



•NOVA•
UCSAL

Universidade Católica do Salvador
Bacharelado em Engenharia de Software

Hugo Vinicius Carneiro
Lucas Pereira Santos

Detecção de Catarata por meio de imagens utilizando Redes
Neurais Convolucionais

Salvador

2020

Hugo Vinicius Carneiro
Lucas Pereira Santos

**Detecção de Catarata por meio de imagens utilizando
Redes Neurais Convolucionais**

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador, como parte dos requisitos necessários para a obtenção do Título de Engenheiro de Software.

Orientador: Prof. Me. André Brasil Vieira Wyzykowski

Universidade Católica do Salvador

Salvador
2020

Hugo Vinicius Carneiro
Lucas Pereira Santos

Detecção de Catarata por meio de imagens utilizando Redes Neurais Convolucionais

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador, como requisito parcial para a obtenção do título de Engenheiro de Software.

Comissão Examinadora

Prof. Me. André Brasil Vieira Wyzykowski
Universidade Católica do Salvador
Orientador

Prof. Me. Arnaldo Bispo de Jesus
Universidade Católica do Salvador

Prof. Me. Semiramis Ribeiro de Assis
Universidade Católica do Salvador

Salvador, 28 de janeiro de 2021

Dedico este trabalho a todos que nos apoiaram e nos incentivaram até aqui.
Principalmente, aqueles que foram motivadores para que não desistíssemos,
impulsionadores dos nossos sonhos, nossos pais.

Agradecimentos

Agradecemos, primeiramente, à Deus, por ter nos dado força para lutarmos. Aos nossos pais, por serem nossa referência, estando sempre ao nosso lado nos momentos felizes ou difíceis. Aos nossos familiares e também a aqueles que fizeram parte da nossa trajetória acadêmica. Assim, agradecemos também ao nosso orientador, por mediar um amplo conhecimento para nossa formação.

"O sucesso é ir de fracasso em fracasso sem perder o entusiasmo."
(Winston Churchill)

Resumo

A catarata é uma das doenças do globo ocular que mais gera cegueira no mundo. A detecção rápida dessa doença, rente ao tratamento adequado, contribui para a melhora da qualidade de vida dos pacientes. Este trabalho utiliza as redes neurais convolucionais para treinar três conjuntos de dados contendo imagens de diferentes pacientes, com o objetivo de classificar os portados de catarata e os não portadores. Para isso, a rede neural convolucional vgg19 teve sua arquitetura modificada e melhorada. O resultado mais próspero obtidos pela rede, foi uma acurácia de 100%.

Palavras-Chave: Aprendizagem Profunda. Rede Neurais Convolucionais. Catarata.

Abstract

Cataract is one of the diseases of the eyeball that generates the most blindness in the world. The rapid detection of this disease, close to the appropriate treatment, contributes to the improvement of the patients' quality of life. This work uses convolutional neural networks to train three sets of data containing images of different patients, in order to classify cataract patients and non-carriers. For this, a convolutional neural network vgg19 had its architecture modified and improved. The most successful results used by the network, was an accuracy of 100 %.

Keywords: Deep Learning. Convolutional Neural Network. Cataract.

Lista de figuras

Figura 1 – Esquema do Sistema Visual mostra o olho humano e sua estrutura interna com córnea, retina, cristalino, e sua conexão com o cérebro. Fonte: (BACKES A. R; JUNIOR, 2016)	19
Figura 2 – Esquema do sistema de visão computacional, com as fases de aquisição, processamento, segmentação, extração de características e reconhecimento de padrões. Fonte: (BACKES A. R; JUNIOR, 2016)	21
Figura 3 – Neurônio biológico: Fonte: (BARRETO, 2002)	21
Figura 4 – A imagem mostra um sinal x na entrada da sinapse j conectado a o neurônio k é multiplicado pelo peso sináptico w_k , Um somador para somar os sinais de entrada, uma função de ativação que limita a faixa de amplitude do sinal de saída, também inclui uma polarização chamada b_k , o <i>bias</i> b_k tem o efeito de aumentar ou diminuir a entrada a da função de ativação, dependendo se é positivo ou negativo, respectivamente. Fonte: (HAYKIN, 1999)	22
Figura 5 – Arquitetura de uma rede neural com imagens de entrada e suas principais camadas: convolucionais, <i>pooling</i> e totalmente conectadas. E por fim a saída. Fonte: (CARNEIRO A. C.; SILVA, 2017)	24
Figura 6 – Matriz bidimensional, com altura e largura e profundidade. Fonte: (ALVES, 2018)	24
Figura 7 – Operação de convolução em uma entrada e um filtro de 3X3. Fonte: (USHIZIMA, 2017)	25
Figura 8 – A imagem corresponde a um passo unitário e a um passo igual a 2. Fonte: (CARNEIRO A. C.; SILVA, 2017)	26
Figura 9 – Efeito da camada de <i>pooling</i> . Fonte: (KHAN, 2018)	26
Figura 10 – A figura mostra uma unidade de <i>pooling</i> gerando a ativação máxima na região de entrada 2×2 . Fonte: (NIELSEN, 2016).	27
Figura 11 – Interface gráfica do google colab, saída de dados	30
Figura 12 – Imagens do conjunto de dados proposto neste trabalho	32
Figura 13 – A imagem mostra que na linha 3 o paciente é diagnosticado com duas doenças, porém só uma foi rotulado.	33
Figura 14 – Imagens do conjunto de dado, com imagens de treino com as classes normal e catarata.	33
Figura 15 – Matriz de confusão, a classe 0 (normal) e a classe 1 (catarata).	38
Figura 16 – A imagem mostra dados do treino e validação logo das épocas.	38
Figura 17 – Matriz de confusão, a classe 0 (normal) e a classe 1 (catarata).	39

Figura 18 – A imagem mostra dados do treino e validação logo das épocas.	40
Figura 19 – Matriz de confusão, a classe 0 (normal) e a classe 1 (catarata).	41
Figura 20 – A imagem mostra dados do treino e validação logo das épocas.	41
Figura 21 – Imagens de validação classificadas corretamente pelo modelo.	42
Figura 22 – Imagem de validação normal, classificada corretamente pelo modelo com normal.	43
Figura 23 – Imagem de validação normal predita pelo modelo erroneamente como catarata.	43
Figura 24 – Imagem de validação classificada como normal, predita corretamente pelo modelo como normal.	44
Figura 25 – Imagem de validação classificada incorretamente pelo modelo, é normal, porém ele predisse como catarata.	44

Lista de tabelas

Tabela 1	– Exemplo de uma matriz de confusão.	28
Tabela 2	– informações do conjunto de dados.	32
Tabela 3	– Arquitetura da Rede, com todas a sua parte, entrada, camadas convu- lucionais, max-polling, dropout, e saidas.	35
Tabela 4	– A tabela contem o relatório de classificação com as métricas acurácia, precisão, cobertura, F1 e suporte.	37
Tabela 5	– A tabela contem o relatório de classificação com as métricas acurácia, precisão, cobertura, F1 e suporte.	39
Tabela 6	– A tabela contem o relatório de classificação de acordo com as métricas acurácia, precisão, cobertura, F1 e suporte.	40
Tabela 7	– resultados sobre a rede e dos conjuntos de dados treinados	42

Lista de Siglas e Abreviaturas

AM *Aprendizado de Máquina*

IA *Inteligência Artificial*

RN *Rede Neural*

RNC *Rede Neural Convolucional*

Sumário

1	INTRODUÇÃO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Visão Humana	18
2.2	Doença da Visão	19
2.2.1	Catarata	19
2.3	Visão Computacional	20
2.4	Rede Neural Artificial	21
2.4.1	Neurônio Biológico e Artificial	21
2.4.2	Função De Ativação	22
2.4.3	Dropout	23
2.4.4	Aprendizado de Maquina	23
2.5	Rede Neural Convolutacional	23
2.5.1	Entrada	24
2.5.2	Camada Convolutacional	25
2.5.3	Stride	25
2.5.4	Camada de Pooling	26
2.5.5	Camada Totalmente Conectada	27
2.6	Métricas	27
2.6.1	Matriz de Confusão	27
2.6.2	Acurácia	28
2.6.3	Precisão	28
2.6.4	Cobertura	28
2.6.5	Log Loss	28
3	DESENVOLVIMENTO	30
3.1	Ferramentas	30
3.2	Conjunto de Dados	31
3.2.1	Pré-processamento	33
3.3	Arquitetura da Rede Neural Convolutacional	34
4	EXPERIMENTO E RESULTADOS	36
4.1	Experimento	36
4.2	Resultados	36
4.2.1	Retina dataset	37
4.2.2	ODIR	39

4.2.3	Conjunto de dados proposto	40
4.2.4	Análise dos Resultados	42
5	CONCLUSÕES	45
5.1	Trabalhos futuros	45
	REFERÊNCIAS	47

1 Introdução

A catarata está entre as três doenças que mais causam cegueira no mundo. De todas as doenças oculares existentes, apenas a catarata e o erro refrativo não corrigido são responsáveis por quase 74,8% de todos os casos de cegueira do planeta (OTTAIANO, 2019).

Ainda segundo OTTAIANO (2019), no Brasil, a catarata acomete pessoas de todas as idades. No entanto, cerca de 73,3% das pessoas acometidas pela catarata são idosos com idades acima de 75 anos. A elaboração deste trabalho foi pensada para rapidez do diagnóstico dessa doença que, além de diminuir a qualidade de vida das pessoas, causa cegueira em indivíduos.

A maneira mais eficiente para combater a catarata, até o momento, é a intervenção cirúrgica. Para o sucesso dessa intervenção, ou de qualquer outra, é preciso que haja um diagnóstico preciso e rápido. Para isso, muitas técnicas de análise de imagens podem ser usadas. Porém, as que têm dado melhores resultados são aquelas que usam a visão computacional.

A visão computacional é uma área da inteligência artificial (IA), que tem como objetivo fazer as máquinas enxergarem melhor, capturando informações de vídeos e imagens em que a visão humana não consegue. Com isso, torna-se um grande aliado da área de saúde, principalmente no campo de diagnóstico de doenças através de imagens (MILANO D ; HONORATO, 2010).

O objetivo desse trabalho é desenvolver um método que identifique a doença catarata por meio de imagens do fundo dos olhos, capturadas por diferentes lentes, ou seja, o sistema deverá ser capaz de classificar os pacientes em normais (sem doença) ou com catarata. Para isso, foram feitas pesquisas sobre o funcionamento do olho e catarata; os dados foram classificados, a aplicação de uma rede neural convolucional (RNC) foi modelada; foram utilizados dois conjuntos de dados diferentes no treinamento e validação da RNC; por fim, os resultados obtidos foram analisados.

