necessários para realizar a classificação de imagens. Ademais, evidenciou-se que nem todo tipo de pré-processamento contribui para o aprimoramento dos resultados obtidos pelas redes neurais.

5 Conclusões

Neste trabalho, analisamos se um pré-processamento em imagens de retinografia, especificamente a adição dos mapas de melanina e oxiemoglobina, melhora os resultados de redes neurais convolucionais para identificar a presença de doenças nos olhos, principalmente para o caso onde temos poucas imagens.

Quando utilizado todas as imagens disponíveis, para as imagens RGB ception obteve uma acurácia maior que os outros modelos ara a AUC a Xception e Principal obtiveram resultados equivalentes.

No regime de poucos dados (165 imagens no nosso caso) o modelo Principal se saiu melhor que a Xception, um modelo mais complexo, possivelmente pela Xception ter feito um sobreajuste nos dados.

A partir dos resultados, podemos concluir que transformar os canais RGB em canais que representam os cromóforos ão é suficiente para melhora a acurácia e AUC de modelos de redes neurais. Usando modelos mais complexos ou modelos pré-treinados conseguimos obter uma acurácia e AUC maiores sem a necessidade de ser feito um pré-processamento dos cromóforos, seja co oucas ou muitas imagens.

REFERÊNCIAS

[1] Ministério da Saúde do Brasil. Doenças oculares — português (brasil). url:https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/doencas-oculares-1,, 2020.

[2] Nihat Sayin, Necip Kara, and Gökhan Pekel. Ocular complications of diabetes mellitus. *World journal of diabetes*, 6(1):92, 2015.

[3] B Robinson. Prevalence of asymptomatic eye disease. *Canadian Journal of Optometry*, 65(5):177–186, 2003.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[5] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

[6] Aditya Jyoti Paul. Advances in classifying the stages of diabetic retinopathy using convolutional neural networks in low memory edge devices. 2021.

[7] Rishab Gargeya and Theodore Leng. Automated identification of diabetic retinopathy using deep learning. *Ophthalmology*, 124:962–969, 2017.

[8] Yaniv Barkana and Syril Dorairaj. Re: Tham et al.: Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis (ophthalmology 2014; 121: 2081-90). *Ophthalmology*, 122(7):e40–e41, 2015.

[9] Karen Allison, Deepkumar Patel, and Omobolanle Alabi. Epidemiology of glaucoma: the past, present, and predictions for the future. *Cureus*, 12(11), 2020.

[10] Robert N Weinreb and Peng Tee Khaw. Primary open-angle glaucoma. *The lancet*, 363(9422):1711–1720, 2004.

[11] Wayne Smith, Jacqueline Assink, Ronald Klein, Paul Mitchell, Caroline CW Klaver, Barbara EK Klein, Albert Hofman, Susan Jensen, Jie Jin Wang, and Paulus TVM de Jong. Risk factors for age-related macular degeneration: pooled findings from three continents. *Ophthalmology*, 108(4):697–704, 2001.

[12] Marcio Bittar Nehemy. Degeneração macular relacionada à idade: novas perspectivas. *Arquivos Brasileiros de Oftalmologia*, 69:955–958, 2006.

[13] Laurence S Lim, Paul Mitchell, Johanna M Seddon, Frank G Holz, and Tien Y Wong. Age-related macular degeneration. *The Lancet*, 379(9827):1728–1738, 2012.

[14] Philip J Rosenfeld, David M Brown, Jeffrey S Heier, David S Boyer, Peter K Kaiser, Carol Y Chung, and Robert Y Kim. Ranibizumab for neovascular age-related macular degeneration. *New England Journal of Medicine*, 355(14):1419–1431, 2006.

[15] Matheus Soares Monteiro. Desenvolvimento de um sistema embarcado para auxílio no diagnóstico e acompanhamento de degeneração macular relacionada a idade utilizando imagens da retina. Master's thesis, Universidade Federal de Pernambuco, 2019.

[16] Quresh Mohamed, Mark C Gillies, and Tien Y Wong. Management of diabetic retinopathy: a systematic review. *Jama*, 298(8):902–916, 2007.

[17] M Sousa, JM Silva, and JF Raposo. ediabetes©: Protocolo de implementação de um programa colaborativo de apoio à auto-gestão e literacia digital na diabetes tipo 2. *Revista Portuguesa de Diabetes*, 16(3):112–117, 2021.

[18] Liebmann JM. Cioffi GA. *Diseases of the visual system. In: Goldman L, Schafer AI, eds.*, chapter 395. Elsevier, Philadelphia, 2020.

[19] Patrick J. Saine and Marshall E. Tyler. Butterworth-Heinemann Medical, UK, 2002.

[20] Caitlin L. M. Kakigi, Kuldev Singh, Sophia Y. Wang, Wayne T. Enanoria, and Shan C. Lin. Self-reported Calcium Supplementation and Age-Related Macular Degeneration . *JAMA Ophthalmology*, 133(7):746–754, 07 2015.

[21] Ling-Ping Cen, Jie Ji, Jian-Wei Lin, Si-Tong Ju, Hong-Jie Lin, Tai-Ping Li, Yun Wang, Jian-Feng Yang, Yu-Fen Liu, Shaoying Tan, et al. Automatic detection of 39 fundus diseases and conditions in retinal photographs using deep neural networks. *Nature communications*, 12(1):1–13, 2021.

[22] B. Cassin and S. Solomon. Triad Publishing Company, Gainesville, Florida, 1990.

[23] Saine PJ. Fundus photography: Fundus camera optics. urlhttp://www.opsweb.org/Op-Photo/Fundus/CFundus/funphot3.htm, 2006.

[24] An Ran Ran, Clement C. Tham, Poemen P. Chan, Ching-Yu Cheng, Yih-Chung Tham, Tyler Hyungtaek Rim, and Carol Y. Cheung. Deep learning in glaucoma with optical coherence tomography: a review. *Eye, Review article*, 35(1):188–201, 2021.

[25] Silke Aumann, Sabine Donner, Jörg Fischer, and Frank Müller. *Optical Coherence Tomography (OCT): Principle and Technical Realization*, pages 59–85. Springer International Publishing, Cham, 2019.

[26] Mark C. Pierce, John Strasswimmer, B. Hyle Park, Barry Cense, and Johannes F. de Boer. Advances in optical coherence tomography imaging for dermatology. *Journal of Investigative Dermatology*, 123(3):458–463, 2004.

[27] Andrew Jr, Alon Harris, Josh Gross, Ingrida Januleviciene, Aaditya Shah, and Brent Siesky. Optical coherence tomography angiography: An overview of the technology and an assessment of applications for clinical research. *British Journal of Ophthalmology*, 101, 10 2016.

[28] Romano A. Waheed Nadia K. Duker Jay S. De Carlo, Talisa E. A review of optical coherence tomography angiography (octa). *International Journal of Retina and Vitreous*, 04 2015.

[29] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[30] Raffaele Nuzzi, Giacomo Boscia, Paola Marolo, and Federico Ricardi. The impact of artificial intelligence and deep learning in eye diseases: A review. *Frontiers in Medicine*, 8, 2021.

[31] Stefano A Bini. Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care? *The Journal of arthroplasty*, 33(8):2358–2361, 2018.

[32] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 1943 5:4*, 5:115–133, 12 1943.

[33] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Abubakar Malah Umar, Okafor Uchenwa Linus, Humaira Arshad, Abdullahi Aminu Kazaure, Usman Gana, and Muhammad Ubale Kiru. Comprehensive review of artificial neural network applications to pattern recognition. *IEEE Access*, 7:158820–158846, 2019.

[34] Qiming Zhang, Haoyi Yu, Martina Barbiero, Baokai Wang, and Min Gu. Artificial neural networks enabled by nanophotonics. *Light: Science & Applications*, 8(1):1–14, 2019.

[35] Leandro Fleck, Maria Hermínia Ferreira Tavares, Eduardo Eyng, Andrieli Cristina Helmann, and MA de M Andrade. Redes neurais artificiais: Princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia*, 1(13):47–57, 2016.

[36] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* "O'Reilly Media, Inc.", 2019.

[37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[38] Andreas C Müller and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists.* "O'Reilly Media, Inc.", 2016.

[39] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015.

[40] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[41] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[42] Aatila Mustapha, Lachgar Mohamed, and Kartit Ali. An overview of gradient descent algorithm optimization in machine learning: Application in the ophthalmology field. In *International Conference on Smart Applications and Data Analysis*, pages 349–359. Springer, 2020.

[43] Ange Tato and Roger Nkambou. Improving adam optimizer. 2018.

[44] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[45] Sebastian Bock, Josef Goppold, and Martin Weiß. An improvement of the convergence proof of the adam-optimizer. *arXiv preprint arXiv:1804.10587*, 2018.

