



**MBA
USP
ESALQ**

CLASSIFICAÇÃO DE IMAGENS NA TRIAGEM DE PACIENTES COM DEFICIÊNCIA VISUAL

Douglas Jonatas do Carmo Dias
Ana Beatriz Pereira Sette

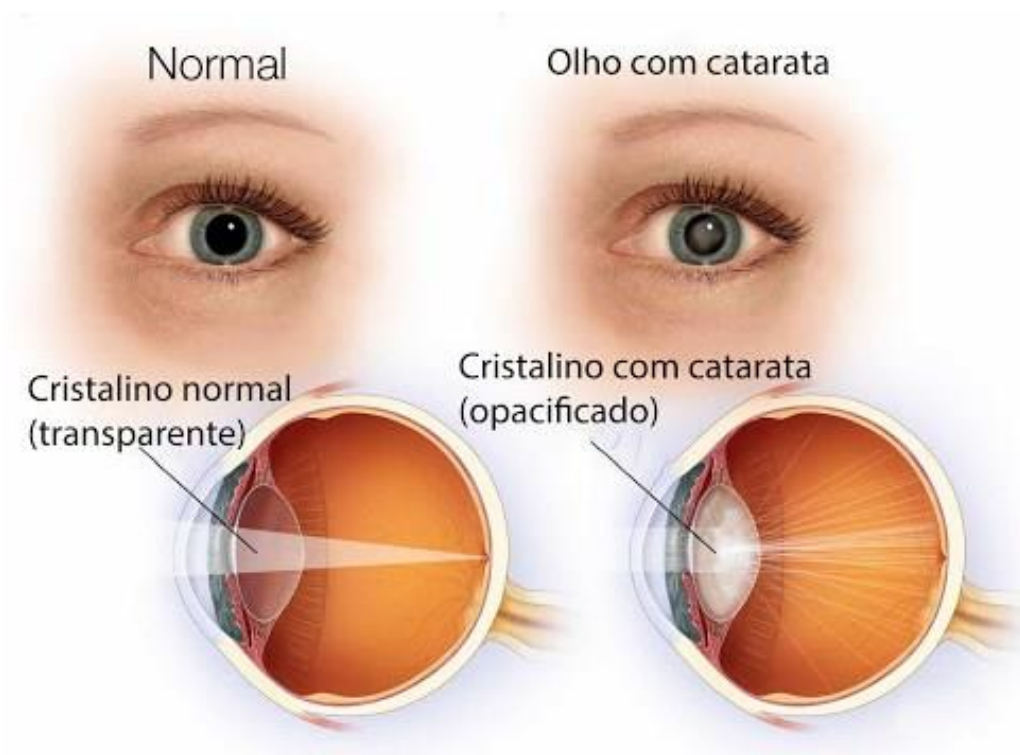


INTRODUÇÃO

- Deficiência Visual
- Catarata
- Retinopatia Diabética
- Glaucoma

INTRODUÇÃO

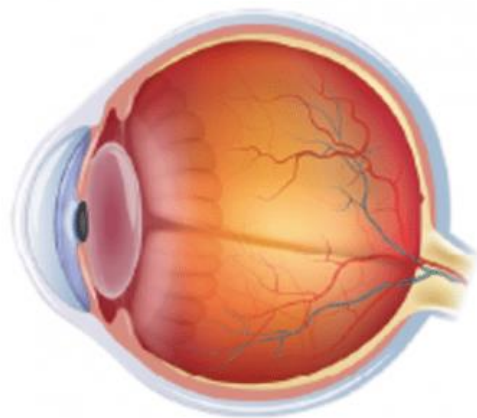
Catarata



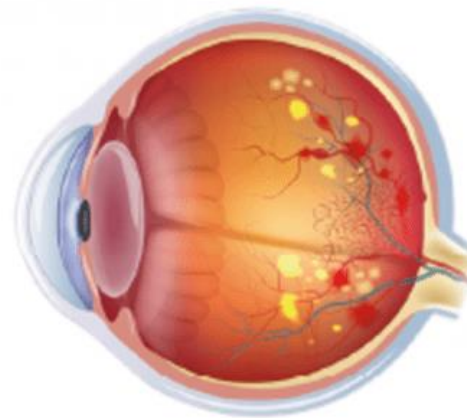
INTRODUÇÃO

Retinopatia Diabética

Olho Saudável

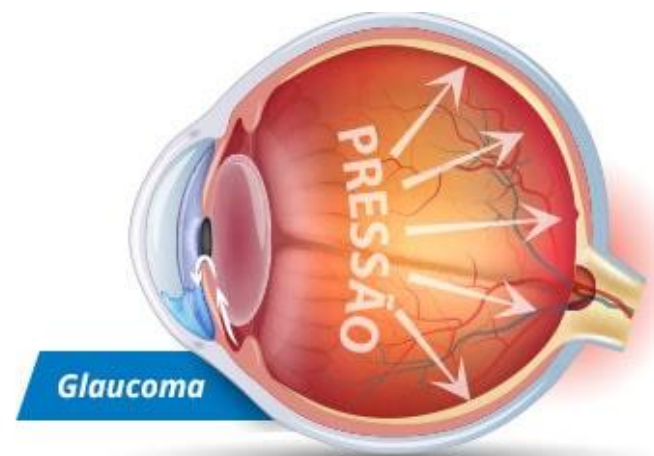
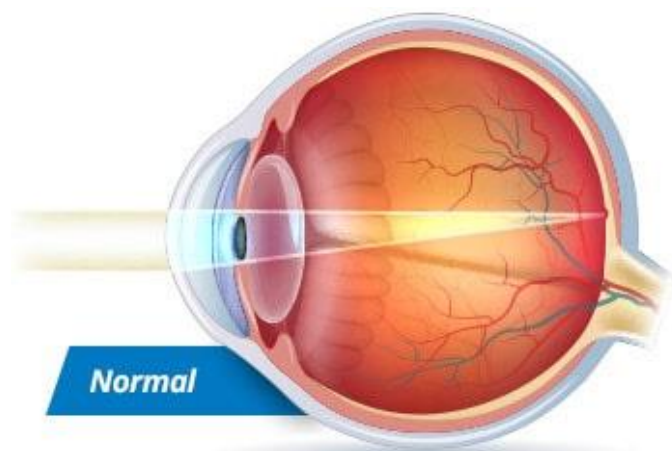


Olho com Retinopatia Diabética



INTRODUÇÃO

Glaucoma



MATERIAIS