Esse trabalho está organizado da seguinte forma: O capítulo 2 dispõe o referencial teórico, que apresenta os conceitos da visão humana, doença da visão, visão computacional, redes neurais artificiais (RNA) e redes neurais convolucionais (RNC), que são de extrema importância para o entendimento do trabalho. O capítulo 3 descreve a metodologia teórica e prática desenvolvida neste trabalho. O capítulo 4 apresenta os testes, resultados e análise. Por fim, no capítulo 5 apresenta a conclusão e sugestões para trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo serão mostrados os conceitos e técnicas utilizadas. Serão apresentados conceitos sobre a visão humana, doenças da visão, visão computacional e as técnicas de: redes Neurais Artificiais (RNAs), Redes Neurais Convolucionais (RNCs) e Aprendizado de Máquina (AM). Primeiramente, faz-se importante discutir a visão humana.

2.1 Visão Humana

De todos os nossos sentidos, a visão é o maior responsável pela forma que percebemos o meio em que estamos inseridos. Cerca de 75% do que percebemos é por meio da visão. Interação social, estabelecimento de padrões estéticos, comunicação, aprendizagem, etc. a visão perpassa nossas estruturas sociais. Tudo começa quando os olhos recebem informações luminosas e se utiliza de suas estruturas internas para transformar a luz em impulsos que são enviados para o cérebro (RAMOS, 2016).

Nesse sentido, entende-se que:

"Primeiramente, luz refletido pelos objetos passa pela córnea e pupila, que é um orifício que regula a luz nos olhos por meio da variação do seu diâmetro. A seguir, a luz prossegue pelo cristalino que funciona como uma lente biconvexa que focaliza a luz na retina. Alterações no cristalino causam problemas como miopia, hipermetropia, presbiopia etc. A retina, por sua vez, é uma camada de tecido nervoso no fundo dos olhos constituída por dois tipos de fotorreceptores: bastonetes e cones. Os bastonetes são células com alta sensibilidade à iluminação, pouca acuidade (ou seja, fornecem imagens de baixa resolução) e que não reconhecem diferença de cor" (BACKES A. R; JUNIOR, 2016).

Na próxima etapa, segundo BACKES A. R; JUNIOR (2016), os cones e bastonetes captam a luz, a transformando em um sinal nervoso que é enviado para o cérebro, onde a memória da imagem é gravada. Este processo da visão humana está ilustrado na Figura 1.

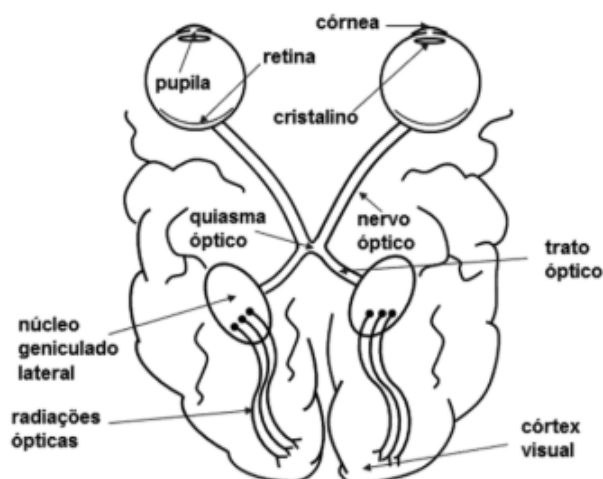


Figura 1 – Esquema do Sistema Visual mostra o olho humano e sua estrutura interna com córnea, retina, cristalino, e sua conexão com o cérebro. Fonte: (BACKES A. R.; JUNIOR, 2016)

A visão humana é um dos sistemas sensorial mais complexo que conhecemos, em conexão com o cérebro forma e processa imagens (HARLEY, 2003).

2.2 Doença da Visão

O olho é formado por um conjunto de estruturas que são responsáveis pelo processo de enxergar mostrado na Figura 1. Qualquer ruptura nessas estruturas causa danos para a saúde do indivíduo e para a forma que ele enxerga e interage com o mundo.

Na Seção 2.2.1 explicamos sobre a doença catarata e suas implicações para aqueles que a contrai, principalmente os mais idosos.

2.2.1 Catarata

Catarata é uma doença da visão que causa opacidade no cristalino, deixando a visão embaçada, diminuindo, assim, a acuidade visual (MENEZES, 2016). É uma doença inevitável e pode levar a cegueira, por isso, sua cirurgia é recomendada no mundo todo (TEMPORINI, 1997).

Geralmente, a catarata se desenvolve ao longo de anos, porém, a catarata congênita (casos raros) acomete também crianças, em consequências de outras doenças (TARTARELLA M, 1995).

A catarata aumenta o risco de queda para os idosos acima de 70 anos, que são 70% dos atingidos. Por isso, Passos (2008) realça que a visão é um sentido importantíssimo na interação do ser humano com o meio que o cerca. Sendo assim, problemas relacionados a visão podem comprometer a qualidade de vida do indivíduo. Para o diagnóstico da doença é preciso a realização de alguns exames, entre eles estão:

- **Teste de acuidade visual:** É o teste que analisa a capacidade de uma pessoa identificar contornos e formas.
- **Exame de lâmpada de fenda:** Também conhecido como microscópio, consegue exibir os olhos por dentro e por fora em 3 dimensões.
- **Exame de retina:** Esse exame permite que o médico analise o fundo do olho.

Os avanços tecnológicos têm reduzido o número de complicações por catarata, no entanto, ainda são necessários diagnósticos e tratamentos mais rápidos e precisos (MERULA R.V; FERNANDES, 2005).

2.3 Visão Computacional

A visão computacional é uma área da IA que se utiliza de algoritmos na tentativa de simular, de maneira cada vez mais eficiente, a visão (BACKES A. R; JUNIOR, 2016).

Assim como a visão humana, a visão computacional recebe como entrada uma imagem, porém a sua saída nem sempre é uma apresentação total da imagem. A intermediação entre a entrada e a saída é feita por um algoritmo (BALLARD D, 1982).

Segundo BACKES A. R; JUNIOR (2016), a visão computacional é constituída por algumas fases, cada autor denominam de maneira diferente. Barks as classifica assim:

- **Aquisição** - Como o nome indica, essa é a etapa de captação de imagens. Para isso podem ser usados celulares, câmeras digitais, aparelhos de ultrassonografia e raio x, satélites, microscópios e aparelhos de tomografia etc.
- **Processamento de Imagens** - Essa fase configura o melhoramento da imagem, considerando a retirada de ruído, suavização, detecção de bordas, retirada de sombras, etc.
- **Segmentação** - É a fase em que é feita a separações nas imagens, geralmente para pegar a parte interessada, contar quantidade de objetos ou tirar o fundo das imagens.
- **Extração de Característica** - Às vezes, mesmo depois que a imagem é segmentada, tem-se a necessidade de diminuir o número de características das imagens, reduzindo o número de *pixels*.
- **Reconhecimento de padrões** - Nessa fase as imagens são classificadas de acordo com suas características.

A Figura 2 mostra os passos de um esquema de visão computacional.

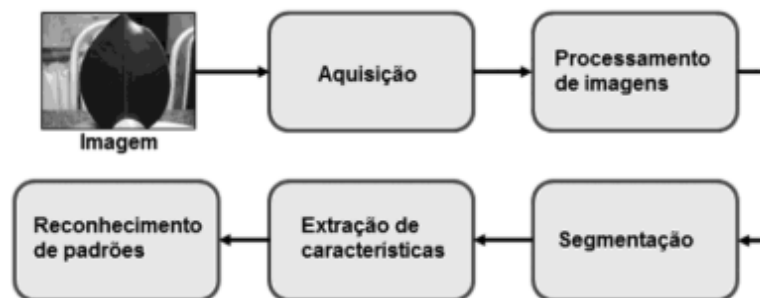


Figura 2 – Esquema do sistema de visão computacional, com as fases de aquisição, processamento, segmentação, extração de características e reconhecimento de padrões. Fonte: (BACKES A. R; JUNIOR, 2016)

2.4 Rede Neural Artificial

As redes neurais foram criadas baseadas no funcionamento do cérebro, pensando na maneira com que ele interage e responde ao ambiente, de acordo com suas experiências e funcionamento de seus neurônios (HAYKIN, 1999). Nesse sentido, a rede neural artificial é composta por conexões de neurônios que recebem entradas e, baseado em alguma regra, apresenta uma saída. Essas conexões entre neurônios artificiais é o que dá origem a arquitetura da rede neural.

2.4.1 Neurônio Biológico e Artificial

O cérebro humano é muito complexo. A ciência, apesar de ter avançado muito no estudo do seu funcionamento, ainda está longe de uma compreensão totalitária. O que se sabe é que o cérebro é composto por bilhões de neurônios que recebem e enviam impulsos neurais entre si, criando assim o que conhecemos por rede neural biológica, responsável pelo aprendizado.

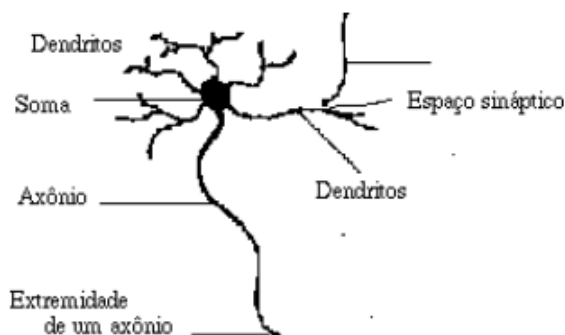


Figura 3 – Neurônio biológico: Fonte: (BARRETO, 2002)

Para BARRETO (2002), um neurônio é constituído por um corpo celular ou soma, por dendritos que enviam estímulos externos para o corpo celular e por axônios, que é

responsável pela comunicação com outros neurônios.

Para além do campo biológico, analisando o funcionamento do neurônio artificial pelo ângulo matemático, entende-se que: um neurônio é composto por sinais de entrada, que é multiplicado pelo peso sináptico, por um somador que soma o produto dos sinais de entrada e por uma função de ativação que é responsável pela saída do neurônio (HAYKIN, 1999).

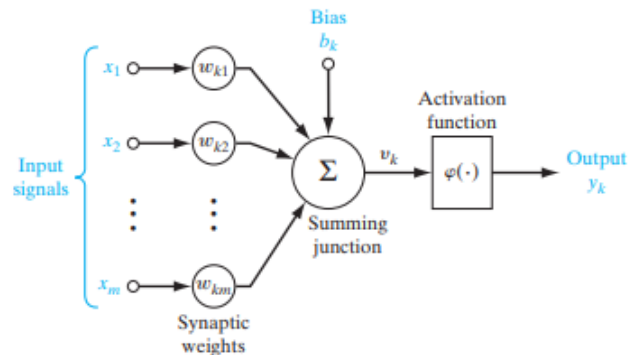


Figura 4 – A imagem mostra um sinal x na entrada da sinapse j conectado a o neurônio k é multiplicado pelo peso sináptico w_k , Um somador para somar os sinais de entrada, uma função de ativação que limita a faixa de amplitude do sinal de saída, também inclui uma polarização chamada b_k , o *bias* b_k tem o efeito de aumentar ou diminuir a entrada a da função de ativação, dependendo se é positivo ou negativo, respectivamente. Fonte: (HAYKIN, 1999)

As Funções de cada elemento do neurônio artificial são: sinal de entrada: (*input signals*), que refere-se aos dados que serão recebidos pela rede neural no momento do treinamento; pesos sinápticos (*synaptic weights*), em que armazenam o conhecimento adquirido, quando a rede recebe os sinais de entrada é feita associação a um peso diferente; a função de ativação (*activation function*), que é quem decide se o neurônio será ativado ou não e de que forma ele será ativado; somatório (*summing junction*), que soma todas as entradas e multiplica pelos pesos sinápticos; e por fim, saída (*output*) que condiz com a saída calculada pelo neurônio.