[46] Anuraganand Sharma. Guided stochastic gradient descent algorithm for inconsistent datasets. *Applied Soft Computing*, 73:1068–1080, 2018.

[47] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[48] Nikhil Ketkar and Jojo Moolayil. Convolutional neural networks. In *Deep Learning with Python*, pages 197–242. Springer, 2021.

[49] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[50] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.

[51] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[52] Yi-Tong Zhou and Rama Chellappa. Computation of optical flow using a neural network. In *ICNN*, pages 71–78, 1988.

[53] Maximum pooling — kaggle. url:https://www.kaggle.com/code/ryanholbrook/maximum-pooling,.

[54] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[55] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[56] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42(11):1–13, 2018.

[57] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature 2017 542:7639*, 542(7639):115–118, jan 2017.

[58] Stephanie A. Harmon, Thomas H. Sanford, Sheng Xu, Evrim B. Turkbey, Holger Roth, Ziyue Xu, Dong Yang, Andriy Myronenko, Victoria Anderson, Amel Amalou, Maxime Blain, Michael Kassin, Dilara Long, Nicole Varble, Stephanie M. Walker, Ulas Bagci, Anna Maria Ierardi, Elvira Stellato, Guido Giovanni Plensich, Giuseppe Franceschelli, Cristiano Girlando, Giovanni Irmici, Dominic Labella, Dima Hammoud, Ashkan Malayeri, Elizabeth Jones, Ronald M. Summers, Peter L. Choyke, Daguang Xu, Mona Flores, Kaku Tamura, Hirofumi Obinata, Hitoshi Mori, Francesca Patella, Maurizio Cariati, Gianpaolo Carrafiello, Peng An, Bradford J. Wood,

and Baris Turkbey. Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. *Nature Communications 2020 11:1*, 11(1):1–7, aug 2020.

[59] Varun Gulshan, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip C. Nelson, Jessica L. Mega, and Dale R. Webster. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, 316(22):2402–2410, dec 2016.

[60] Philippe M. Burlina, Neil Joshi, Katia D. Pacheco, David E. Freund, Jun Kong, and Neil M. Bressler. Use of Deep Learning for Detailed Severity Characterization and Estimation of 5-Year Risk Among Patients With Age-Related Macular Degeneration. *JAMA Ophthalmology*, 136(12):1359–1366, dec 2018.

[61] Yanyan Dong, Qinyan Zhang, Zhiqiang Qiao, and Ji Jiang Yang. Classification of cataract fundus image based on deep learning. *IST 2017 - IEEE International Conference on Imaging Systems and Techniques, Proceedings*, 2018-January:1–5, jul 2017.

[62] Zhixi Li, Yifan He, Stuart Keel, Wei Meng, Robert T. Chang, and Mingguang He. Efficacy of a Deep Learning System for Detecting Glaucomatous Optic Neuropathy Based on Color Fundus Photographs. *Ophthalmology*, 125(8):1199–1206, aug 2018.

[63] José Ignacio Orlando, Huazhu Fu, João Barbossa Breda, Karel van Keer, Deepti R. Bathula, Andrés Diaz-Pinto, Ruogu Fang, Pheng Ann Heng, Jeyoung Kim, Joon Ho Lee, Joonseok Lee, Xiaoxiao Li, Peng Liu, Shuai Lu, Balamurali Murugesan, Valery Naranjo, Sai Samarth R. Phaye, Sharath M. Shankaranarayana, Apoorva Sikka, Jaemin Son, Anton van den Hengel, Shujun Wang, Junyan Wu, Zifeng Wu, Guanghui Xu, Yongli Xu, Pengshuai Yin, Fei Li, Xiulan Zhang, Yanwu Xu, and Hrvoje Bogunović. REFUGE Challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs. *Medical Image Analysis*, 59:101570, jan 2020.

[64] Tripti Goel, R Murugan, Seyedali Mirjalili, and Deba Kumar Chakrabartty. Optconet: an optimized convolutional neural network for an automatic diagnosis of covid-19. *Applied Intelligence*, 51(3):1351–1366, 2021.

[65] Ali Abbasian Ardakani, Alireza Rajabzadeh Kanafi, U Rajendra Acharya, Nazanin Khadem, and Afshin Mohammadi. Application of deep learning technique to manage covid-19 in routine clinical practice using ct images: Results of 10 convolutional neural networks. *Computers in biology and medicine*, 121:103795, 2020.

[66] Karl Weiss, Taghi M. Khoshgoftaar, and Ding Ding Wang. A survey of transfer learning. *Journal of Big Data*, 3, 12 2016.

[67] Suayder Milhomem Costa. Uso de transferência de aprendizado e rede neural sem peso para detecção de imagens de defeitos em vias pavimentadas. 2019.

[68] Bens Pardamean, Tjeng Wawan Cenggoro, Reza Rahutomo, Arif Budiarto, and Etikan Kandasamy Karuppiah. Transfer learning from chest x-ray pre-trained convolutional neural network for learning mammogram data. *Procedia Computer Science*, 135:400–407, 2018.

[69] Brady Kieffer, Morteza Babaie, Shivam Kalra, and Hamid R Tizhoosh. Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2017.

[70] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[71] Nelson González Machín. Estimación de pose de la cara para aplicaciones de reidentificación y biometría blanda. B.S. thesis, 2019.

[72] MZ Naser and Amir Alavi. Insights into performance fitness and error metrics for machine learning. *arXiv preprint arXiv:2006.00887*, 2020.

[73] Davide Chicco, Niklas Tötsch, and Giuseppe Jurman. The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData mining*, 14(1):1–22, 2021.

[74] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

[75] Mark H Zweig and Gregory Campbell. Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. *Clinical chemistry*, 39(4):561–577, 1993.

[76] PE Griner. Selection and interpretation of diagnostic tests and procedures: Annals of internal medicine. *Ann Intern Med*, 94:555–600, 1981.

[77] Antonio Alberto de Sousa Dias, Murilo Sanches Sampaio, and George Cunha Cardoso. Mapeamento de oxigenação de fundo de olho através de fotografias e lei de beer-lambert. 9 2019.

[78] Colab gpu. https://blog.tensorflow.org/2022/09/colabs-pay-as-you-go-offers-more-access-to-powerful-nvidia-compute-for-machine.html. Accessed: 2022-10-30.

[79] Zhuo Zhang, Feng Shou Yin, Jiang Liu, Wing Kee Wong, Ngan Meng Tan, Beng Hai Lee, Jun Cheng, and Tien Yin Wong. Origa-light: An online retinal fundus image database for glaucoma analysis and research. In *2010 Annual international conference of the IEEE engineering in medicine and biology*, pages 3065–3068. IEEE, 2010.

[80] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[81] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[82] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[83] K Shankar, Yizhuo Zhang, Yiwei Liu, Ling Wu, and Chi-Hua Chen. Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification. *IEEE Access*, 8:118164–118173, 2020.

[84] Erdal Tasci, Caner Uluturk, and Aybars Ugur. A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection. *Neural Computing and Applications*, 33(22):15541–15555, 2021.

[85] Tabulated molar extinction coefficient for hemoglobin in water. url:https://omlc.org/spectra/hemoglobin/summary.html.

Apêndices

A Separação de cromófaros

Algumas das doenças mais comuns que causam cegueira, como retinopatia diabética, glaucoma e oclusão vascular da retina, afetam a retina e estão relacionadas com a entrega deficiente de oxigênio ou à redução no metabolismo. A oximetria da retina permite a medida desses parâmetros. Entretanto, o diagnóstico e monitoramento atualmente dependem de mudanças na estrutura causada por danos previamente causados pela doença na retina. Estudos anteriores obtiveram sucesso para a determinação de oxigenação da retina, porém utilizando mais de uma câmera RGB, filtros ópticos passa-banda e espectrômetros, ou inferiram indiretamente a oxigenação da retina utilizando medidas de oxigenação de outros tecidos do corpo humano. A proposta deste estudo é determinar os níveis de oxigenação de tecidos de fundo de olho usando uma câmera RGB, utilizando retinógrafos convencionais já existentes, e permitindo obter medidas quantitativas antes que danos maiores ocorram na retina.

