# **Classificação de imagens na triagem de pacientes com deficiência visual**

Douglas Jonatas do Carmo Dias¹; Ana Beatriz Pereira Sette2

¹ Centro Universitário de Lins. Engenheiro de Computação. Rua: José Bonifácio n°244 – Ribeiro; 16401-115. Lins, São Paulo, Brasil.

2 Universidade Federal de Viçosa. Doutora em Economia Aplicada. Departamento de Economia Rural, Avenida Purdue, s/nº, Edifício Edson Potsch Magalhães – Campus Universitário; 36570-900. Viçosa, Minas Gerais, Brasil.

# **Classificação de imagens na triagem de pacientes com deficiência visual**

# **Resumo**

Devido ao acúmulo de atividades no cotidiano estar crescendo exponencialmente, a automação da mesma tornou-se algo fundamental nos dias atuais, sendo essencial em qualquer área. Uma das ferramentas que pode causar grandes impactos na sociedade independente da área que estiver é a visão computacional.Com a aplicação de redes neurais convolucionais é possível automatizar a identificação de existência ou não de pacientes com deficiencia visual. Para isso, utilizou-se técnicas como “Dropout” e “image augmentation”. Este trabalho deu grande importância à explicação de conceitos de visão computacional visando a facilitação da replicação dos procedimentos aqui aplicados. A rede neural convolucional utilizada neste trabalho foi: efficientnetv2-b2-21k. Como resultados, foram obtidas algumas métricas oriundas do treinamento realizado, comprovando os resultados obtidos pela rede e observou-se uma acurácia igual a 81%. Apresentando assim um resultado muito próximo do esperado para a classificação de imagens na triagem de pacientes com deficiência visual.

**Palavras-chave:** catarata; glaucoma; retinopatia diabética;classificação de imagens; visão computacional.

# **Introdução**

Devido ao crescimento das divícias sociodemográficas e da perspectiva de vida mais longa, vários países ao redor do mundo começaram a notar sua população atingindo a idade adulta, aumentando assim a idade média da população e como consequência trazendo mudanças na carga de doenças para doenças não transmissíveis e deficiências. A maioria das principais causas de deficiência visual é a catarata (Bourne et al., 2021).

Carata é a causa mais dominante de cegueira reversível no mundo, sendo o motivo por aproximadamente 50% dos 50 milhões de casos. Por conceito, refere-se à opacidade do cristalino, sendo adquirida no processo de envelhecimento. As fontes causais da catarata não foram completamente elucidadas, porém existem fatores de riscos que devem ser evitados, tonando a catarata é uma questão de saúde pública intimamente relacionada com a longevidade populacional do Brasil e do mundo. (Domingues et al., 2016).

Além da catarata uma das complicações mais comuns é a retinopatia diabética, esta se mostra presente tanto no diabetes tipo 1 quanto no tipo 2, geralmente em pacientes com longo tempo de doença e mau controle glicêmico. Quando atinge a perda visual é considerada trágica e constitui fator importante de morbidade e como consequência elevando o impacto econômico, uma vez que a retinopatia diabética é uma das causas mais

frequentes de cegueira adquirida. No momento as alternativas de tratamento são a foto coagulação a laser de argônio e, em alguns casos, a vitrectomia. O sucesso do tratamento está relacionado com a detecção precoce das lesões (Bosco et al., 2005).

Outra doença que contribui para a cegueira no mundo é o glaucoma, destacando-se como a principal causa de cegueira e deficiência visual irreversível no Brasil e no mundo. Estima-se que globalmente, em 2020, 76 milhões de indivíduos sejam portadores de glaucoma e que este número deve alcançar a marca de 95,4 milhões em 2030. Os tipos de glaucomas mais frequentes são os glaucomas primários de ângulo aberto (GPAA) e de ângulo fechado (GPAF). A presença destes tipos de glaucoma é relativa à área geográfica e da etnia da população. No Brasil, assim como no mundo ocidental, o mais comum é sem dúvida o GPAA, sendo responsável por volta de 80% dos casos. Tanto a prevalecimento quanto a ocorrência do glaucoma aumentam com a idade, sofrendo grande influência da raça do indivíduo. Alcançando os 40 anos, aparecem anualmente aproximadamente 1,6 novos casos de glaucoma para cada 100.000 habitantes e aos 80 anos, 94,3/100.000 habitantes. A predominância para indivíduos brancos e negros, respectivamente, na faixa etária de 73 a 74 anos é de 3,4% e 5,7%. As taxas aumentam para 9,4% e 23,2% para estes mesmos grupos se considerarmos a faixa etária de 75 anos ou mais.