2.4.2 Função De Ativação

Além de ser responsável pela ativação dos neurônios, a função de ativação faz com que pequenas variações no peso de entrada causem pequenas variações no valor de saída dos neurônios. Existem várias funções de ativação, nesse trabalho serão utilizadas a ReLU (*Rectified Linear Units*) e a *sigmoide*, que já foram largamente testadas, hoje são as mais utilizadas na resolução de problemas usando redes neurais convolucionais.

2.4.3 Dropout

Essa técnica é responsável pela desativação de uma porcentagem dos neurônios a cada rodada, com o intuito de generalizar o modelo. Também causa uma interdependência entre os neurônios no momento do treinamento, em que os neurônios presentes terão que fazer as previsões dos que foram desativados temporariamente, evitando dessa forma overfitting (CARNEIRO A. C.; SILVA, 2017).

2.4.4 Aprendizado de Máquina

Aprendizado de máquina se refere a metodologia usada para que as máquinas aprendam a partir dos dados de entrada. Após ter aprendido, a máquina deve executar uma tarefa desconhecida com precisão (MEHRYAR, 2012).

O aprendizado supervisionado é uma categoria do aprendizado de máquina, em que os parâmetros são configurados antes da execução. No aprendizado supervisionado, os dados de entrada e saída já são conhecidos pelo instrutor (professor). O instrutor é o responsável por configurar os parâmetros e fornecer para o algoritmo os dados e o resultado esperado. Um exemplo disso é a classificação de imagens, em que é passado para o algoritmo um conjunto de imagens já rotuladas, com o intuito que ele aprenda baseado nas características dessas imagens e retorne como saída sua classificação.

2.5 Rede Neural Convolutiva

Na visão computacional, as redes neurais convolucionais é a técnica mais utilizada no reconhecimento de imagem e processamento de vídeos. Desde que foi proposta por Yann LeCun (LECUN (1998)) as CNN tornou-se cada vez mais complexa e eficiente.

Segundo KHAN (2018), uma diferença fundamental das redes neurais convolucionais em comparação com as outras redes neurais, é que para cada camada a rede vai aprendendo cada vez mais.

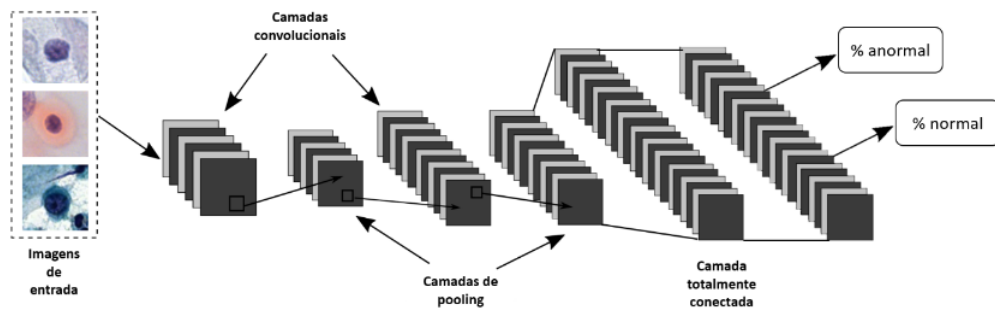


Figura 5 – Arquitetura de uma rede neural com imagens de entrada e suas principais camadas: convolucionais, *pooling* e totalmente conectadas. E por fim a saída. Fonte: (CARNEIRO A. C.; SILVA, 2017)

2.5.1 Entrada

Uma imagem para um computador é uma matriz, suas posições formam um conjunto de *pixels*, em que cada pixel representa um número inteiro. Nas imagens coloridas, dentro de cada pixel existem três canais chamados de RGB, essas imagens são conhecidas por imagens de cores reais e sua representação são três matrizes diferentes (BARELLI, 2018).

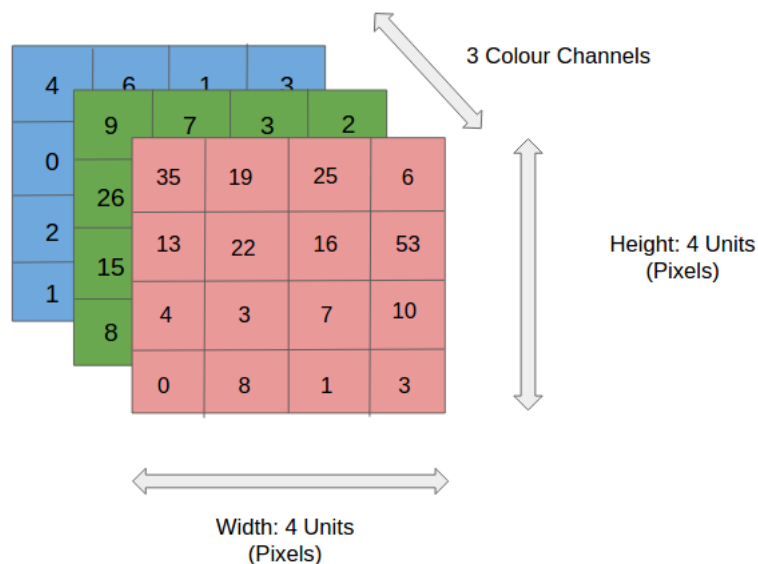


Figura 6 – Matriz bidimensional, com altura e largura e profundidade. Fonte: (ALVES, 2018)

Por exemplo, usando imagens de 28x28x3 (altura, largura e os canais de cores RGB) a RNC terá 784 (=28x28) neurônios de entrada sendo a saída a identificação da imagem.

2.5.2 Camada Convolutiva

As convoluções são uma espécie de filtro, ou kernel, que percorre toda a imagem, capturando suas características mais marcantes. O kernel é utilizado para operação de multiplicação de matrizes. Tal operação é usada várias vezes em partes diferentes da imagem e o somatório do produto dos filtros em cada parte da imagem é chamado de operação de convolução (USHIZIMA, 2017).

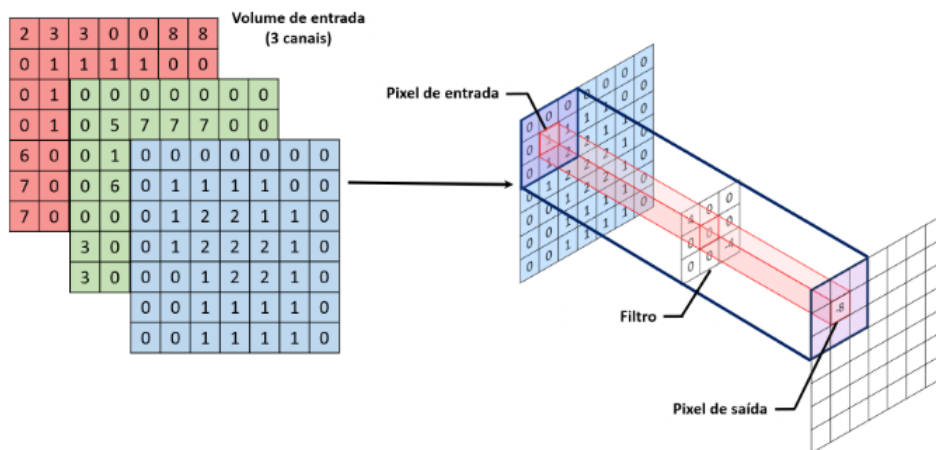


Figura 7 – Operação de convolução em uma entrada e um filtro de 3X3. Fonte: (USHIZIMA, 2017)

2.5.3 Stride

A ação de usar filtros para percorrer toda imagem vista na Seção 2.5.2 é feita de forma sequencial, em passos chamados de *stride*, passando por cada um dos *pixels* da imagem, tratando a como uma matriz. Segundo CARNEIRO A. C.; SILVA (2017), quando passo (*Stride*) é igual a 1, a altura e a largura da camada de saída será igual a 1. Se for igual 2, a saída será metade do tamanho da entrada. A Figura 8, contém a ilustração de como o passo influencia no deslocamento de um filtro 3x3 em duas etapas sucessivas da convolução.

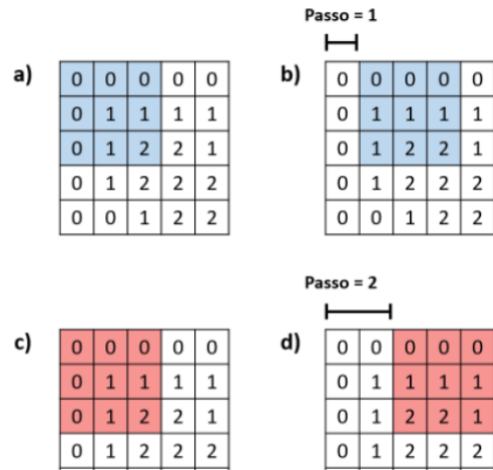


Figura 8 – A imagem corresponde a um passo unitário e a um passo igual a 2. Fonte: (CARNEIRO A. C.; SILVA, 2017)

2.5.4 Camada de Pooling

Depois que a camada de convolução vista na Seção 2.5.2, pega as características mais importante das imagens, a camada de *pooling* é a responsável por diminuir as características vinda da camada anterior. Por exemplo, a camada convolucional envia neurônios 24x24; na camada de *pooling*, ele será reduzido para 12x12.

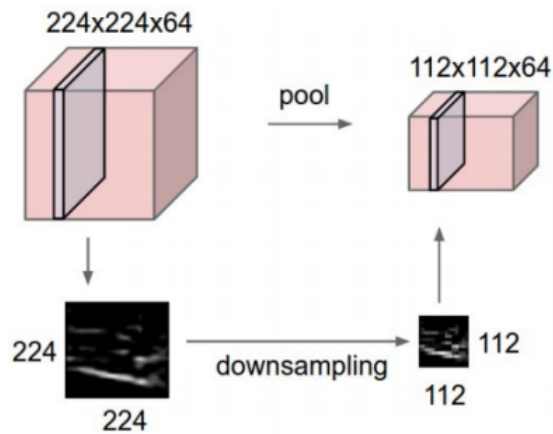


Figura 9 – Efeito da camada de *pooling*. Fonte: (KHAN, 2018)

Uma operação muito usada na camada de *pooling* é a *max-pooling* que é responsável por usar a ativação máxima de um neurônio (NIELSEN, 2016). Além da *max-pooling*, existem outras operações como: *min-pooling*, *sum-pooling* e a *revenge-pooling*.