Após obtidas as imagens, foram utilizados métodos de processamento de imagem utilizando conhecimento espectroscópico dos principais cromóforos presentes na retina, a fim de estimar o nível de oxigenação dos vasos. A metodologia de processamento das imagens, que utiliza a física da interação da luz com os tecidos, é detalhada abaixo. Utilizando o modelo de [77] assume-se que a retina é constituída principalmente de oxihemoglobina, desoxihemoglobina e melanina, sendo que cada substância possui um espectro de absorção diferente. A lei de Beer-Lambert diz que existe uma relação entre a concentração de uma substância e a transmissão da luz através dela. Também, a relação entre a transmissão e o comprimento de penetração da luz.

$$\frac{I_1}{I_0} = 10^{-\epsilon(\lambda)*l*C}. \tag{A.1}$$

Onde I_0 é a intensidade da luz incidente, I_1 é a intensidade da luz refletida, l é o comprimento de interação da luz com a substância (m), c é a concentração molar da substância (mol/m^3) $\epsilon(\lambda)$ é a absorvidade molar do cromóforo, em $Lmol^{-1}m^{-1}$. Conhecidos os comprimentos de penetração e a absorvidade molar, determina-se a concentração da substância através da equação (2)

$$A = -\log\frac{I_1}{I_0} = -\epsilon(\lambda) * l * C, \tag{A.2}$$

onde A é a absorvância.

Podemos reescrever a equação A.2 para oxihemoglobina, desoxihemoglobina e melanina

$$-\log I(\lambda) = \epsilon_{HbO2}(\lambda) l_{HbO2}(\lambda) c_{HbO2} + \epsilon_{HbR}(\lambda) l_{HbR}(\lambda) c_{HbR} + \epsilon_{Mel}(\lambda) l_{Mel}(\lambda) c_{Mel} \quad \text{(A.3)}$$

Faremos l o mesmo para todos, diminuindo uma variável no sistema. Como usamos uma câmera de 3 canais (RGB), teremos um sistema de equações

$$\begin{pmatrix} -\log(I_r) \\ -\log(I_g) \\ -\log(I_b) \end{pmatrix} = \begin{pmatrix} \epsilon_{HbO_2}(\lambda_r) & \epsilon_{Hb}(\lambda_r) & \epsilon_{Mel}(\lambda_r) \\ \epsilon_{HbO_2}(\lambda_g) & \epsilon_{Hb}(\lambda_g) & \epsilon_{Mel}(\lambda_g) \\ \epsilon_{HbO_2}(\lambda_b) & \epsilon_{Hb}(\lambda_b) & \epsilon_{Mel}(\lambda_b) \end{pmatrix} \begin{pmatrix} C_{HbO_2} \\ C_{Hb} \\ C_{Mel} \end{pmatrix} \quad \text{(A.4)}$$

C_i com i = HbO2, HbR, Mel, representa a distribuição de concentração do cromóforo por unidade de área.

Considerando que os pixels que compõem imagem são independentes, podemos representar sua posição por uma coordenada (x,y) e reescrever (4) como

$$-\log \vec{I}(x,y) = (M) \cdot \vec{C}(x,y) \quad \text{(A.5)}$$

Onde $\vec{I}(x,y)$ é o vetor das intensidades de cada pixel, $\mathbf{M}$ é a matriz constante com os coeficientes de extinção, $\vec{C}(x,y)$ é o vetor de concentração de cada um dos cromóforos em cada pixel. Os coeficientes de extinção são tabelados e foram selecionados em 600 nm, 540 nm e 440 nm [85].

B Artigo

Image haziness contrast metric describing optical scattering depth

André R. Vitor[a], Arie Shaus[b,c], George C. Cardoso[a]

a) Physics Department, FFCLRP, Universidade de São Paulo, Ribeirão Preto, 14040-901, Brazil

b) Department of Genetics, Harvard Medical School, Boston, MA, USA.

c) Jacob M. Alkow Department of Archaeology and Ancient Near Eastern Civilizations, Tel Aviv University, Tel Aviv, Israel.

***Abstract*— Contrast is not uniquely defined in the literature. In particular, there is a need for a contrast measure that scales linearly and monotonically with the optical scattering depth of a translucent scattering layer that covers an image. Here, we address this issue by proposing an image contrast metric, which we call haziness contrast metric, and experimentally test it using milk as a scattering medium to simulate a decline in image contrast. Compared to other contrast metrics in the literature, the proposed metric is the closest to linear, as a function of the increasing density of the scattering material on the image.**

Keywords*— *Image contrast, optical scattering.

I. Introduction

Contrast is not uniquely defined in the literature. The Michelson measure of contrast is commonly used for images with patterns, in which the bright and dark intensities occupy similar fractions of the image [1]. One of the oldest ways to calculate contrast is the Weber contrast, appropriate for large and constant backgrounds [2]. Another common way to define contrast is the root-mean-square (RMS) contrast [3], which is useful to compare the contrast between two images. RMS contrast is independent of spatial frequency nor spatial distribution [2]. Finally, there are contrast scales based on the shape of the image histogram spread, such as the scale described in [4].

Existing contrast scales successfully measure contrast for image optimization and image haziness removal [5]–[8], which require low levels of scattering, such as shown in the top row of Fig. 1. However, the literature is lacking a contrast metric that monotonically changes with the haziness or fogginess of an image, and is linear for a very large dynamic range of scattering depth.

Optical depth is a ratio related to the intensity of the incident to transmitted radiant flux. The optical depth is a monotonically increasing function of optical path length; it measures the attenuation of the transmitted radiant power in a material.

If the medium is air or water for example, light rays are absorbed and scattered by the medium and by materials dissolved in the medium. The absorption and scattering are both dependent on the wavelength and on the size of the particles in the water [9].

In this paper, we present a quantitative way of measuring contrast that is nearly linear for a large dynamic range of optical scattering depth. We use actual photographs where milk is added on the optical path to simulate a decline in image contrast (Fig. 1). This scale, which we call "Haziness scale" fulfills the aspects of linearity as a function of increasing density of the scattering material, and works well over a dynamic range wider than other contrast scales shown in the literature. Applications of the haziness scale include, for example, the study of optical coherence tomography (OCT), eye retina images, measurement of the amount of fat present in milk, and eye fundus photography. In this paper, we focus on the definition of the haziness scale, and comparison to metrics from the literature in a controlled environment.

This paper is organized as follows: first, we describe some metrics from the literature and how we used them in the images; in the($\boldsymbol{g}$)subsection we describe the proposed metric. In the next section, we describe how the pictures were taken. Finally, we compare the new metric with those from the literature and some applications.

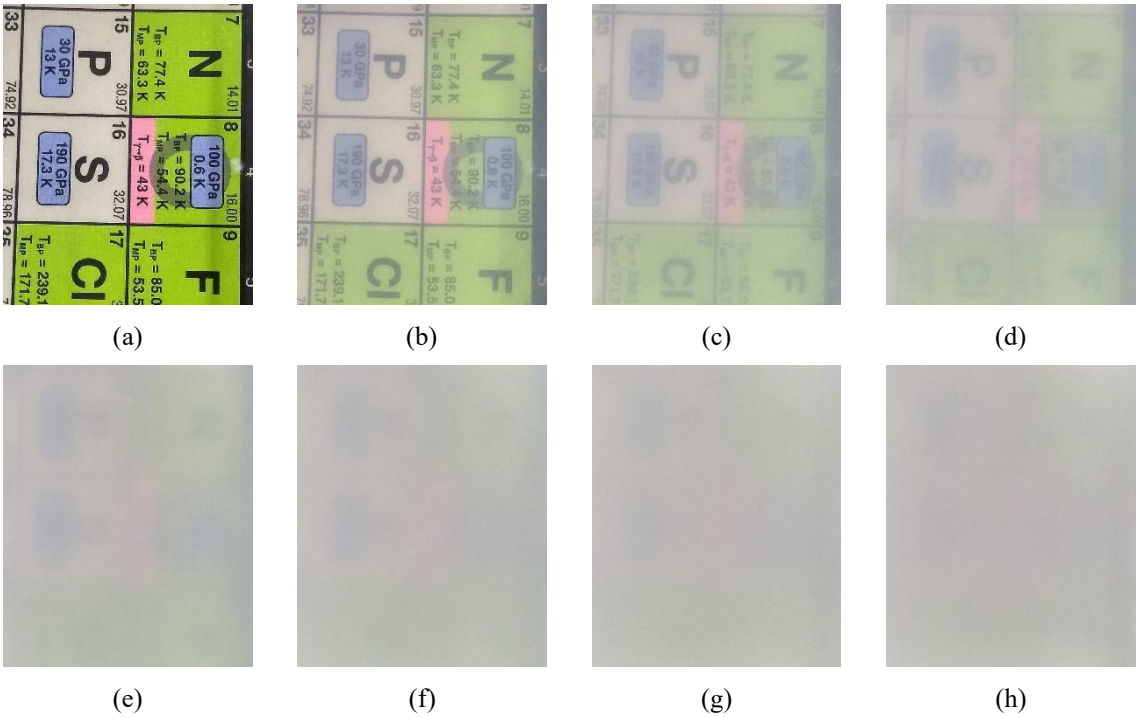


Fig. 1. Contrast worsening by the addition of milk in a transparent bowl filled with water, above an image. From left to right, first row: (a) just water, (b) 5 ml, (c) 10 ml, (d) 15 ml of milk. Second row: (e) 20 ml, (f) 25 ml, (g) 30 ml, (h) 35 ml of milk.