Sendo o glaucoma uma doença de origem genética, a sua prevenção primária, ou seja, evitar o seu aparecimento, ainda é impraticável e o único modo de se evitar a progressão para a cegueira é através da prevenção secundária realizando o diagnóstico precoce e aplicando tratamento eficaz, a prevenção terciária está relacionada a limitar as sequelas da doença e realizar a reabilitação, e a prevenção quaternária está associada ao evitar intervenções diagnósticas e/ou terapêuticas inapropriadas, iatrogênicas ou eticamente questionáveis (Guedes.,2021).

A triagem para os pacientes tanto com possibilidade de adquirirem catarata, retinopatia diabética ou glaucoma é de extrema valia pois utilizando a classificação de imagem irá fortalecer a prevenção secundário, ou seja, possibilitando um diagnóstico precoce e como consequência possibilitando um tratamento eficaz.

**Objetivo da Pesquisa**

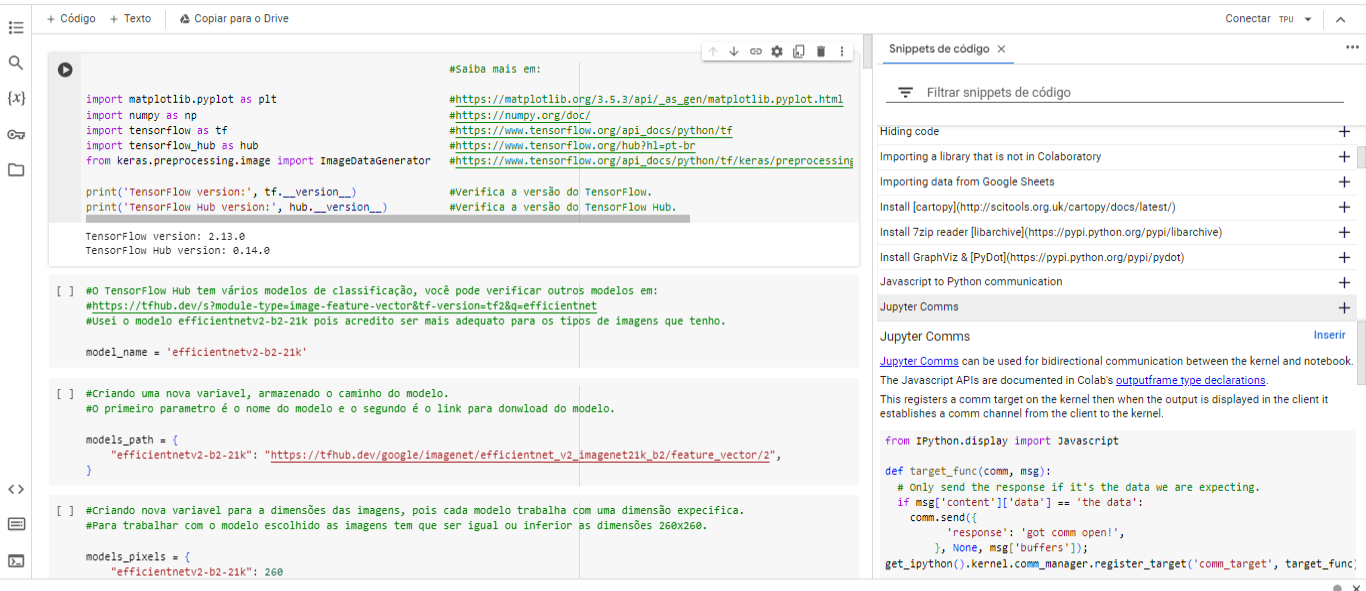
O objetivo da pesquisa é criar um modelo de visão computacional utilizando redes neurais convolucionais para automatizar o processo de recolhimento de imagens do fundo dos olhos para realizar uma triagem entre os pacientes que possuem deficiência visual para que assim os mesmos sejam priorizados no tratamento, como foco principal utilizar no SUS.

# **Material e Métodos**

Os materiais utilizados para o desenvolvimento do trabalho foram:

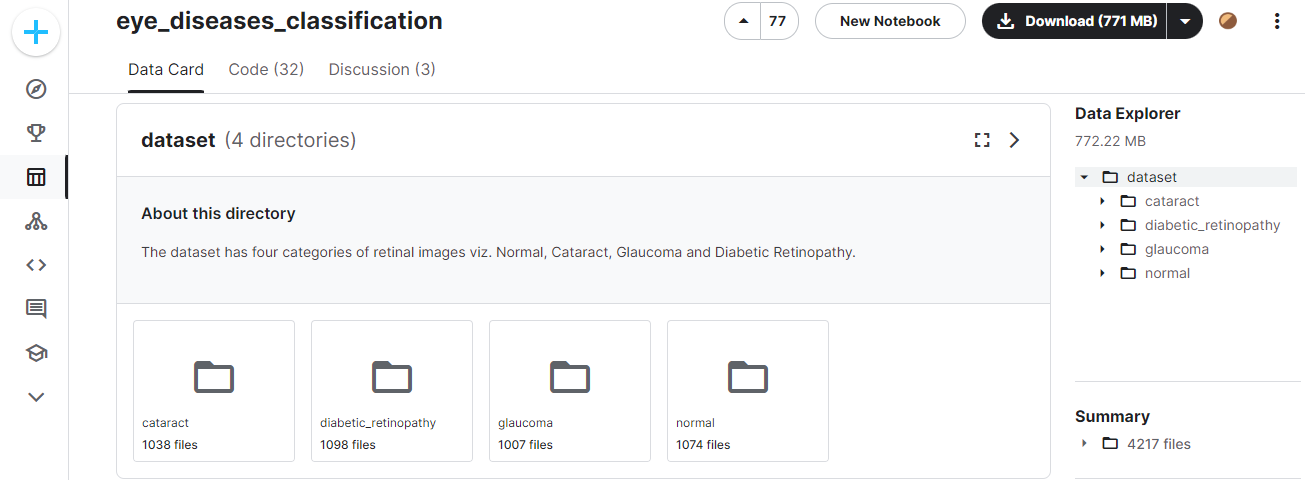
* + Google Colaboratory;
  + Python;
  + Dataset;
  + Matplotlib;
  + Pyplot;
  + TensorFlow;
  + TensorFlow Hub;
  + Keras;
  + efficientnetv2-b2-21k;
  + Google Driver;

O **Google Colaboratory** mais conhecido como Google Colab ou Colab é um serviço de nuvem gratuito hospedado pela próprio Google, é uma ferramenta que permite criar código fonte e texto com imagens e tudo isso sem a necessidade de fazer download de software, além de te acesso a GPUs sem custo financeiro e com possibilidade de compartilhamento de uma forma fácil. O google colaboratory foi utilizado como ambiente para o desenvolvimento do modelo, a imagem a seguir ilustram o que foi mencionado.



**Python** é uma linguagem de programação com proposito de uso geral. Sua filosofia de design enfatiza a legibilidade do código com o uso de recuo significativo**.** Python é uma das linguagens mais usadas quando se trata de data science, principalmente pela sua linguagem flexível e código aberto, o que facilita seu uso para computações mais quantitativas, potencializado pelas suas enormes bibliotecas que permite uma melhor análise de dados, por tal motivo foi escolhida como a linguagem de programação para o desenvolvimento da pesquisa.

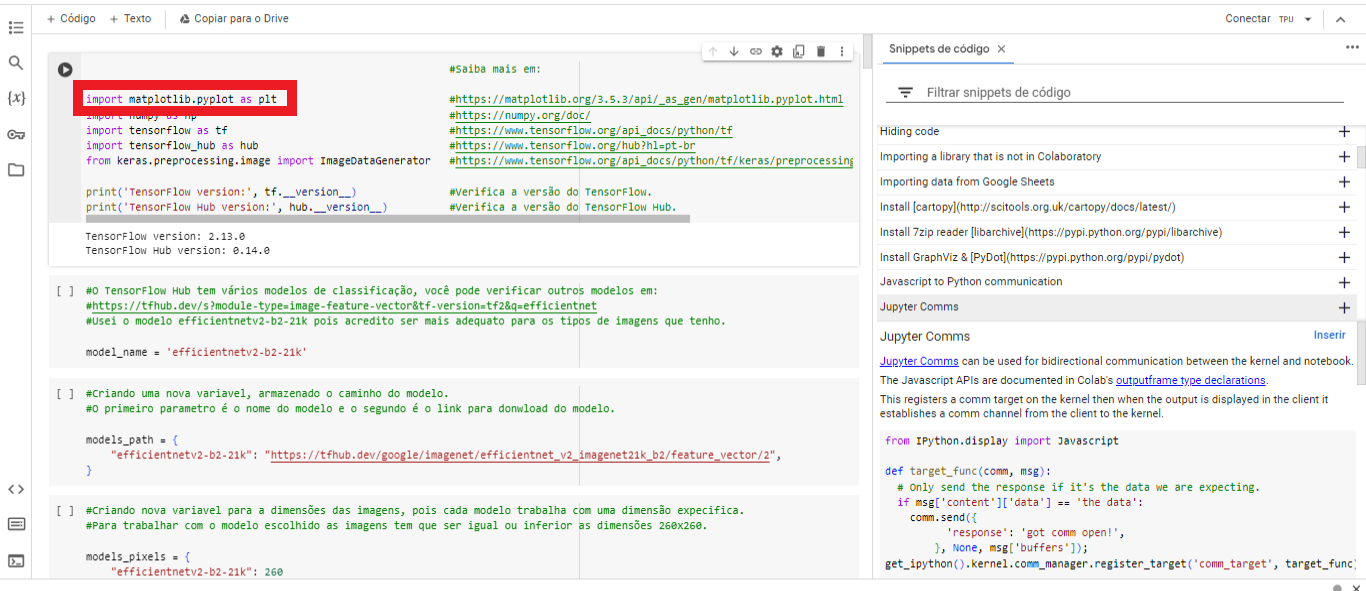
Os **Datasets** ou conjunto de dados, são o principal "material” utilizado nos processos de análise de dados, pois a pesquisa será realizada sobre as imagens coletadas, para esse trabalho foi utilizado um conjunto de imagem já rotuladas retiradas do Kaggle, na seção coleta e tratamento de dados será explicado mais sobre o mesmo. Logo abaixo é possível visualizar a área de onde as imagens foram retiradas.