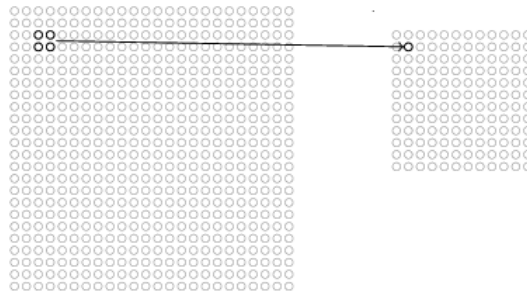


Figura 10 – A figura mostra uma unidade de *pooling* gerando a ativação máxima na região de entrada 2×2 . Fonte: (NIELSEN, 2016).

Segundo USHIZIMA (2017), além de diminuir o tamanho da imagem pegando os seus valores máximos, a *max-pooling* ajuda na redução do consumo computacional e cria pequenas variações nas imagens, tornando a rede mais especializada.

Posteriormente na Seção 2.5.5, tem-se a última camada conhecida como a camada totalmente conectada.

2.5.5 Camada Totalmente Conectada

A proposta da camada totalmente conectada é que todos os neurônios da camada anterior, estejam conectados a cada neurônio da próxima camada e a ela é adicionada uma camada de saída. A saída desta camada é alguma classe pre-definida no modelo (KARN, 2016)

2.6 Métricas

Nessa Seção serão apresentadas as métricas utilizadas para avaliar e comparar desempenho das RNC usadas nesse trabalho.

2.6.1 Matriz de Confusão

Matriz de confusão, também chamada de matriz de erro, é bastante usada em MV para ajudar na apresentação visual de informações sobre a acurácia do modelo (STEHMAN, 1997).

Em uma matriz de confusão, a classe que esta no topo esquerdo junto a classe que esta no canto inferior direito formam a diagonal das classes que foram corretamente classificadas, todos os elementos que não pertencem a essa diagonal foram classificados errados.

A Tabela 1 mostra, no lado esquerdo e na vertical, o valor predito que o algoritmo classificou, e no topo o valor real da classe. Além disso, revela que para a classe normal, o algoritmo classificou 5 instâncias de pacientes como normal e que realmente eram normais

(TP). O algoritmo classificou uma instância como doente que na verdade era normal (do inglês, False Positive - FP). Para a classe doença o algoritmo classificou 2 pacientes como normais (FN), quando na verdade eram doentes e 6 como doentes a que realmente eram doentes (do inglês, True Negative - TN).

Tabela 1 – Exemplo de uma matriz de confusão.

	Normal	Doença
Normal	5(TP)	1(FP)
Doença	2(FN)	6(TN)

2.6.2 Acurácia

Para problemas de classificação de modelo, a acurácia é o calculo de previsões feitas corretamente pelo modelo sobre todas a previsões feitas.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

2.6.3 Precisão

Precisão é a razão das previsões acertadas (TP) sobre todas positivas prevista pela classe. É possível observar na Tabela 1, quando os paciente foram preditos como normais e, de fato, eram normais.

$$P = \frac{TP}{TP + FP} \quad (2.2)$$

2.6.4 Cobertura

Traduzido do inglês *recall*, cobertura é a proporção de observações positivas preditas corretamente para as observações na classe.

$$R = \frac{TP}{TP + FN} \quad (2.3)$$

As fórmulas citadas foram extraídas do trabalho de HACKELING (2014).

2.6.5 Log Loss

Log loss, ou *loss*, é uma função de perda que tem como objetivo avaliar a perda, é inversamente proporcional a precisão, ou seja, quanto maior a precisão menor *loss*. A fórmula da *loss* é:

$$H_p(q) = \frac{1}{N} \sum_i^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (2.4)$$

Na fórmula y é o rótulo visto no exemplo da Tabela 1 (1 para doença e 0 para normal) e $p(y)$ é a probabilidade prevista da classificação ser doença para todos os N classificados. Para cada classificação como doença ($y = 1$), adiciona $\log(p(y))$ à perda. Por outro lado, adiciona $\log(1-p(y))$, para cada classificação como normal ($y = 0$).

Formula do loss foi extraída do trabalho de GODOY (2018).

3 Desenvolvimento

Nesse capítulo serão descritas as ferramentas que foram utilizadas para desenvolver o trabalho, o conjunto de dados, com seus problemas, alterações, a RNC proposta, como os dados de entrada, saída e camadas.

3.1 Ferramentas

O Google Colab ¹ é uma ferramenta em nuvem, desenvolvida pela empresa Google, com alto poder computacional, disponibilizada de maneira gratuita e muito utilizado no aprendizado de máquina usando a linguagem python nas versões 2.7 e 3.6. A Figura 11 exibe o Google Colab treinando uma rede neural convolucional:

Figura - Execução do código de treinamento no Google Colab.

```

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas
+ Código + Texto
Non-trainable params: 20,024,384

[ ] 1 from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
    2 checkpoint = ModelCheckpoint("vgg19.h5",monitor="val_acc",verbose=1,save_best_only=True,
    3                               save_weights_only=False,period=1)
    4 earllystop = EarlyStopping(monitor="val_acc",patience=5,verbose=1)

WARNING:tensorflow:'period' argument is deprecated. Please use 'save_freq' to specify the frequency in number of batches seen.

1 history = model.fit(x_train,y_train,batch_size=32,epochs=20,validation_data=(x_test,y_test),
2                       verbose=1,callbacks=[checkpoint,earllystop])
3 # history = model.fit(x_train,y_train,batch_size=35,epochs=40,validation_data=(x_test,y_test),
4                       verbose=1,callbacks=[checkpoint,earllystop])

Epoch 1/20
28/28 [=====] - ETA: 0s - loss: 0.0125 - accuracy: 0.9943WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [=====] - 11s 401ms/step - loss: 0.0125 - accuracy: 0.9943 - val_loss: 0.0673 - val_accuracy: 0.9679
Epoch 2/20
28/28 [=====] - ETA: 0s - loss: 0.0140 - accuracy: 0.9943WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [=====] - 11s 400ms/step - loss: 0.0140 - accuracy: 0.9943 - val_loss: 0.0713 - val_accuracy: 0.9725
Epoch 3/20
28/28 [=====] - ETA: 0s - loss: 0.0160 - accuracy: 0.9943WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [=====] - 11s 400ms/step - loss: 0.0160 - accuracy: 0.9943 - val_loss: 0.0533 - val_accuracy: 0.9771
Epoch 4/20
28/28 [=====] - ETA: 0s - loss: 0.0202 - accuracy: 0.9920WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [=====] - 11s 400ms/step - loss: 0.0202 - accuracy: 0.9920 - val_loss: 0.0788 - val_accuracy: 0.9679
Epoch 5/20
28/28 [=====] - ETA: 0s - loss: 0.0152 - accuracy: 0.9943WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [=====] - 11s 401ms/step - loss: 0.0152 - accuracy: 0.9943 - val_loss: 0.0543 - val_accuracy: 0.9862
Epoch 6/20
28/28 [=====] - ETA: 0s - loss: 0.0264 - accuracy: 0.9920WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [=====] - 11s 401ms/step - loss: 0.0264 - accuracy: 0.9920 - val_loss: 0.0560 - val_accuracy: 0.9725
Epoch 7/20
28/28 [=====] - ETA: 0s - loss: 0.0074 - accuracy: 1.0000WARNING:tensorflow:Can save best model only with val_acc available, skipping.
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [=====] - 11s 401ms/step - loss: 0.0074 - accuracy: 1.0000 - val_loss: 0.0856 - val_accuracy: 0.9771

```

Figura 11 – Interface gráfica do google colab, saída de dados

O Google Colab é muito parecido com o jupyter notebook e foi um grande facilitador para o desenvolvimento desse trabalho em integração com a github, aceleração de GPU e bibliotecas, como a TensorFlow, e keras já instaladas.

¹ <https://colab.research.google.com/>

O TensorFlow é uma ferramenta python de código aberto, construída pela Google em novembro de 2001, muito utilizado por pesquisadores e cientistas. Ela fornece algoritmos prontos para serem utilizados na construção e configuração das redes neurais, tendo suporte para rodar em quase todos os sistemas operacionais.

O Keras é uma biblioteca de rede neural de alto nível executada no TensorFlow, bastante utilizado na construção de redes neurais. Enquanto o TensorFlow é responsável pela comunicação em baixo nível, ou se já próxima à máquina, o Keras roda sobre ele como uma espécie de *interface*, facilitando o trabalho do desenvolvedor, sendo mais amigável porque tem o Python integrado. Um exemplo dessa facilitação é a permissão do uso de unidade de processamento gráfico (GPU) e unidade de processamento tensorial (TPU).

3.2 Conjunto de Dados

O (ODIR) reconhecimento inteligente de doenças oculares é um conjunto de dados oftalmológicos contendo 5,000 pacientes com idade, imagens coloridas do fundo dos olhos e os diagnósticos das doenças nos olhos esquerdos e direitos.

Cada olho desse conjunto de dados está classificado em 8 doenças oculares, contendo mais uma classificação que é o olho normal, sem nenhuma doença.

Cada classe recebe os seguintes rótulos:

- Normal (N)
- Diabetes (D)
- Glaucoma (G)
- Catarata (C)
- DMRI (A)
- Hipertensão (H)
- Miopia (M)
- Doenças / anormalidades (O)

O conjunto de dados não tem a mesma quantidade de imagens para cada doença. Foi verificado que a classe hipertensão tem apenas 100 imagens no seu conjunto, enquanto a classe normal tem 1088 imagens.

Além disso as imagens não são todas do mesmo tamanho e cada paciente pode ter uma doença, duas ou nenhuma doença.

Outro conjunto de dados usado nesse trabalho foi o *retina dataset*, esse conjunto de dados foi disponibilizado por Chen (2016) e contém 600 imagens do fundo do olho divididas em 4 classes:

- Normal
- Glaucoma
- Catarata
- Doença da retina

A Tabela 2 mostra as doenças contidas no conjunto de dados e quantidade de imagens para cada doença.

Tabela 2 – informações do conjunto de dados.

Doenças	Normal	Glaucoma	Catarata	Doença da retina
Números de imagens	300	100	100	100
Total de imagens				600

Por último tem-se o conjunto de dados proposto nesse trabalho, que é a junção do ORDIR e o *retina dataset*. A Figura 12 é possível ver as imagens dos dois conjuntos de dados unidas em um único conjunto de dados.