A. *Michelson Contrast*

Contrast ratio measure by Michelson [1], [2]

$$Michelson = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}, \tag{1}$$

where I_{max} and I_{min} are the maximal and minimal luminance values of the image. The Michelson contrast is a metric originally used for images with sinusoidal patterns and is a poor measure for complex images.

B. *Root Mean Square Contrast*

There are other contrast metrics, such as the root-mean-square contrast [3]:

$$RMS = \left[\frac{1}{n}\sum_{i=1}^{n} \left(x_i - \underline{x}\right)^2\right]^{\frac{1}{2}}, \tag{2}$$

where $x_i \in [0,1]$ is a normalized gray-level value, $\underline{x}$ is a normalized gray level $\underline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, and n is the number of pixels in the image. For color images, we separate the RGB channels and x_i represents one of these channels. The RMS metric has been related to human perception [10], [11] and is widely used as an image summary statistic.

C. *Histogram Spread*

Histogram Spread is defined as the interquartile range of the cumulative histogram divided by the pixel value range [4]: We first take the image's histogram and normalize such that the sum is 1. Next, we calculate

the positions of the 1st and 3rd quartiles of the cumulative histogram and take the difference from those positions. Histogram Spread is this difference divided by the pixel range, the difference between the highest and lowest possible intensity for the pixels

$$HS = \frac{Q_3 - Q_1}{p_{max} - p_{min}}, \tag{3}$$

where Q_n is the n-th quartile and p_{max} and p_{min} are the maximum and minimum values for the pixels, respectively. Histogram Spread has a range of values from 0 to 1.

D. Weber Contrast

Weber contrast [2] is one of the oldest contrast metrics, used to measure the contrast when there is a uniform background and a well-defined target: $Weber = (I - I_b)/I_b$, with I and I_b representing the luminance of the target and background, respectively. However, this is not a satisfactory global contrast measurement, since some very bright or dark spots would determine the contrast of the entire image. Therefore, we modified in an attempt to improve the measure for complex images, changing the denominator to the average luminance of the image, denoted by $\underline{I}$.

$$Weber = \frac{I - I_b}{\underline{I}}, \tag{4}$$

E. Rizzi

Their algorithm [12] estimates global and local components of contrast. The algorithm works as follows: First, it makes an under-sampling on the original image, then the under-sampled images are transformed to CIELab. After this step, it calculates the 8-Neighborhood local contrast for each pixel in the L channel, and finally calculates the sum of the averages of each under-sampled image to obtain a global measure.

II. HAZINESS SCALE: DEFINITION

Here we describe our proposed metric. In its essence, the Haziness scale compares normalized histograms of multiple blocks of the image, a pair at a time. The metric is inspired by the foreground to background histogram contrast as described by Shaus et al. [13]. One of the several differences here is that the two blocks, i and j, are at random positions in the image, and there is no need to manually select the foreground and background—since we assume the scattering medium covers the entire image.

The haziness contrast metric is calculated as follows. Two random image blocks, i and j, of $s \times s$ pixels are sampled. The $s \times s$ pixels image blocks have area-normalized histograms $\overrightarrow{H_i}$ and $\overrightarrow{H_j}$, respectively, where the vectors $\vec{H}$ represent the values of each image block. By area-normalized we mean that the sum of their entries is 1, that is, $\| \overrightarrow{H_i} \|_1 = 1$, where $\| \cdot \|_1$ is the taxicab l_1-norm [14]. Each histogram vector, $\overrightarrow{H_i}$ and $\overrightarrow{H_j}$, has 2^b entries, where b is the bit-depth of the image (e.g., 8-bit). With these definitions we have:

$$Haziness = \left\langle \frac{\left\| \overrightarrow{H_i} - \overrightarrow{H_j} \right\|_1}{\left\| \overrightarrow{H_i} + \overrightarrow{H_j} \right\|_1} \right\rangle_N, \tag{5}$$

where $\langle \cdot \rangle$ represents the average of N random pairs of blocks selection $N \gg 1$.

The number of patches N was chosen based on the convergence of the results. For N ~ 1000 the haziness metric starts to converge to a constant value for the image. In our analysis we used N = 10^4. The size s of the square must be small compared to the image's size. For a large s, in the order of the size of the chosen image, the image blocks are almost identical, since the intersection of the patches covers most of the image, resulting in similar histograms for the different patches, and the metric value will tend to zero. Measures at a granular

level (s < 10) provide a more monotonic behavior for the haziness metric as a function of increasing optical scattering. In our analysis we used s = 2.

III. Methods

Instead of using digital image processing techniques to change the image, such as gaussian blur, we decided to take a more physical/empirical approach. To test the performance of the metric, we simulated fogginess or haziness using water and milk. Starting with a clean image, we gradually added more milk, blurring the image.

Photographs were taken using a cell phone. The phone was positioned above a container filled initially with water and positioned above an image. A photo was taken for every 5 mL of milk added to the water container, with a pipette, up to 35 ml to simulate a linear decline in contrast, or similarly, an increase in the amount of fog in the image. The images have a size of 1188 by 1446 pixels and a resolution of 96 dpi, in jpeg format.

The conversion from RGB to grayscale was made using OpenCV, in which a transformation from RGB to Luma is made as $Y = 0.299 * R + 0.587 * G + 0.114 * B$. The Python code we built and examples are made available on GitHub https://github.com/Andre-Vitor/Haziness_paper.

IV. Results and Discussion

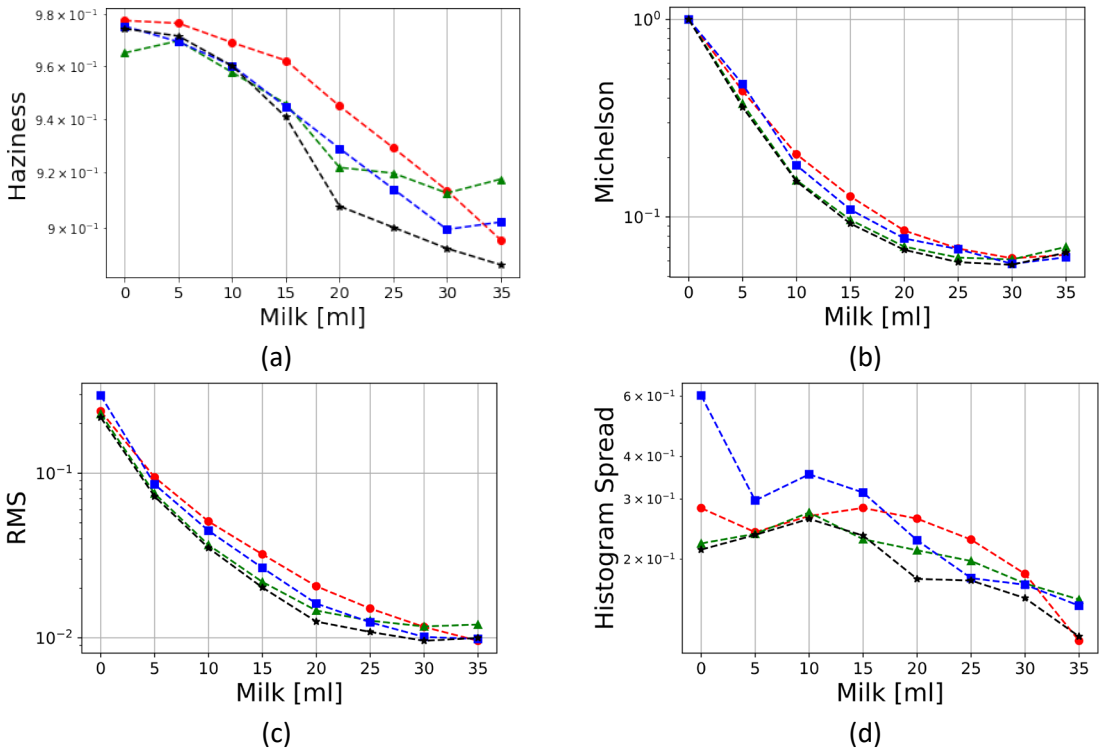


Fig. 2: Values in log scale for the (a) Haziness, (b) Michelson, (c)RMS, and (d) Histogram Spread metrics in each of the RGB channels and in grayscale. Red (denoted by circles), Green (denoted by triangles), Blue (denoted by squares), grayscale (denoted by stars). Notice that metrics (b) and (c) are monotonic but have poor discrimination power for high scattering depths. (d) shows a high optical scattering depth for the different colors, and despite poor monotonicity, correctly predicts that blue scatters more.