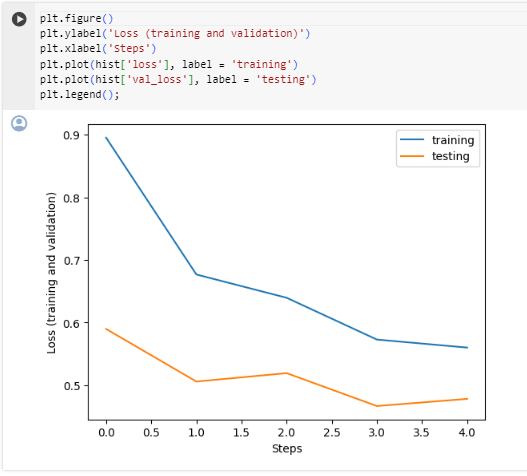
É possível ter acesso a esse conjunto de dados através do site:

<https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification>

**Matplotlib** é uma biblioteca da linguagem de programação Python, utilizada para visualização de dados e plotagem gráfica. Ela é utilizada pela extensão de matemática numérica do Python, a NumPy, e pela biblioteca SciPy. Na mesma imagem utilizada para mostrar o ambiente de desenvolvimento é possível visualizar a importação do módulo Pyplot dessa biblioteca.



**Pyplot** é um módulo do Matplotlib que oferece uma interface parecida ao MATLAB. Matplotlib é projetado para ser tão usável quanto MATLAB, porém com a capacidade de usar Python e a vantagem de ser gratuito e de código aberto, esse módulo foi utilizado para gerar os gráficos do modelo, a seguir um trecho do código utilizando o Pyplot.

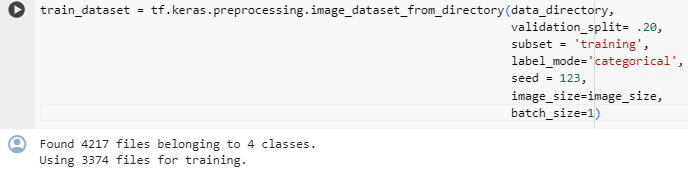


plt é o “apelido” dado para Pyplot no momento da importação, assim é possível chamar o módulo de modo rápido.

**TensorFlow** é uma biblioteca de software de código aberto gratuita e para aprendizado de máquina e inteligência artificial, por isso será muito utilizado para treinar o nosso modelo além de auxiliar no pré-processamento.

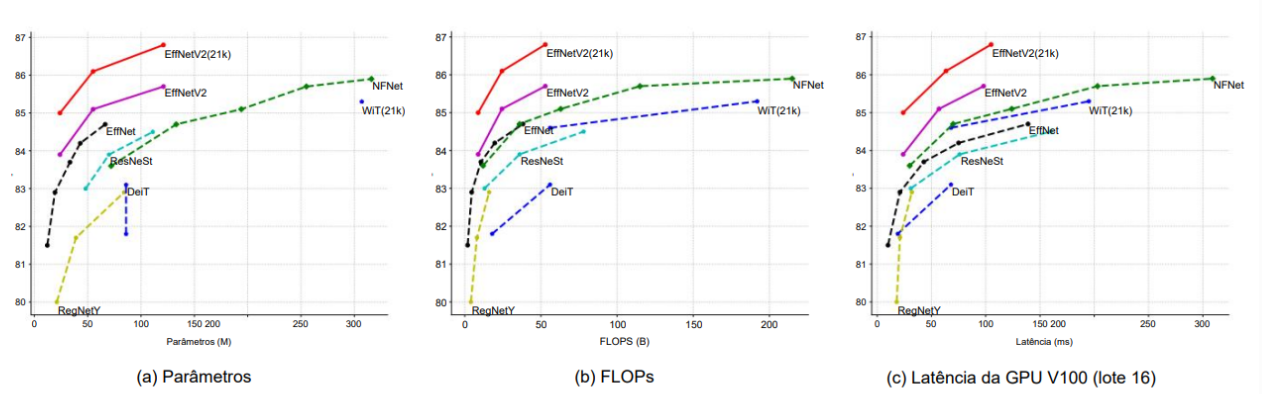
O **TensorFlow Hub** é um repositório de modelos de machine learning treinados prontos para ajustes finais e implantação em qualquer lugar. Possibilitando assim a reutilização de modelos já treinados, foi através desse repositório que foi retirado o modelo EfficientNetV2.