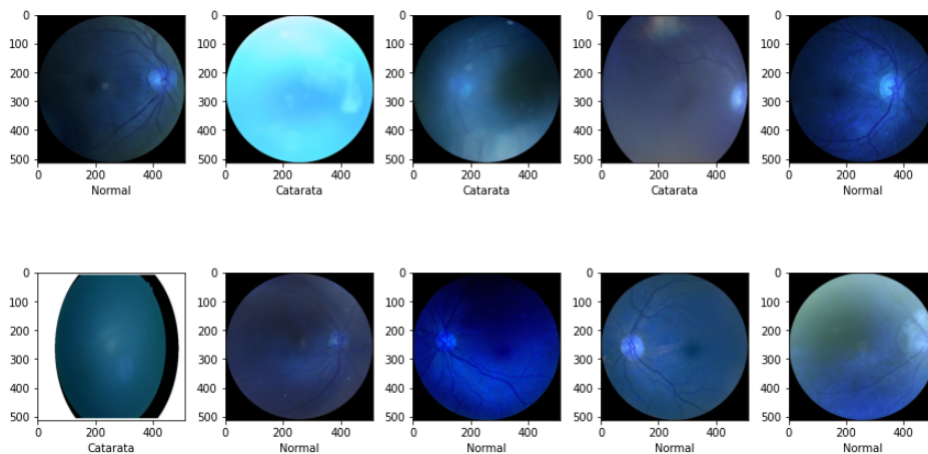


Figura 12 – Imagens do conjunto de dados proposto neste trabalho

Depois do processo de pré-processamento mostrado na Seção 3.2.1, o conjunto de dados proposto totalizou 1488 imagens dando origem assim a um conjunto de dados mais robusto e diversificado.

3.2.1 Pré-processamento

Nesse trabalho foram utilizadas duas classes, normal e catarata. Isso se deu porque observou-se um problema no conjunto de dados ODIR. Verificou-se que as marcações de uma doença se referiam aos dois olhos, sendo que cada olho poderia ter uma doença diferente, ou apenas um ter a doença. Por exemplo, um paciente que tem catarata em um olho e miopia no outro, o arquivo só classifica uma doença.

Left-Fundus	Right-Fundus	Left-Diagnostic Keywords	Right-Diagnostic Keywords	N	D	G	C	A	H	M	O	filepath	labels	target	filename
0_left.jpg	0_right.jpg	cataract	normal fundus	0	0	0	1	0	0	0	0	../input/ocular-disease-recognition-odir5k/ODI...	[N]	[1, 0, 0, 0, 0, 0, 0]	0_right.jpg
1_left.jpg	1_right.jpg	normal fundus	normal fundus	1	0	0	0	0	0	0	0	../input/ocular-disease-recognition-odir5k/ODI...	[N]	[1, 0, 0, 0, 0, 0, 0]	1_right.jpg
2_left.jpg	2_right.jpg	laser spot, moderate non proliferative retinopathy	moderate non proliferative retinopathy	0	1	0	0	0	0	0	1	../input/ocular-disease-recognition-odir5k/ODI...	[D]	[0, 1, 0, 0, 0, 0, 0]	2_right.jpg

Figura 13 – A imagem mostra que na linha 3 o paciente é diagnosticado com duas doenças, porém só uma foi rotulado.

Sabendo disso, antes de extrair as imagens com diagnóstico de catarata do conjunto de dados, foi verificado se o diagnóstico se referia ao olho esquerdo ou ao olho direito. Para saber se os olhos estavam normais foi verificado se o diagnóstico era diferente de catarata e igual a normal em cada olho separadamente, esquerdo e direito, totalizando 558 imagens com catarata e 1088 normais.

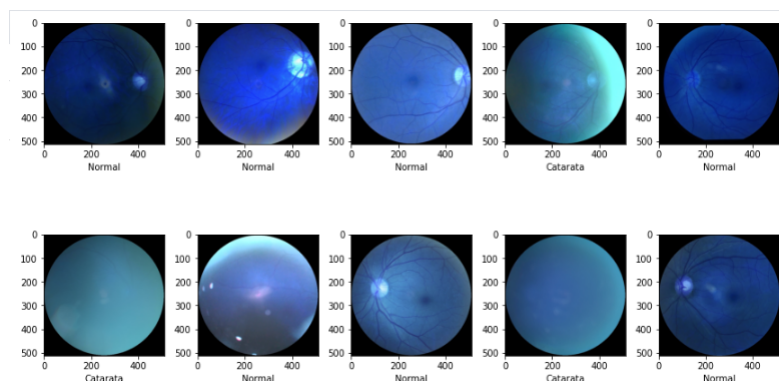


Figura 14 – Imagens do conjunto de dado, com imagens de treino com as classes normal e catarata.

No conjunto de dados retina dataset não foi preciso separar as as imagens porque apesar de o conjunto de dados terem outras classes com imagens de outras doenças, as

classes catarata e normal já estão devidamente separadas. Porém foi preciso rotular as imagens dizendo se ela pertencia a classe 0 (normal) ou 1 (catarata).

Com os dois conjuntos de dados separados em duas classes e as imagens rotuladas, as imagens da classe catarata do ODIR foram juntadas com as imagens da classe catarata do retina dataset e o mesmo aconteceu com as imagens das classes normais, dando origem ao conjunto de dados proposto nesse trabalho. Todas as imagens foram redimensionadas para 512x512 *pixels* e RNC treinou os 3 conjuntos de dados separadamente.

3.3 Arquitetura da Rede Neural Convolutiva

Uma RNC é dividida em algumas partes distintas, onde cada parte executa tarefas diferentes e essas tarefas são executadas em dados de entradas (VARGAS, 2016).

A rede usada nesse trabalho foi retirada da plataforma *Kaggle*, essa plataforma foi criada no ano de 2010 e tem a maior competição de aprendizado de máquina do mundo, além disso, é a maior plataforma em hospedagem de projetos existente atualmente, possuindo 400 mil projetos públicos e mais de 50 mil conjuntos de dados.

Especificamente a RNC foi aproveitada do trabalho do JASHORE K (2020), e sofreu modificações afim de melhorar seu desempenho e resultados.

Recebeu como entrada as imagens dos olhos normais e com catarata, que foram divididas em duas partes, 80% para os dados de treino e 20% para teste. As principais bibliotecas usadas foram *keras* e *tensorflow*, e os parâmetros que foram passados para a rede foram 20 épocas e o *batch-size* igual a 32. Foram testadas várias estruturas de rede, com diversos números de camadas e filtro.

A primeira camada convolutiva possui 16 filtros de tamanho 2x2 e recebe como entrada uma imagem 512x512 com 3 canais. As outras duas camadas convolucionais, segunda e a terceira se diferenciam da primeira, porque se utilizam de 32 e 64 filtros, sucessivamente.

Em todas as camadas convolucionais são aplicadas uma função de ativação ReLU, *Rectified Linear Units*.

Para diminuir o tamanho das imagens e ganhar mais processamento, foi adicionado às camadas convolucionais 3 camadas de *Max-Pooling*, vistas nas Figuras 9 e 7. Também foram utilizados 3 métodos *dropout* vinculados às camadas de convoluções e a *Max-Pooling*, com a função de desativar 50% dos filtros a cada rodada, afim de generalizar o modelo e evitar *overfitting*.

Logo depois, a função *flatten* transforma tudo em um *array*, a camada *dense* conecta todos os filtros, com a função de ativação sendo *sigmoide*.

Para DENNY (2020), a função *sigmoide* deve ser usada em resolução de problemas envolvendo probabilidade de saída ser de uma classe ou de outra, o que chamamos de classificação binária.

Por ultimo, tem-se a última camada com a saída do diagnóstico. A arquitetura completa pode ser vista na Tabela 3.

Tabela 3 – Arquitetura da Rede, com todas a sua parte, entrada, camadas convolucionais, max-polling, dropout, e saidas.

Layer (type)	Output Shape	Param #
VGG19 (Functional)	(None, 16, 16, 512)	20024384
conv2d (Conv2D)	(None, 15, 15, 16)	32784
max_pooling2d (MaxPooling2D)	(None, 7, 7, 16)	0
conv2d_1 (Conv2D)	(None, 6, 6, 32)	2080
max_pooling2d_1 (MaxPooling2D)	(None, 3, 3, 32)	0
dropout (Dropout)	(None, 3, 3, 32)	0
conv2d_2 (Conv2D)	(None, 2, 2, 64)	8256
max_pooling2d_2 (MaxPooling2D)	(None, 1, 1, 64)	0
dropout_1 (Dropout)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 256)	16640
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

No treinamento, a função de perda usado foi a binary crossentropy e para otimizar o resultado foi utilizado o adam, proposto por KINGMA D. P.; BA (2015).

4 Experimento e Resultados

Nesse capítulo serão apresentados soluções e resultados diferentes para o problema proposto.

4.1 Experimento

A rede utilizada foi a VGG-19, propostas por SIMONYAN K;ZISSERMAN (2014), sua arquitetura é composta por camadas convolucionais 3x3 em forma de pilhas, ou seja, uma sobre a outra. Além da sua arquitetura original, foi acrescentada a essa rede outras camadas, filtros e funções.

Primeiramente, as camadas foram coladas uma a uma. A cada camada colocada, foi verificada se o modelo obteve melhoras. Os filtros de cada camada foram colocados, à princípio, de forma decrescente, porém, as redes não tiveram melhoras em comparação com a arquitetura inicial. Depois de alguns testes, observou-se que a arquitetura que obteve melhor resultado para as duas redes foi uma arquitetura com 3 camadas, em que os seus filtros foram colocados de forma crescente. A arquitetura proposta pode ser vista na Seção 3.3.

O experimento foi feito usando os conjuntos de dados (ODIR), *retina dataset* e o conjunto de dados proposto nesse trabalho, mostrados na Seção 3.2.

No treinamento foi utilizado 20 épocas, pois verificou-se que a quantidade de épocas maiores que 20, a rede apresentou indicadores de overfitting.

4.2 Resultados

Nessa Seção será apresentado os resultados, da rede VGG-19 usando três conjuntos de dados diferentes *retina dataset* o ODIR e conjunto de dados proposto tendo como base as métricas mostradas na Seção 2.6.

As Tabelas 4 e 5, mostram as métricas acurácia, precisão, cobertura, F1, que é a média ponderada da precisão e cobertura e o suporte que avalia o equilíbrio da rede. Para a extração desses dados foi usado o relatório de classificação (*Classification Report*) fornecido pela biblioteca *Scikit-learn*.

Nas Figuras 15 e 17 tem-se as matrizes de confusão e nas Figuras 16 e 18 os gráficos com a acurácia e o *loss* de cada rede ao logos das épocas.

4.2.1 Retina dataset

A Tabela 4 mostra resultados máximo tanto na classe 0 (normal) enquanto na classe 1 (catarata), porém apresenta um desequilíbrio já que o suporte o ideal seria a metade de 80 para cada classe, ou seja 40 para catarata e 40 para normal, já que estamos tratando de duas classes. Isso acontece porque o número de imagens para as duas classes não são iguais, na Tabela 2 pode ser visto que o número de imagens normais é superior a imagens com catarata, por isso seu suporte é maior.