In this section, we focus on how the metrics correlate with optical scattering depth, for each RGB channel and in grayscale.

One can notice that the new metric is monotonic and quasi-linear as a function of increasing scattering medium concentration. None of the other metrics showed a perfectly linear behavior. Moreover, the Haziness measurement has a wider dynamic range. Disadvantages of the haziness scale include the existence of two tuning parameters: the number and size of patches.

In this study, we do not consider the polarization caused by scattering [15]. The novel metric's purpose is to quantify the contrast in the image, thus it does not identify the haze nor tries to de-haze the images, as some previous studies have done [5]–[8]. The scales were analyzed in two ways: in grayscale and RGB, with each channel treated separately. The measurements are shown in Fig. 2, where we used the log scale to better discriminate the scattering for high depths. The red, green, and blue colors represent the respective RGB channels, and the black line represents the grayscale measurements.

For the second and third graphs, and Fig. 2(c), we have the values for the Michelson metric [1], [2] in Fig. 2(b) and RMS metric in Fig. 2(c). The slope of the curve for low milk concentrations is high, we can clearly see a difference as milk is added to the bowl of water. However, from 20 mL and forwards, there are no significant differences, both for the RGB values and for the grayscale. The Michelson and RMS measurement is monotonic but has poor discrimination power for high scattering depths. The graphs for the Weber [2] and Rizzi [12] metrics were omitted, since their behavior is very similar to RMS and Michelson.

In the last graph in Fig. 2 are the values for the Histogram Spread metric. This metric has a non-linear behavior and converges to zero as the milk is added. The Histogram Spread measure shows good discrimination for the different colors and despite poor monotonicity, correctly predicts that blue scatters more. The measurements converge to zero, as the image becomes white.

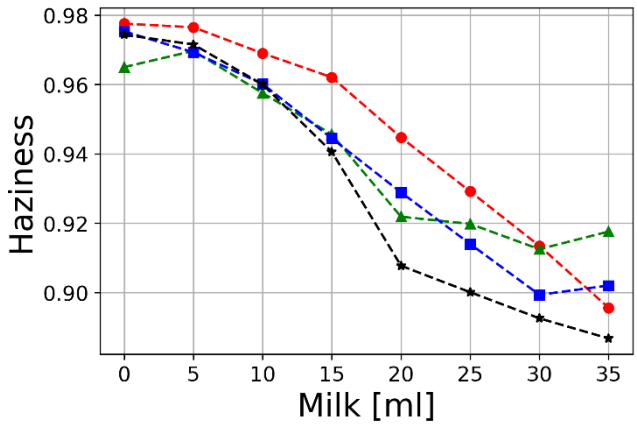


Fig. 3: Haziness metric values. The RGB channels are denoted by the circles, triangles, and squares, respectively. The grayscale is denoted by the star. The metric is monotonic in the R channel and for grayscale. Notice that the blue light displays a steeper increase in scattering with concentration, as compared to the red light, as expected.

The Weber, Michelson, and RMS metrics can detect large contrast variations when milk density is low, but as the density increases, these scales do not change too much. The Weber, Michelson, and RMS metrics also do not differentiate between the RGB channels. Alternatively, the proposed haziness metric, shown in Fig. 3. is much more linear and can distinguish the changes even with high densities of milk, as seen in Fig. 3. Depending on the amount of milk, it is possible to identify which color is most scattered. The haziness metric also has a wider dynamic range.

The values of the proposed metric are illustrated in Fig. 3. For the Haziness metric, in particular, it is necessary to choose the number of iterations that will be made. At each iteration, two small patches of the image are selected and compared as previously described. The final result is the average of all the measurements. In our analysis, we used $s = 2$ for the patches' size, this way the comparison of the histograms is local. Empirically this was the best value for the parameter. As the size of the patches increases, they become more

similar, diminishing the final value of the measurement, converging to zero as the patch size converges to the whole image.

For the grayscale, the haziness values are monotonic and quasi-linear, as the concentration of milk increases the range of color diminishes and the contrast declines. The blue channel is also quasi-linear, and the spread is bigger than the red channel, as expected, because the blue has a higher wavelength than red.

We interpret the scale as, the bigger the value, i.e., the closer it is to one, the better the image contrast. A possible interpretation for the colors is the amount of scattering of one color in the image. In Fig. 3, the blue color has lower values than red (except in the last measurement), since blue scatters more than red. The green color is scattered in a way similar to blue, but after a threshold, around 20 mL of milk, the scattering of green stabilizes. Since the ranges of the values of the metrics are different, we normalized the grayscale values to the interval [0,1] in Fig. 4. The curves shapes remain the same. The RMS, Michelson, Weber and Rizzi have similar nonlinear monotonic curves.

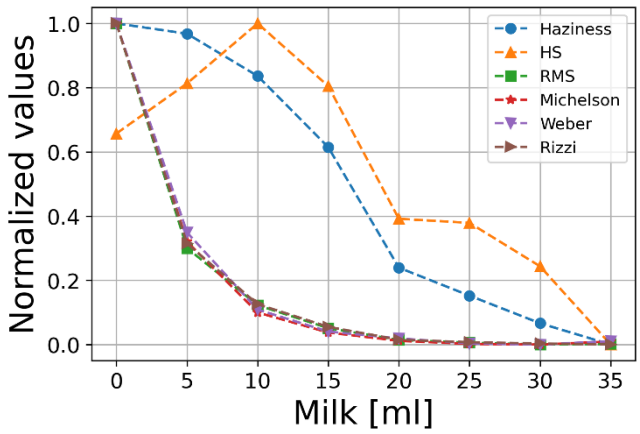


Fig. 7: Normalized metrics for grayscale images. The haziness (denoted by circles) is monotonic and is the closest to linear. The other metrics are Histogram Spread (HS, denoted by triangles), RMS (denoted by squares), Michelson (denoted by stars), Weber (denoted by inverted triangles), Rizzi (denoted by left triangles).

How do the metrics behave when we change the contrast and brightness and when we apply a histogram equalization?

In image processing, images are modified. To check whether the studied metrics are invariant under transformations.