**Keras** é uma biblioteca de código aberto criada para Deep Learning com Python. Ele é utilizado na criação de redes neurais para resolução de várias tarefas diferentes, naessa pesquisa foi utilizado para fazer a classificação de imagens. Keras atua como uma interface para a biblioteca TensorFlow, a seguir é demostrado o uso das bibliotecas no pré-processamento das imagens.



Para realizar o treino as imagens foram retiradas do data\_directory e 20% do dataset foi utilizado para validação, esse conjunto de dados será utilizado para o treinamento sendo as etiquetas do modelo por categorias como: ['0\_normal', '1\_cataract', '2\_glaucoma', '3\_diabetic\_retinopathy'] além da configuração no formato das imagens e o número de amostras processadas antes do modelo ser atualizado. Como resultado temos um total de 4217 imagens e 4 classes, porém para o treinamento será utilizado apenas 3374 imagens.

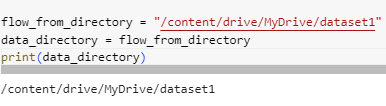
**EfficientNetV2**, é uma nova família de redes convolucionais que possuem velocidade de treinamento e melhor eficiência de parâmetros do que os modelos anteriores, por esse motivo foi utilizado nesse trabalho, os gráficos a seguir fazem uma comparação com outros modelos de redes convolucionais em relação aos parâmetros, Flops e Latência.



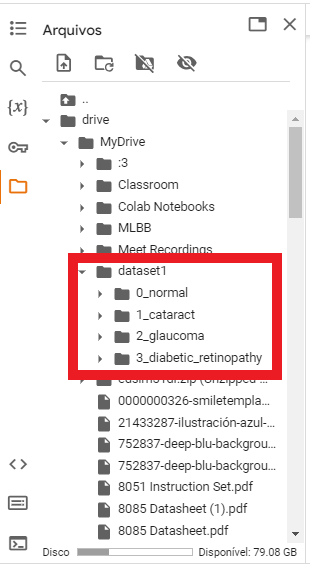
Como observado o EfficientNetV2 demostrou vantagens significativas nos 3 critérios de avaliação.

**Google Drive** é um serviço de armazenamento em nuvem oferecido pelo Google. ele permite que os usuários enviem e armazenem os arquivos na plataforma, podendo acessá-los de qualquer lugar e dispositivo. Para isso, basta ter acesso à internet.

O armazenamento do dataset foi realizado utilizando o Google Driver, pois assim é possível ter um acesso mais rápido aos dados dentro da plataforma de desenvolvimento, abaixo está um trecho do código que exemplifica isso.



Após o dataset ter sido configurado e armazenado no google driver, é possível verificar as imagens dentro da plataforma de desenvolvimento como mostra a imagem a seguir.



Facilitando assim qualquer desenvolvimento com as imagens.

## Coleta e tratamento de dados

Os dados coletados para esse trabalho foram retirados do Kaggle. As imagens já estavam com os rótulos e as mesmas foram colocas no google driver, sendo possível fazer o carregamento para a área de trabalho do Google Colab.

Tratando-se de qualquer modelo de machine learning ou deep learning uma das partes mais importantes no treinamento de qualquer modelo é a parte do pré-processamento dos dados de entrada. Sendo alguns pontos importantes para o mesmo:

* Conjunto de imagens e rótulos;

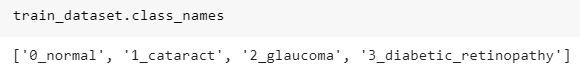


Figura 1: Rótulo das imagens

Fonte: Dados originais da pesquisa

Como o dataset foi retirado do kaggle e o mesmoa já estavam com os devidos rótulos não foi necessário fazer manualmente.