Tabela 4 – A tabela contém o relatório de classificação com as métricas acurácia, precisão, cobertura, F1 e suporte.

	precisão	cobertura	F1	suporte
0	1.00	1.00	1.00	61
1	1.00	1.00	1.00	19
acurácia			1.00	80

A matriz de confusão na Figura 15 apresentou valores altos na sua diagonal positiva, isso quer dizer que a rede teve um número alto de acertos, já na diagonal negativa o valor do resultado foi zero, isso quer dizer que a rede não errou nenhuma das imagens da amostra que foi apresentada a rede.

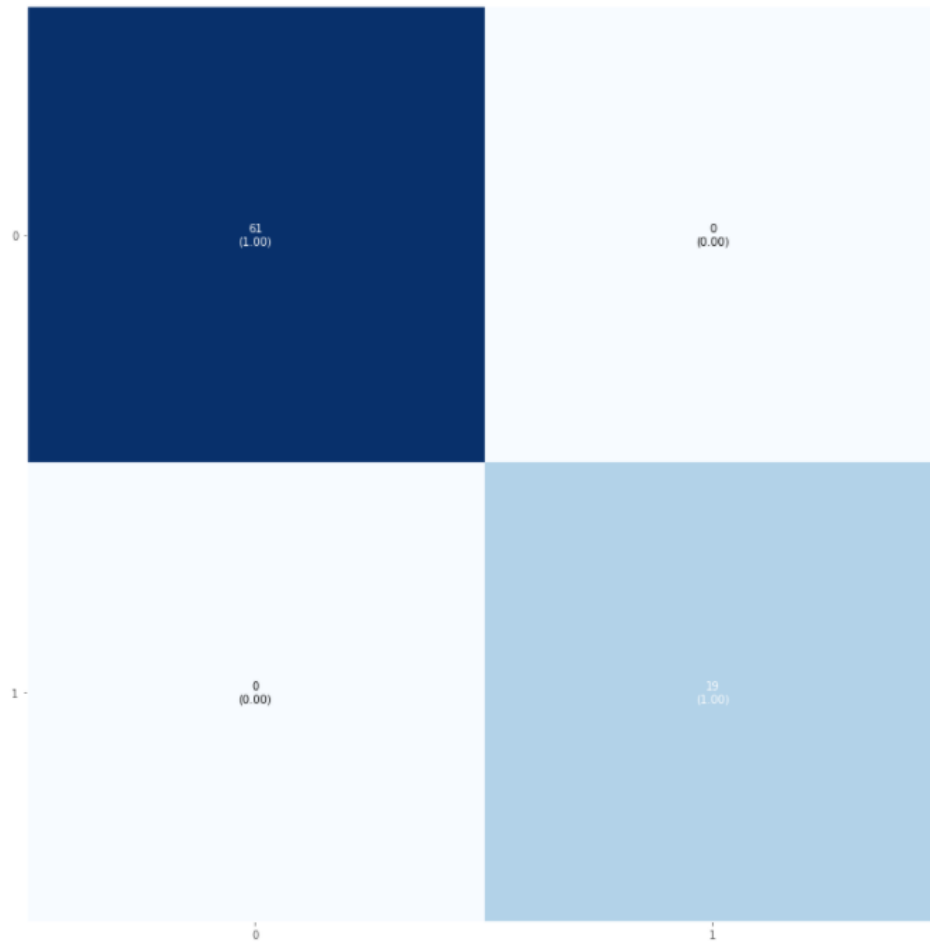


Figura 15 – Matriz de confusão, a classe 0 (normal) e a classe 1 (catarata).

Na Figura 16 o modelo apresenta valor máximo de acurácia e baixo valor de *loss* o que é ótimo, mostra também que o modelo se comportou bem tanto no treino quanto na validação.

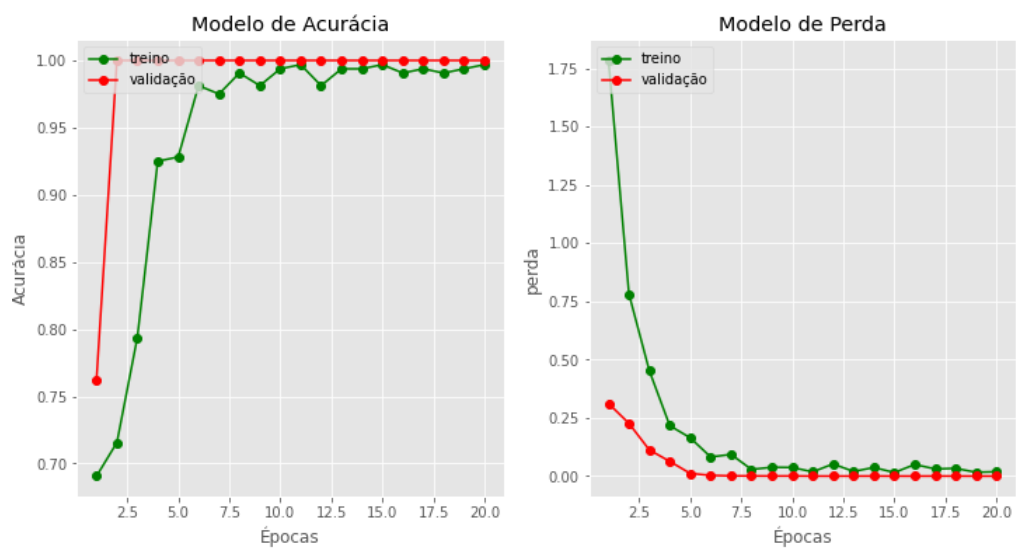


Figura 16 – A imagem mostra dados do treino e validação logo das épocas.

4.2.2 ODIR

Na Tabela 5 são apresentados ótimos valores para as duas classes analisadas, 0 (normal) e 1 catarata, as classes mostram um valor de equilíbrio satisfatório, muito perto do ideal.

Tabela 5 – A tabela contém o relatório de classificação com as métricas acurácia, precisão, cobertura, F1 e suporte.

	precisão	cobertura	F1	suporte
0	1.00	0.97	0.99	105
1	0.97	1.00	0.99	113
acurácia			0.99	218

A Figura 17 tem-se a matriz de confusão apresentando altos valores na sua diagonal positiva, errando muito pouco.

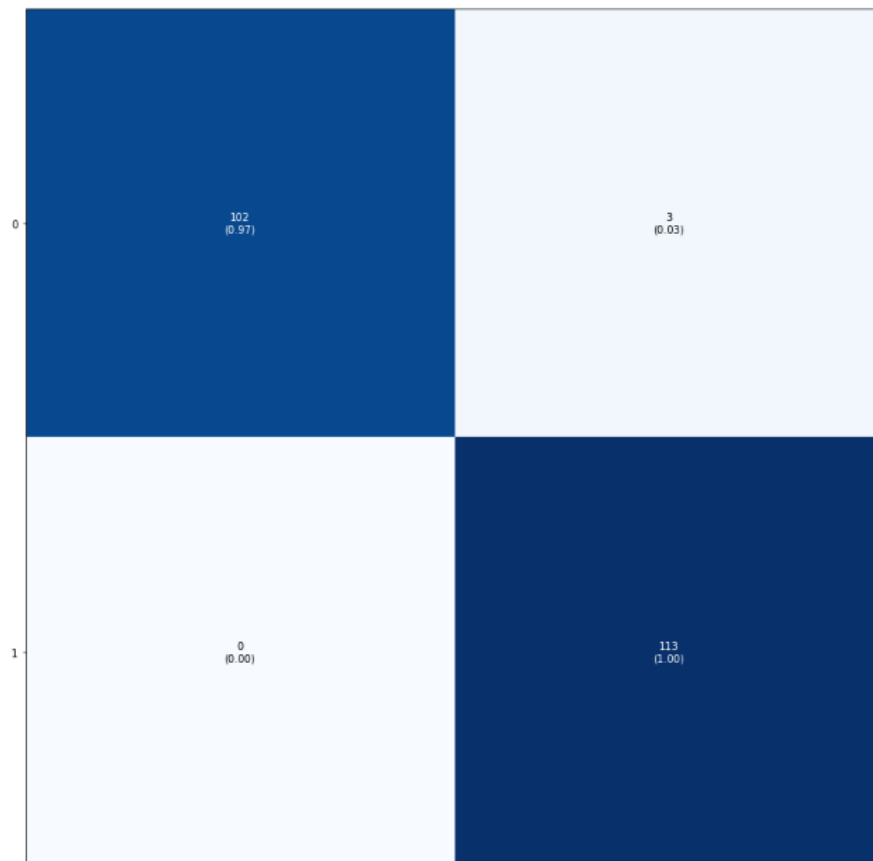


Figura 17 – Matriz de confusão, a classe 0 (normal) e a classe 1 (catarata).

Na Figura 18 tem-se os valores da acurácia e *loss* ao longo das épocas para os dados de treino e validação. O gráfico mostra que a rede conseguiu ótimos resultados para essas duas métricas e que a rede conseguiu se comportar de maneira satisfatória com os dados de validação, dados desconhecidos pelo modelo.

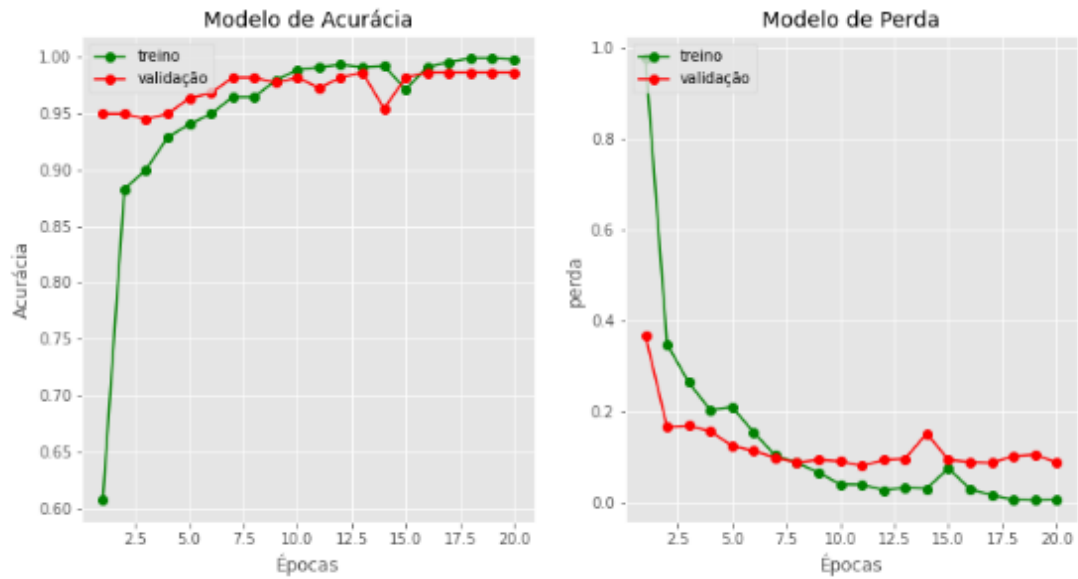


Figura 18 – A imagem mostra dados do treino e validação logo das épocas.