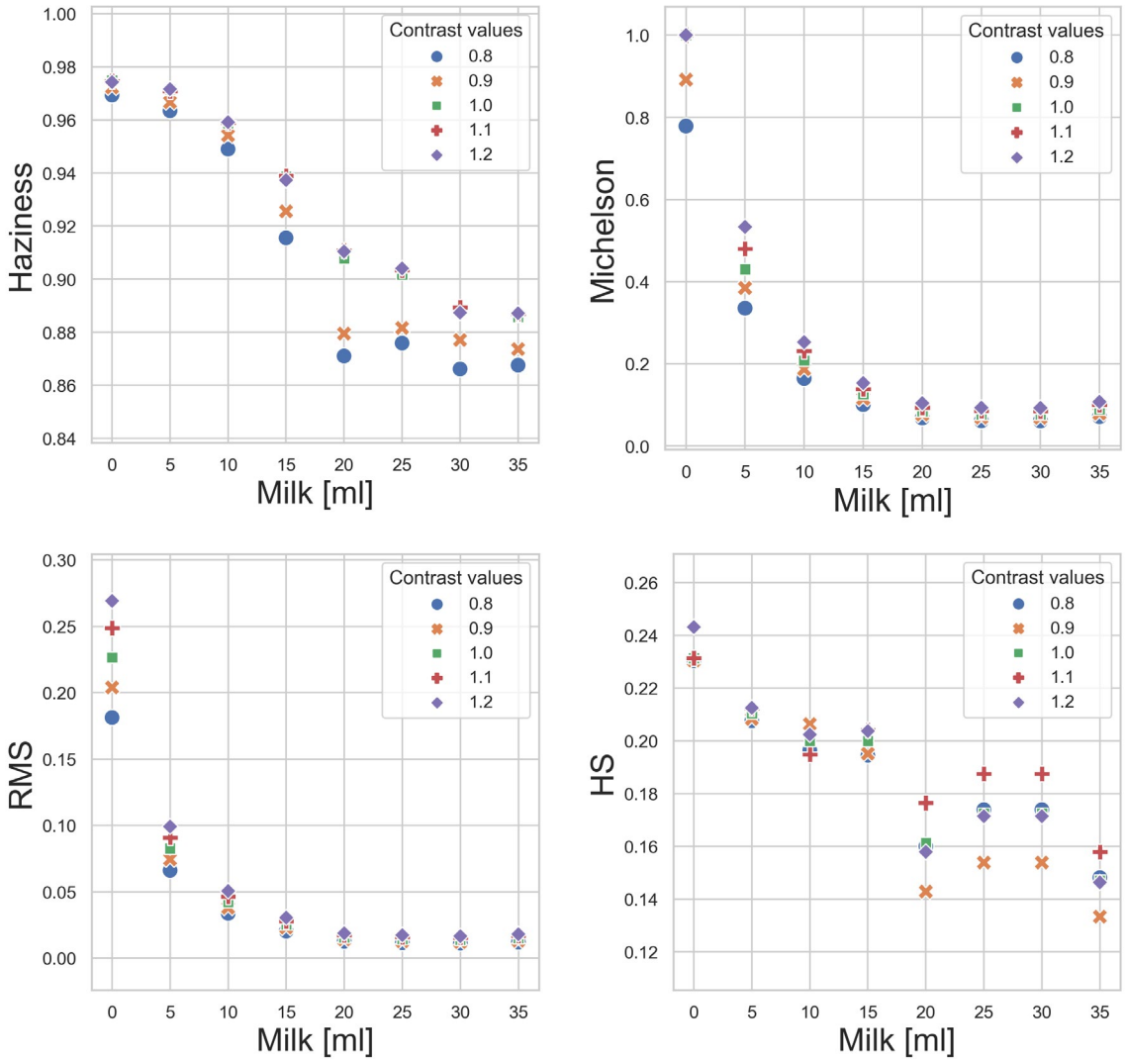


Fig. 4: Effect of contrast variation on various metrics: Haziness, Michelson, RMS and Histogram spread values. See text for contrast definition and modification methodology.

Fig. 4 shows the behavior of the metrics when we change the contrast. Michelson and RMS metrics vary linearly with the variation in contrast, higher contrast values translate in higher metric values and lower contrast values in lower metric values. Using the Pillow library in Python, we changed the contrast from [0.8 to 1.2] and the brightness from [0.8, 1.2]. An enhancement factor of 0.0 in contrast gives a solid grey image and an enhancement factor of 0.0 in brightness gives a black image. A factor of 1.0 gives the original image.

The Histogram Spread has a non-linear behavior. The Haziness metric is monotonic for contrast values ≥ 1, but is not monotonic for lower contrasts and high milk density $\geq 15\ ml$.

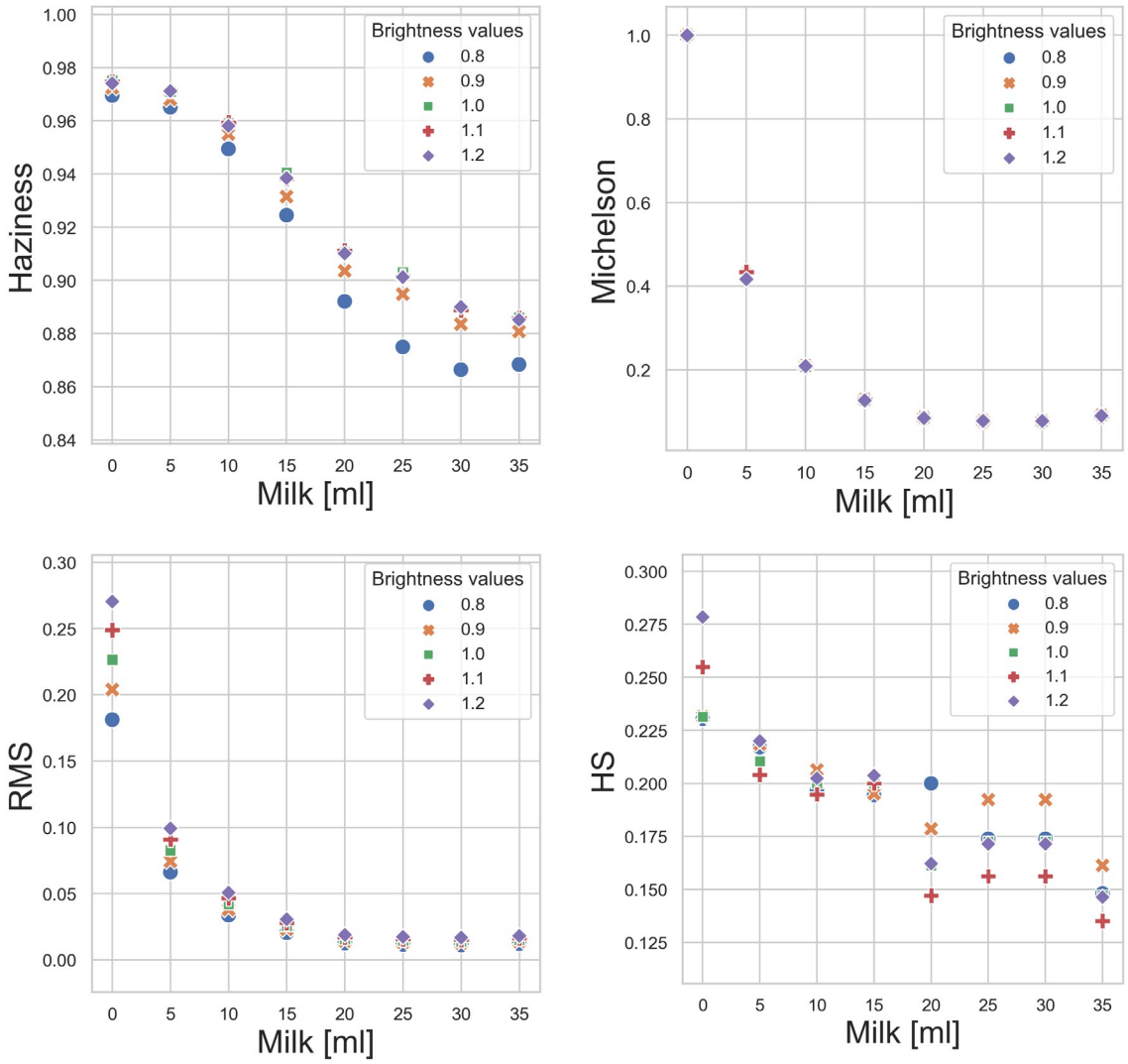


Fig. 5: Effect of brightness variation on various metrics: Haziness, Michelson, RMS and Histogram Spread.

Fig. 5 shows the behavior of the metrics when we change the brightness. The RMS metrics vary linearly with the variation in brightness, higher brightness values translate in higher metric values and lower brightness values in lower metric values. The Histogram Spread has a non-linear behavior. The Michelson metric is not affected by changes in brightness. The Haziness metric has higher variance in the values when the brightness is low and the density of milk is high, above 15 ml.