* Configuração do tamanho da imagem para o modelo;

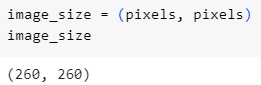


Figura 2: Configuração do tamanho das imagens

Fonte: Dados originais da pesquisa

É necessário definir o tamanho das imagens de acordo com o modelo utilizado, nesse caso as imagens terão o tamanho de 260x260 pixels pois é com essa dimensão que o modelo efficientnetv2-b2-21k trabalha e por tanto as imagens não devem ultrapassar essa configuração.

* Divisão das imagens para o treino;

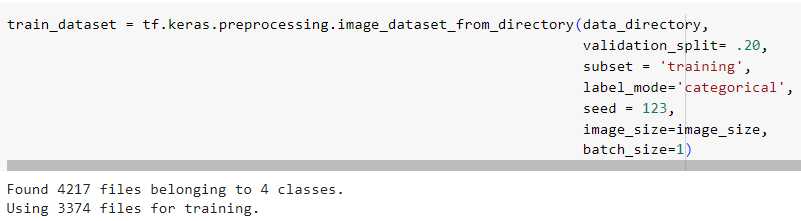


Figura 3: Treino do dataset

Fonte: Imagem proveniente da pesquisa

Para fazer o treinamento foi utilizado 80% das imagens (3374) e ficando 20% (843) para o teste(validação) para isso foi utilizado o método split, a etiqueta desse modelo será por categoria, ou seja: [‘0\_normal’, ‘1\_cataract’, ‘2\_glaucoma’, ‘3\_diabetic\_retinapathy’] toda vez que a função for chamada será composta pelas mesmas imagens, o tamanho das imagens serão 260x260 e o número de amostras processadas antes do modelo ser atualizado será 1.

* Pré-processamento e “criando” das imagens.

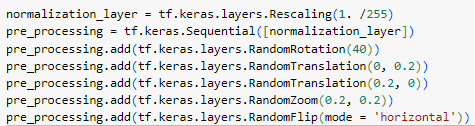


Figura 4: Adicionando e “criando” as imagens

Fonte: Autor

Foi feito a transformação de dados (com normalization) que alinha os valores dos dados a uma escola comum, utilizado o rescaling para reduzindo a escala (pixels) das imagens de 0-255 para 0-1.

Como será passado uma sequencial de camadas para a rede neural será utilizado o Sequential.

Também serão adicionadas algumas imagens no pré-processamento, pois pode ser que as imagens separadas para o treinamento não sejam o suficiente, logo será necessário “criar” imagens para o mesmo, sendo adicionadas algumas imagens aleatórias com algumas modificações como: girando 40°, deslocando 20% das imagens no horizontal e vertical, aumentando o zoom em 20% e com o zoom reduzido em 20% e girando as imagens horizontalmente.

* Criando e treinando a rede neural.

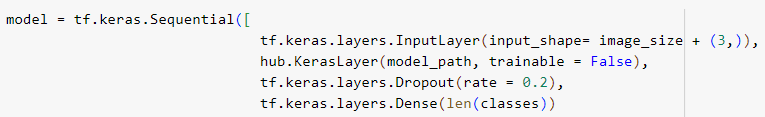


Figura 5: configuração e criação da rede neural

Fonte: Autor

Criando uma sequência de camadas, sendo a primeira a entrada da imagem como canal de cores RGB, a segunda camada irá fazer o download do modelo (efficientnetv2-b2-21k) porém sem alterar os pesos do modelo, na camada seguinte foi aplicado o Dropout um recurso utilizado para prevenir overfitting e por último será a densidade da camada de acordo com a quantidade de classes.

* Divisão das imagens para o teste.

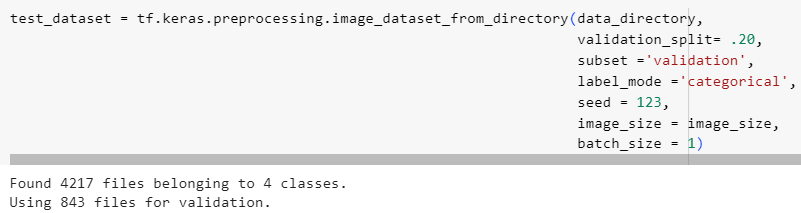


Figura 6: Imagens para o teste

Fonte: Criado pelo Autor

O pré-processamento adotado neste trabalho englobou o embaralhamento das imagens depois da divisão em conjunto de treino e validação, o redimensionamento da imagem para o formato requerido pela rede neurais convolucionais e a transformação da imagem em um objeto do tipo numpy.array, que é uma espécie de representação de tensores em python.

Além do embaralhamento das imagens também foi necessário dividir um conjunto de dados em dois subconjuntos distintos: um conjunto de treinamento, utilizando 80% das imagens e um conjunto de teste utilizando 20%. Essa divisão é essencial para avaliar o desempenho do modelo de forma realista e evitar problemas de “overfitting”.