4.2.3 Conjunto de dados proposto

Na Tabela 6 são apresentado ótimos valores para as duas classes analisadas, 0 (normal) e 1 catarata, as classes mostram um certo desequilíbrio natural, já que, o número de imagens das classes não são iguais.

Tabela 6 – A tabela contem o relatório de classificação de acordo com as métricas acurácia, precisão, cobertura, F1 e suporte.

	precisão	cobertura	F1	suporte
0	1.00	0.95	0.97	166
1	0.94	1.00	0.97	132
acurácia			0.97	298

A Figura 19 tem-se a matriz de confusão apresentando ótimos valores na diagonal positiva, e na diagonal negativa errando pouquíssimas imagens normais, mostrando eficiência da RNC.

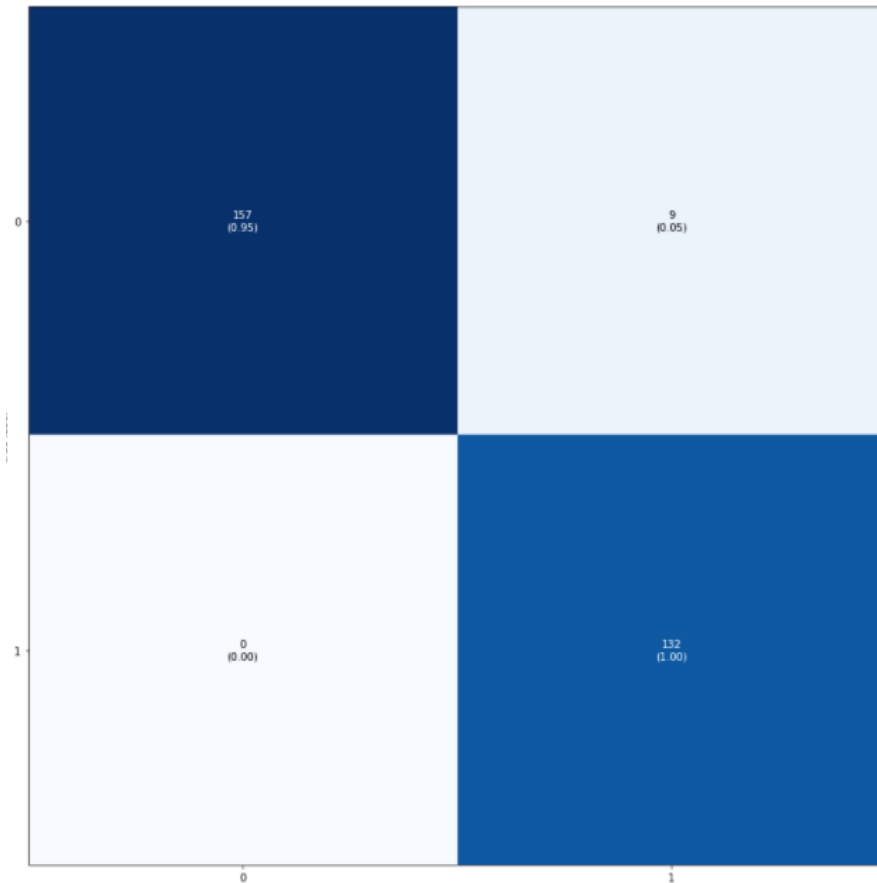


Figura 19 – Matriz de confusão, a classe 0 (normal) e a classe 1 (catarata).

Na Figura 20 o gráfico mostra que a rede conseguiu ótimos resultados para as métricas acurácia e *loss* no treino, porém os dados de validação apresenta algumas variações mostrando que a rede teve um pouco de dificuldade com os dados desconhecidos.

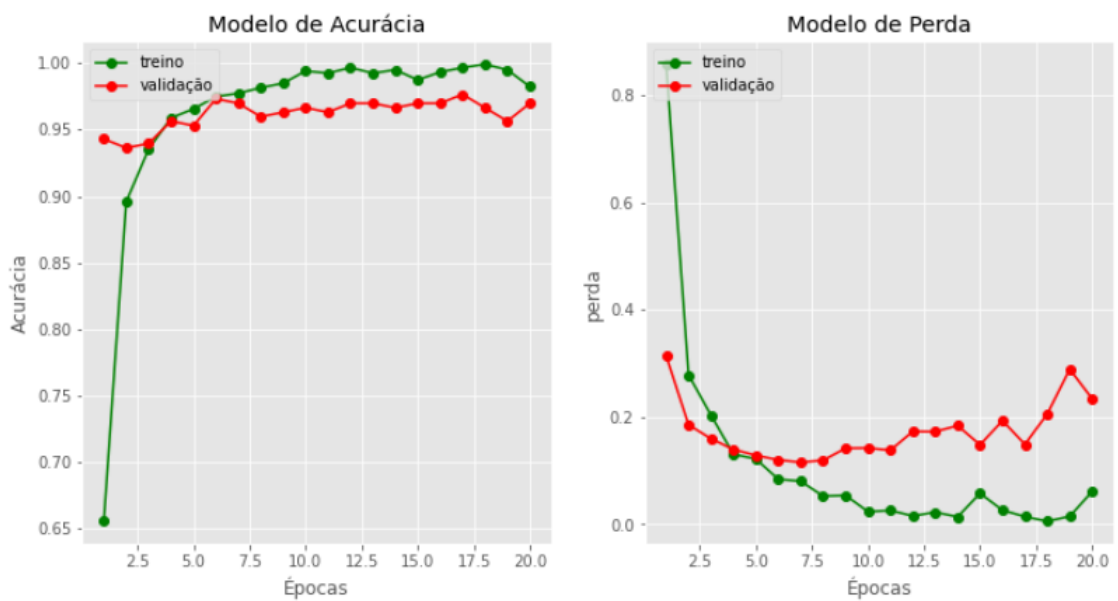


Figura 20 – A imagem mostra dados do treino e validação logo das épocas.

4.2.4 Análise dos Resultados

A validação do modelo foi feita usando imagens já rotuladas e que não foram usadas no processo de treinamento, isso é feito para certificar se a rede realmente aprendeu. A Figura 21 mostra o resultado da validação do conjunto de dados proposto.

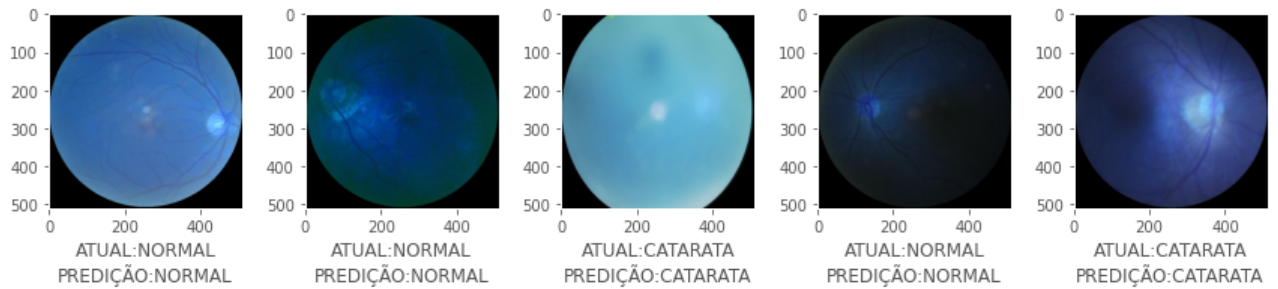


Figura 21 – Imagens de validação classificadas corretamente pelo modelo.

A RNC obteve valores animadores para todos os conjuntos de dados usados nesse trabalho. Esses valores pode ser visto na Tabela 7.

Tabela 7 – resultados sobre a rede e dos conjuntos de dados treinados

RNC	Conjunto de dados	Total de Imagens	Números de épocas	Acurácias
VGG-19	Retina dataset	600	20	1.00
VGG-19	ODIR	1.088	20	0.99
VGG-19	Proposto	1.688	20	0.97

Mesmo a rede VGG-19 apresentados ótimos resultados nas métricas que foram submetidas durante o processo de validação, vista na Seção 2.6, percebeu-se alguma inconsistência nas suas predições. Como já foi visto na Seção 2.2.1, a catarata causa uma mancha nos olhos, mas ora a rede classifica as imagens como machas de forma correta, ora clássica de forma errônea. Existem também situações em que a imagem contém deformações e a rede consegue classificar corretamente, em outras situações, a imagem parece ser legível, mas a rede erra.

Na Figura 22, é feita classificação corretamente de uma imagem como normal, porém essa imagem apresenta uma sombra na sua parte inferior.

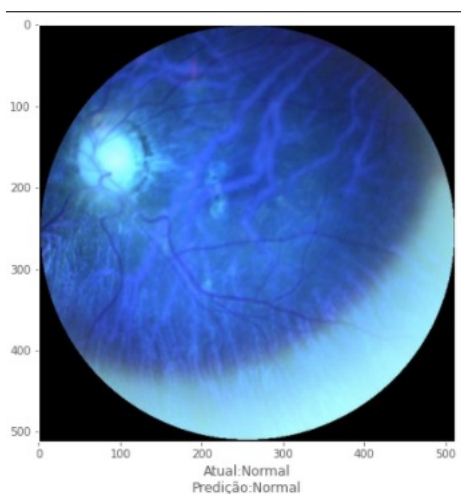


Figura 22 – Imagem de validação normal, classificada corretamente pelo modelo com normal.

Na Figura 23, o modelo erra a imagem com o mesmo defeito da Figura 22, classificando uma imagem normal como se ela tivesse com catarata. Nota-se que a sombra da Figura 23, é menos perceptível, em comparação com a Figura 22.

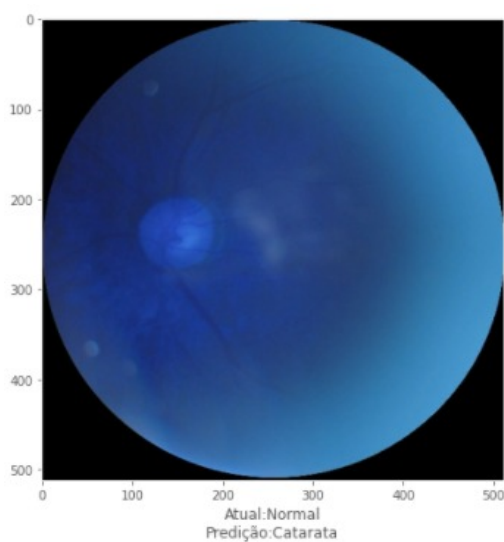


Figura 23 – Imagem de validação normal predita pelo modelo erroneamente como catarata.

Na Figura 24, a classificação é correta, mesmo a imagem apresentando uma deformação.