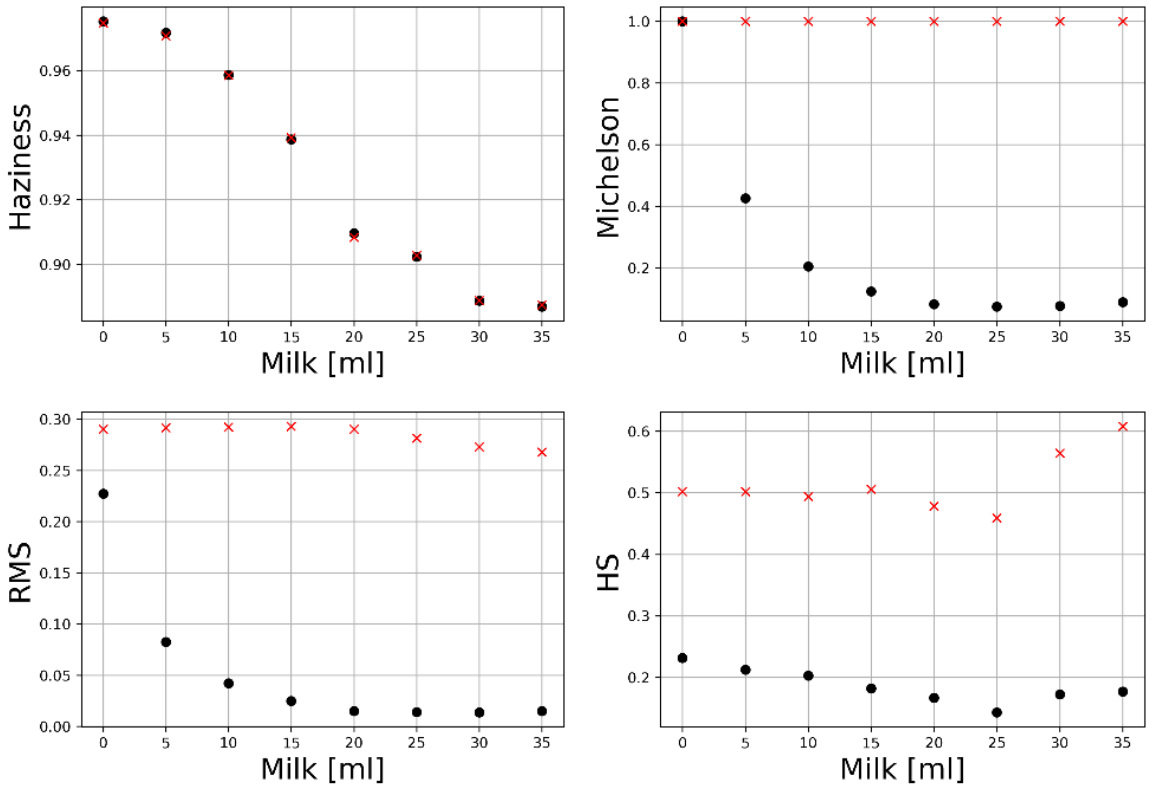
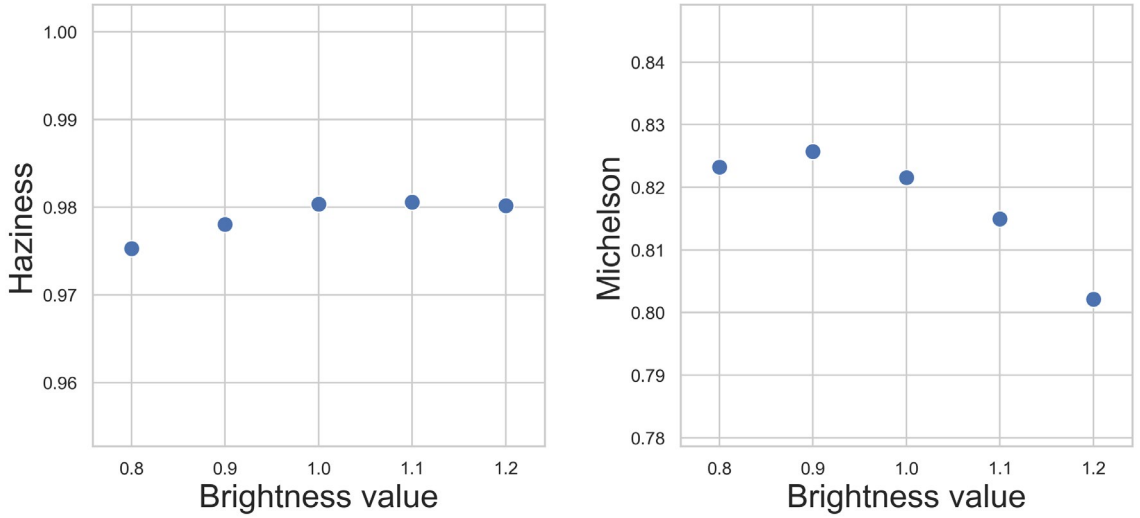


Fig. 6: Effect of histogram equalization on various metrics: Haziness, Michelson, RMS and Histogram Spread values for histogram equalization. Unprocessed (black circles) vs. histogram equalized images (red crosses).

Histogram equalization is a common and useful method in image processing of contrast adjustment. We applied the *equalizeHist* method of OpenCV to the metrics and observed their behavior. We observe from Fig 6, that the Haziness metric is robust with respect to histogram equalization, while Michelson, RMS and Histogram Spread metrics are equalization sensitive.



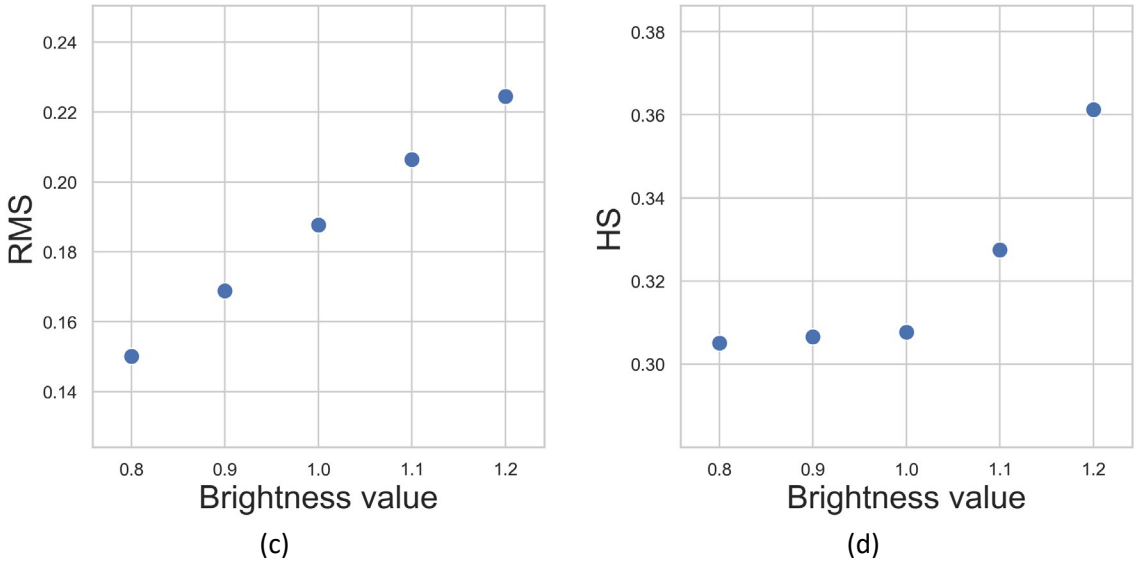
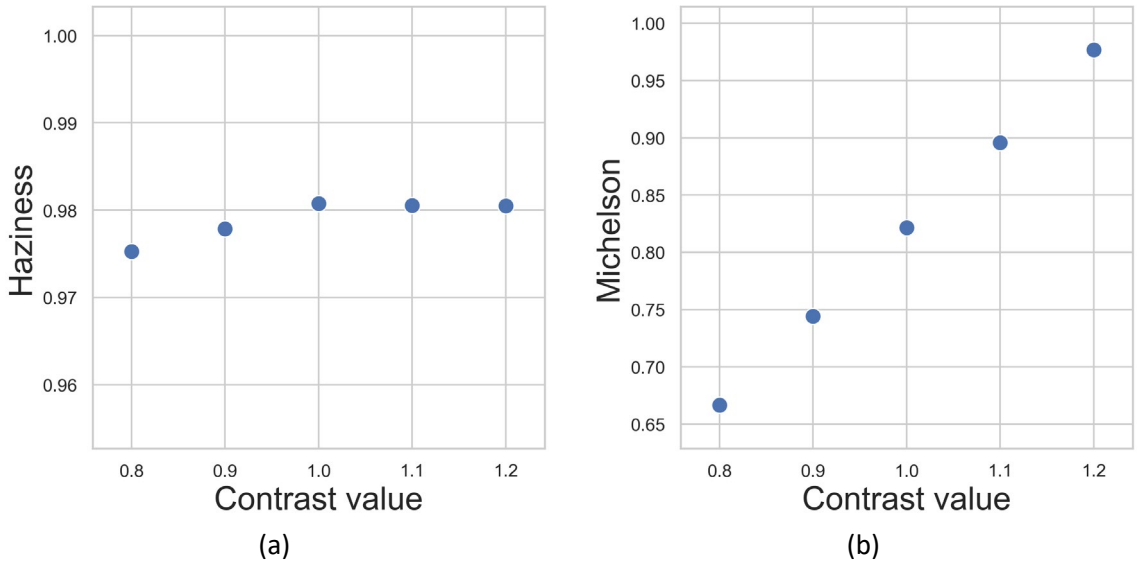


Fig. 7: Haziness, Michelson, RMS and Histogram Spread values for changes in Brightness values in the Lena image.



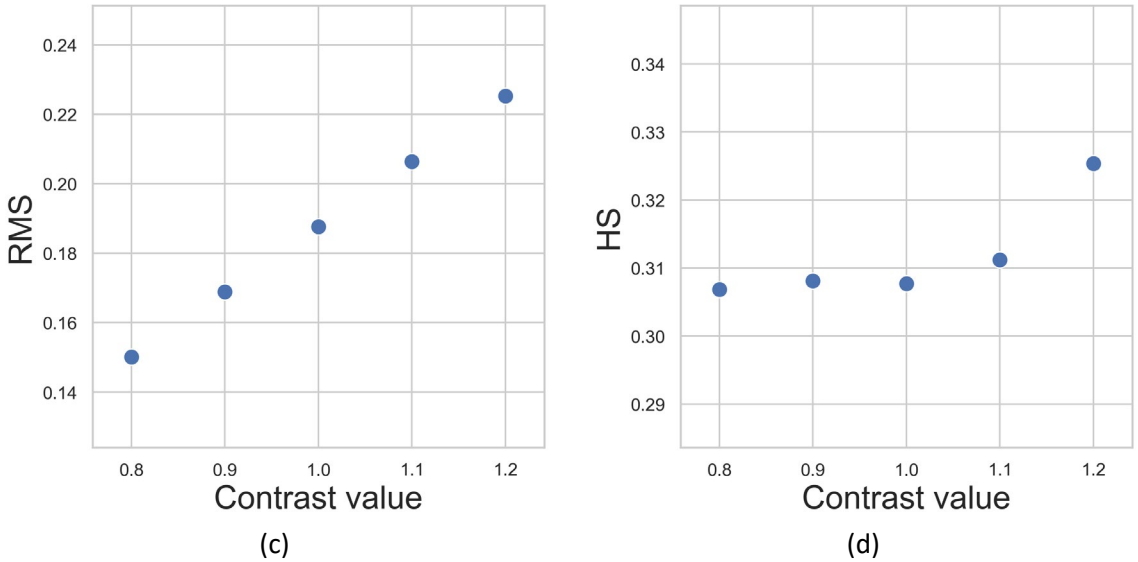


Fig. 8: Haziness, Michelson, RMS and Histogram Spread values for changes in Contrast values in the Lena image.