**Resultados e Discussão**

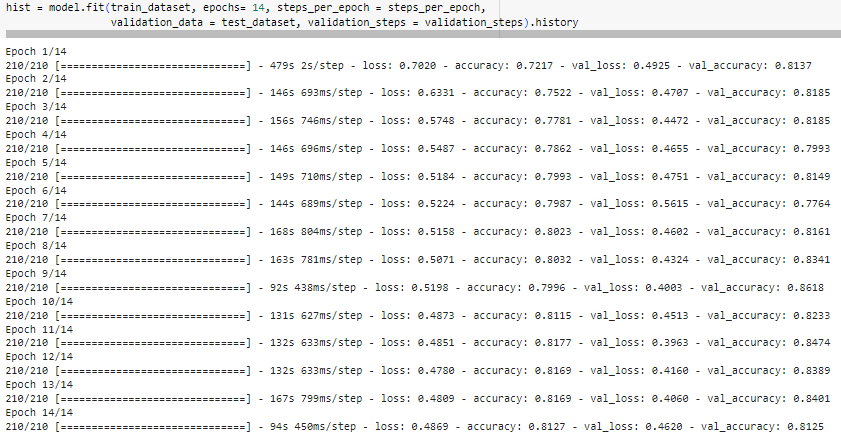


Figura 7: Treinamento do modelo

Fonte: Autor

O modelo foi treinado por 14 épocas, tendo como resultado final uma perda no treinamento de 48% e uma acurácia de 81% números bem semelhantes aos valores do teste, sendo 46% para perda e 81% para acurácia, as imagens a seguir descreve melhor a situação de acordo com as épocas.

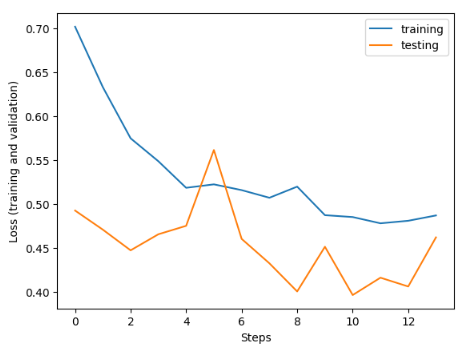


Figura 8:Perda no treinamento e teste

Fonte: Autor

Na figura acima é possível visualizar a perda tanto no treinamento quanto na validação através das épocas em que o modelo estava sendo treinado.

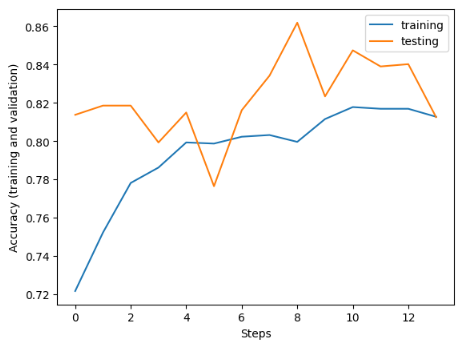


Figura 9: Precisão no treino e teste

Fonte: Autor

Já no gráfico acima é possível fazer uma comparação entre a precisão do treinamento e do teste, o interessante desse gráfico é que ele mostra em qual época o nosso modelo teve uma melhor precisão em relação as épocas em que ele foi treinado.

Após as análises feitas serão criadas duas variáveis, x para os pixels das imagens e y para os rótulos.

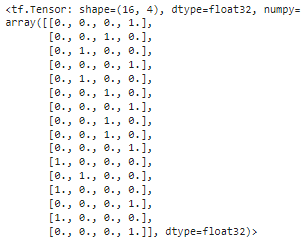


Figura 10: Rótulos da variável y

Fonte: Autor

A imagem acima mostra que para cada lote existe 16 imagens com 4 classes possíveis [‘0\_normal’, ‘1\_cataract’, ‘2\_glaucoma’, ‘3\_diabetic\_retinapathy’], sendoa as linhas as imagens e as colunas as classes, em seguida para realizar a classificação da imagem selecionada, iremos armazenar as informações da imagem em uma viável chamda “image”, passando o código image = x[0, :, :, :], sendo x as coordenadas dos pixels, o 1° parametro é a posição da imagen que se encontra no y, o 2° e 3° são os pixels (dimensão da imagem) e o último é o canal de cores.

Para selecionar a primeira imagem (da matriz y) é realizado da seguinte maneira:

y\_true = y[0]

y\_true

Tendo como saída:

<tf.Tensor: shape=(4, ), dtype=float32, numpy=array([0., 0., 0., 1.], dtype=float32)>

Com o shape da imagem é aplicado a função np.argmax para retonar a posição do maior valor no conjunto de array, sendo o maior valor na posição 3 é que a classe [‘3\_diabetic\_retinapathy’], como mostra a imagem abaixo.