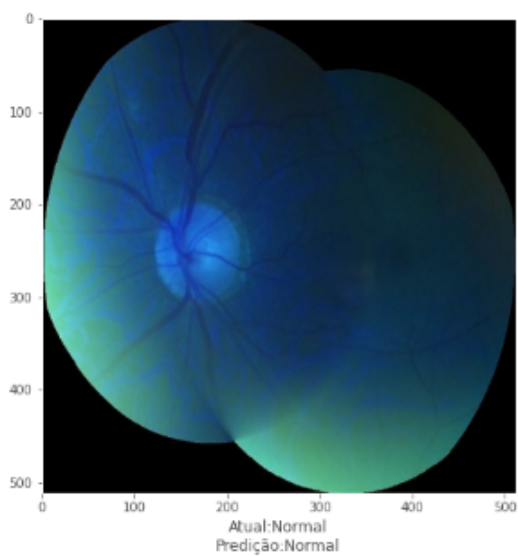


Figura 24 – Imagem de validação classificada como normal, predita corretamente pelo modelo como normal.

Por outro lado, erra uma imagem legível e sem nenhuma deformação na Figura 25.

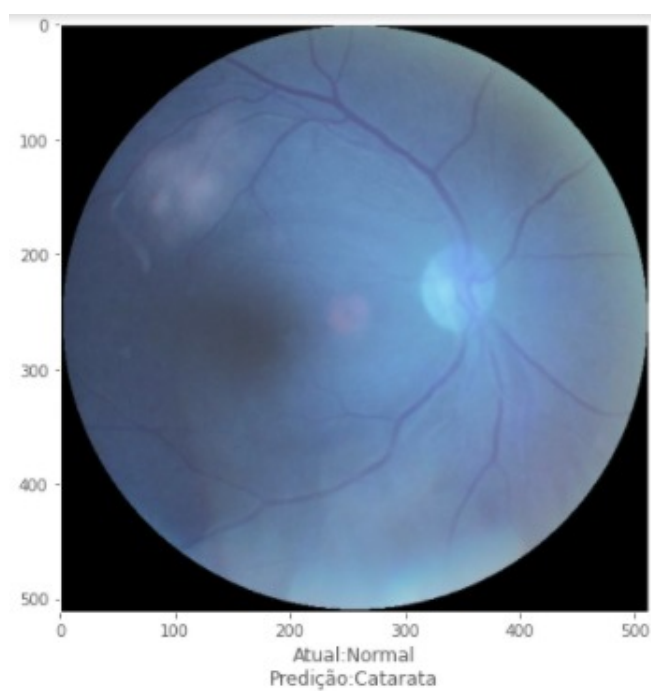


Figura 25 – Imagem de validação classifica incorretamente pelo modelo, é normal, porem ele predisse como catarata.

5 Conclusões

Em suma, foi utilizada a RNC VGG19 com o objetivo de reconhecer a doença catarata através de imagens do fundo dos olhos. Para isso, foram usadas três bases de dados, a ODIR, a *retina dataset* e uma outra base de dados proposta nesse trabalho.

A rede VGG19, que foi modificada e usada para treinar as base de dados, obteve bons resultados para todos os conjuntos de dados usados. Isso significa que a rede proposta nesse trabalho é uma rede generalista, podendo, assim, ser usada para identificar a doença catarata usando imagens capturadas de diferentes lentes, das mais variadas câmeras. Como apresentado nos experimentos e resultados que foram mostrados no capítulo 4.

Apesar de a rede proposta apresentar resultados expressivos, como a acurácia de 99%, precisão de 97% e cobertura de 100% para a base de dados ODIR; acurácia de 100%, precisão de 100% e cobertura de 100% para a base de dados *retina dataset*; acurácia de 97%, precisão de 1.00% e cobertura de 95% para a base de dados proposto, ao analisarmos algumas imagens de validação do conjunto de dados proposto, percebemos algumas falhas na predição da rede. O modelo erra algumas imagens simples e acerta outras imagens consideradas complexas. Porém, esse fato não invalida a RNC, nem sua característica de especialista no reconhecimento da doença catarata usando imagens do fundo olhos.

5.1 Trabalhos futuros

Mesmo com o objetivo do trabalho cumprido, percebemos que existem melhoras a serem efetuadas. São elas:

- **Identificar outras doenças oculares:** Na base de dados existem outras classes contendo outras doenças da visão que podem ser identificadas.
- **Melhorar a base de dados:** O conjunto de dados apesar de ter muitas classes, possui um número pequeno de imagens para cada classe, dificultando o treinamento e a validação.
- **Validar com outros conjuntos de dados:** A fase de validação é a etapa em que é verificado se o modelo realmente aprendeu, por isso, é importante usar vários conjuntos de dados diferentes para fazer a validação.
- **Usar outras redes:** Existem outras redes, tão populares quanto a VGG-19, que podem ser usadas para resolver o problema.
- **Usar novas técnicas:** Algumas técnicas podem contribuir para a melhoria do desempenho da rede e resultados, como, por exemplo: *data augmentation*, essa

técnica consiste em criar dados genéricos que são adicionados a redes a fim de generalizar o modelo e aumentar os números de dados (GOODFELLOW, 2016); Aprendizado multitarefa, técnica que usa informações específicas para melhorar a capacidade de generalização do algoritmo (CARUANA, 1997).

Referências

- ALVES, G. Entendendo redes convolucionais (cnns). 2018. Acessado em 01 de dezembro de 2020. Disponível em: <<https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>>. 11, 24
- BACKES A. R; JUNIOR, J. J. *Introdução a Visão computacional Usando MATLAB*. 1. ed. Rio de Janeiro: Alta Books, 2016. 11, 18, 19, 20, 21
- BALLARD D, H. *Computer vision / Dana H. Ballard, Christopher M. Brown*. 1 ed. ed. [S.l.]: Englewood Cliffs, N.J. : Prentice-Hall, c1982., 1982. 20
- BARELLI, F. *Introdução à Visão Computacional: Uma abordagem prática com Python e OpenCV*. [S.l.]: Editora casa do código. [S.l.: s.n.], 2018. 24
- BARRETO, J. Introdução às redes neurais artificiais. *Laboratório de Conexionismo e Ciências Cognitivas*, Universidade Federal de Santa Catarina, p. 41–50, 2002. 11, 21
- CARNEIRO A. C.; SILVA, R. A. F. Rede neural convulucional com tensorflow: Teoria e prática. *III Escola Regional de informática do Piauí. Livro Anais – Artigos e Minicursos*, III Escola Regional de informática do Piauí. Livro Anais, v. 1, n. 1, p. 382–406, 2017. 11, 23, 24, 25, 26
- CARUANA, R. Multitask learning. *machine learning*. 1997. 46
- CHEN, Y. *Retina Dataset*. 2016. Disponível em: <https://github.com/yiweichen04/retina_dataset>. 32
- DENNY, C. *Funções de ativação: definição, características, e quando usar cada uma*. 2020. Disponível em: <<https://iaexpert.academy/2020/05/25/funcoes-de-ativacao-definicao-caracteristicas-e-quando-usar-cada-uma/>>. 34
- GODOY, D. Understanding binary cross-entropy / log loss: a visual explanation. 2018. Acessado em 01 de dezembro de 2020. Disponível em: <<https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>>. 29
- GOODFELLOW, I. e. a. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>> Acessado em 01 de dezembro de 2020. 46
- HACKELING, G. *Mastering machine learning with scikit-learn*. 2014. 28
- HARLEY, E. A. e. a. *Visão e percepção visual*. San Diego, 2003. Acessado em 01 de dezembro de 2020. Disponível em: <https://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27492003000600001>. 19
- HAYKIN, S. *Neural Networks and Learning Machines*. 3rd. ed. [S.l.]: ed. p. cm. Rev. ed of: Neural networks, 1999. 11, 21, 22
- JASHORE K, D. Cataract prediction using vgg19. 2020. Acessado em 01 de novembro de 2020. Disponível em: <<https://www.kaggle.com/taha07/cataract-prediction-using-vgg19>>. 34

- KARN, U. Neural Networks and Deep Learning An Intuitive Explanation of Convolutional Neural Networks. 2016. Disponível em: <<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>>. 27
- KHAN, S. *A Guide to Convolutional Neural Networks for Computer Vision*. organ claypool publishers. [S.l.: s.n.], 2018. 11, 23, 26
- KINGMA D. P.; BA, J. Adam: A method for stochastic optimization. San Diego, 2015. 35
- LECUN, Y. e. a. Gradient-based learning applied to document recognition. v. 86, n. 11, 1998. 23
- MEHRYAR, M. e. a. Foundations of machine learning. 2012. 23
- MENEZES, C. e. a. Quedas e qualidade de vida de idosos com catarata. Revista Brasileira de Oftalmologia, v. 75, n. 1, p. 41, 2016. Acessado em 01 de Agosto de 2020. Disponível em: <<https://doi.org/10.5935/0034-7280.20160009>>. 19
- MERULA R.V; FERNANDES, L. Catarata infantil: importância do diagnóstico e tratamento precoces. Arquivos Brasileiros de Oftalmologia, v. 68, 2005. Acessado em 01 de novembro de 2020. 20
- MILANO D ; HONORATO, L. B. Visão computacional. *Universidade Estadual de Campinas*, p. 1–7, 2010. 17
- NIELSEN, M. Neural Networks and Deep Learning. 2016. Acessado em 01 de dezembro de 2020. Disponível em: <<http://neuralnetworksanddeeplearning.com/>>. 11, 26, 27
- OTTAIANO, J. *As Condições de Saúde Ocular no Brasil 2019*. 1 ed. ed. são Paulo: Conselho Brasileiro de Oftalmologia, 2019. 17
- RAMOS, A. *Fisiologia da Visão Um estudo sobre o ver e enxergar*. Pontifícia universidade católica do rio de janeiro - puc-rio. [S.l.: s.n.], 2016. 18
- SIMONYAN K; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. 2014. 36
- STEHMAN, S. V. Selecting and interpreting measures of thematic classification accuracy. remote sensing of environment. *III Escola Regional de informática do Piauí. Livro Anais – Artigos e Minicursos*, ISSN 00344257, v. 62, n. 1, p. 77–89, 1997. 27
- TARTARELLA M, B. e. a. Aspectos cirúrgicos em catarata congênita. Revista Brasileira de Oftalmologia, v. 58, n. 1, p. 41, 1995. Acessado em 01 de setembro de 2020. Disponível em: <<http://dx.doi.org/10.5935/0004-2749.19950087>>. 19
- TEMPORINI, E. R. e. a. Catarata senil: Características e percepções de pacientes atendidos em projeto comunitário de reabilitação visual. Revista Brasileira de Oftalmologia, v. 60, n. 1, p. 41, 1997. Acessado em 01 de dezembro de 2020. Disponível em: <<https://doi.org/10.5935/0004-2749.19970103>>. 19
- USHIZIMA, D. M. Redes neurais convolucionais com tensorflow: Teoria e prática. amostras III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos,, v. 1, n. 1, p. 382–406,, 2017. Acessado em 01 de agosto de 2020. 11, 25, 27

VARGAS, A. C. G. e. a. *Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres*. s. in: Proceedings of the xxix conference on graphics, patterns and images. [S.l.: s.n.], 2016. 34