Figures 7 and 8 shows how a change in brightness and contrast affects the metrics, respectively. Haziness is vary less than the other metrics.

An interesting area of study in image processing is the dehazing of images. Removing haze can increase the visibility of the scene. We now compare two single image dehaze algorithms Fattal [16] and Berman [17], checking whether the algorithms improved or worsened the haziness value of the images, we compared the values of the original images and the dehazed ones.

Table 1: Haziness values for the original images and Berman and Fattal dehazing algorithms, with their respective p-values…. Whether the Haziness values for the output images increased or decreased after the dehaze algorithm, and their p-values. (+) if the value increased and (-) if the value decreased. Here the standard uncertainty is determined from 10^2 *runs of the haziness metric with* $N = 10^3$.

	Original	Std of the mean	Berman [17]	p-value	Fattal [16]	p-value
cityscape	0.967	0.0003	$0.983^{(+)}$	3.90E-11	$0.978^{(-)}$	8.20E-11
forest	0.974	0.0003	$0.979^{(+)}$	3.50E-04	$0.965^{(-)}$	2.10E-05
pumpkins	0.980	0.0002	$0.979^{(-)}$	0.27	$0.975^{(+)}$	3.50E-04
train	0.972	0.0003	$0.975^{(+)}$	1.60E-02	$0.971^{(-)}$	0.27

The values for Berman increased, except with pumpkins and the values for Fattal increased only in cityscape. But we can't say that the pumpkins for Berman or train for Fatall changed, within 95% C.I. Links to the images and algorithms: Fattal https://www.cs.huji.ac.il/%7Eraananf/projects/dehaze_cl/results/ and Berman https://openaccess.thecvf.com/content_cvpr_2016/html/Berman_Non-Local_Image_Dehazing_CVPR_2016_paper.html

Table 1 summarizes the results for the Berman and Fattal algorithms, with their respective standard deviations and standard uncertainties. It also shows whether the haziness values for the output images increased (upperscript +) or decreased (upperscript -). As argued in [17], the Fattal method leaves some haze and artifacts in the results, so generally the Berman non-local image dehazing method is usually better. Berman also has more improved values of haziness than Fattal.

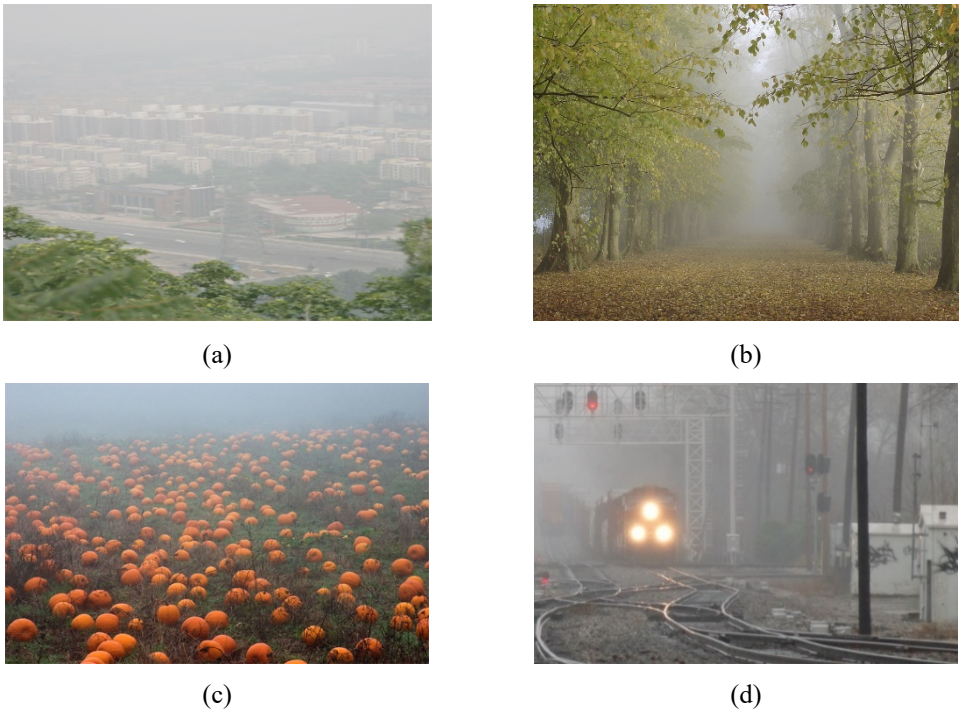


Fig. 9: Figures used to make the comparison of the de-haze algorithms. (a) cityscape, (b) forest, (c) pumpkins and (d) train.

V. CONCLUSION

The haziness metric developed is monotonic and closer to linear as a function of the optical scattering depth, compared with other metrics in the literature. It also has a wider dynamic range, being able to quantify haziness levels with 50% higher scattering depth. Finally, the haziness metric correctly predicts the correct order of scattering depth for the red, green, and blue channels of the RGB image. Another application of the metric is to compare the performance of dehazing algorithms.

REFERENCES

[1] A. A. Michelson, *Studies in optics*. Courier Corporation, 1995.

[2] E. Peli, “Contrast in complex images,” 1990.

[3] M. Pavel, G. Sperling, T. Riedi, and A. Vanderbeek, “Limits of visual communication: the effect of signal-to-noise ratio on the intelligibility of American Sign Language,” 1987.

[4] A. K. Tripathi, S. Mukhopadhyay, and A. K. Dhara, “Performance metrics for image contrast,” *ICIIP 2011 - Proceedings: 2011 International Conference on Image Information Processing*, no. Iciip, pp. 0–3, 2011.

[5] J. H. Kim, W. D. Jang, J. Y. Sim, and C. S. Kim, “Optimized contrast enhancement for real-time image and video dehazing,” *Journal of Visual Communication and Image Representation*, vol. 24, no. 3, pp. 410–425, 2013.

[6] R. T. Tan, “Visibility in bad weather,” *In Computer Vision and Pattern Recognition. CVPR 2008*, pp. 1–8, 2008.

[7] S. G. Narasimhan and S. K. Nayar, “Contrast restoration of weather degraded images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 713–724, 2003.

[8] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011.

[9] K. O. Amer, M. Elbouz, A. Alfalou, C. Brosseau, and J. Hajjami, “Enhancing underwater optical imaging by using a low-pass polarization filter,” *Optics Express*, vol. 27, no. 2, p. 621, Jan. 2019.

[10] P. J. Bex, S. G. Solomon, and S. C. Dakin, “Contrast sensitivity in natural scenes depends on edge as well as spatial frequency structure,” *Journal of Vision*, vol. 9, no. 10, pp. 1–19, 2009.

[11] P. J. Bex and W. Makous, “Spatial frequency, phase, and the contrast of natural images,” *Journal of the Optical Society of America A*, vol. 19, no. 6, p. 1096, 2002.

[12] A. Rizzi, T. Algeri, G. Medeghini, and D. Marini, “A proposal for contrast measure in digital images,” *CGIV 2004 - Second European Conference on Color in Graphics, Imaging, and Vision and Sixth International Symposium on Multispectral Color Science*, pp. 187–192, 2004.

[13] A. Shaus, S. Faigenbaum-Golovin, B. Sober, and E. Turkel, “Potential contrast - A new image quality measure,” *IS and T International Symposium on Electronic Imaging Science and Technology*, pp. 52–58, 2017.

[14] E. F. Krause, *Taxicab geometry: An adventure in non-Euclidean geometry*. Courier Corporation, 1986.

[15] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar, “Instant dehazing of images using polarization,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 325–332, 2001.

[16] R. Fattal, “Single image dehazing,” ACM Trans. Graph., vol. 27, no. 3, 2008.

[17] D. Berman, T. Treibitz, and S. Avidan, “Non-local Image Dehazing,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 1674–1682, 2016.