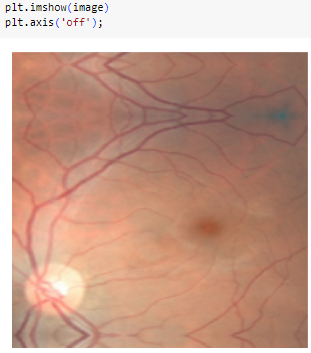


Figura 11: Imagem da 1 posição na matriz y

Fonte: Autor

Para mandar a imagem para a rede neural para ser feita a predição é necessário primeiro converter a imagem para o formato de lotes e depois fazer a classificação, como mostra a imagem abaixo

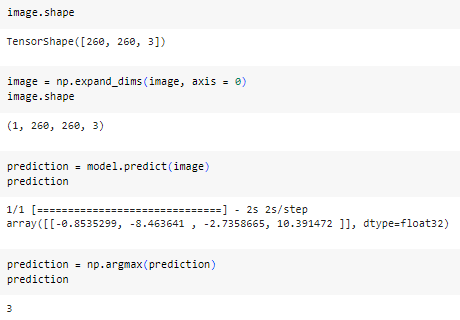


Figura 12: Conversão do formato para lotes e predição da imagem

Fonte: Autor

A predição retornou o valor 3, que é a mesma classe da imagem ([‘3\_diabetic\_retinapathy’]) que foi mandada para a rede neural fazer a predição.

**Conclusão**

Embora a acurácia tenha tido um valor relativamente alto, com 81%, esse valor ainda não é o suficiente para uma aplicação em um sistema de saúde, por se tratar de uma área bem delicada, onde várias pessoas dependem de uma análise precisa para não ficarem cegas, não é admissível usar uma ferramenta que não ofereça essa segurança para os pacientes.

Esse resultado só demostra que o projeto ainda tem muito que ser trabalhado e que no momento não é aplicável na área da saúde, como trabalho futuro será analisado formas de melhorar a acurácia, como por exemplo treinar o modelo por mais épocas, incluir mais imagens, testar outro modelo, entre ouros.

## Agradecimento

Agradeço a todos os professores que de alguma forma me transmitiram conhecimento, sou grato por todos aqueles que compartilharam um pouco do precioso tempo comigo. Agradeço especialmente à Natalia, minha namorada, pelo apoio incondicional a todo instante. Assim como Carl Sagan, eu também fico lisonjeado em compartilhar com todos vocês um planeta e uma época, Obrigado!

## Referências

A. Bosco et al.2005. Retinopatia Diabética. Arquivos Brasileiros de Endocrinologia. Volume 58

Disponível em: https://www.scielo.br/j/abem/a/cKy7w6RMzN64YMvbzngZRtg/?format=pdf&lang=pt

A. Krizhevsky et al. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research.

Disponível em: <https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer>,

E. A. McGlynn et al. 2003. The Quality of Health Care Delivered to Adults in the United States. The New England Journal of Medicine.

Disponível em: <https://www.nejm.org/doi/full/10.1056/nejmsa022615>

G. V. Doddi. Eye\_Diseases\_Classification. Eye Disease Retinal Images.

Disponível em: <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification>

Mingxing Tan, Quoc V. Le. 2021. EfficientNetV2: Smaller Models and Faster Training. Proceedings of the 38th International Conference on Machine Learning. Volume 139.

Disponível em: <http://proceedings.mlr.press/v139/tan21a.html>

R. A. P. Guedes. 2021.Glaucoma, saúde coletiva e impacto social. Revista Brasileira de Oftalmologia.

Disponível em: <https://www.scielo.br/j/rbof/a/kHLnFkWBc6jDWz3sQbvyhtR/>

R. Bourne et al.2020. Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: the Right to Sight: an analysis for the Global Burden of Disease Study.The Lancet Global Health.Volume 9.

Disponível em: <https://www.thelancet.com/JOURNALS/LANGLO/ARTICLE/PIIS2214-109X(20)30489-7/FULLTEXT>

V. O. Domingues et al.2016. Catarata senil: uma revisão de literatura. Revista de Medicina e Saúde de Brasília. Volume 5.

Disponível em: <https://portalrevistas.ucb.br/index.php/rmsbr/article/view/6756>

## Apêndice

O código gerado nesse trabalho foi armazenado em: <https://github.com/Paimonz/computer-vision/blob/main/ImageClassification.ipynb> com o intuído de aperfeiçoamentos futuros.