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

MATERIAIS



DATASET



MÉTODOS

kaggle



Google Drive

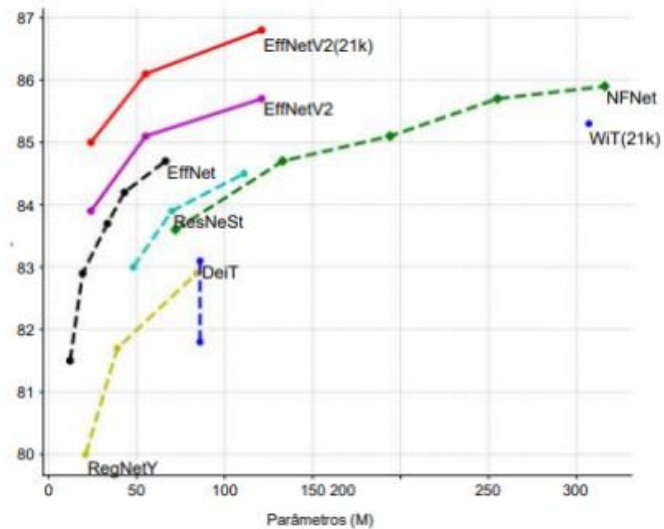
MÉTODOS

Criando e Treinando a Rede Neural

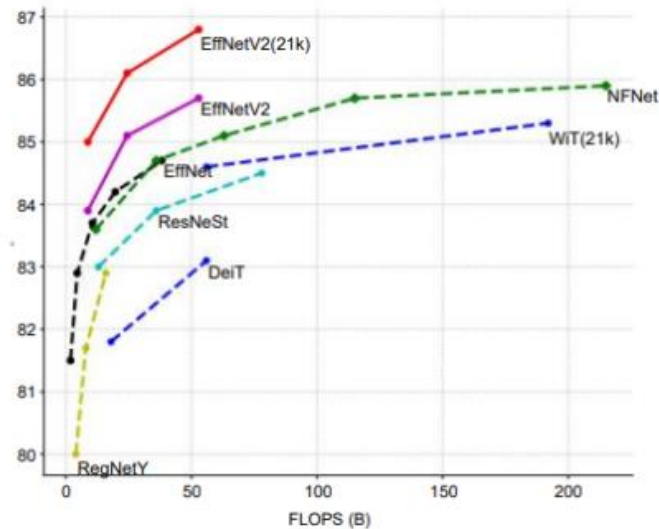
```
model = tf.keras.Sequential([  
    tf.keras.layers.InputLayer(input_shape= image_size + (3,)),  
    hub.KerasLayer(model_path, trainable = False),  
    tf.keras.layers.Dropout(rate = 0.2),  
    tf.keras.layers.Dense(len(classes))  
])
```

MÉTODOS

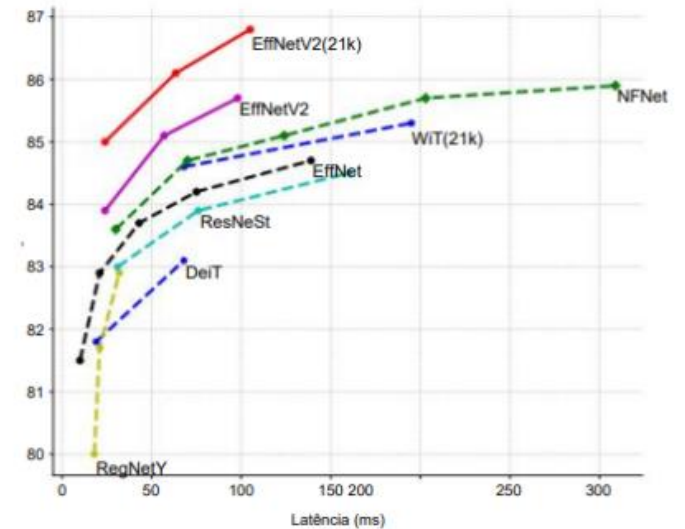
Efficientnetv2-b2-21k



(a) Parâmetros



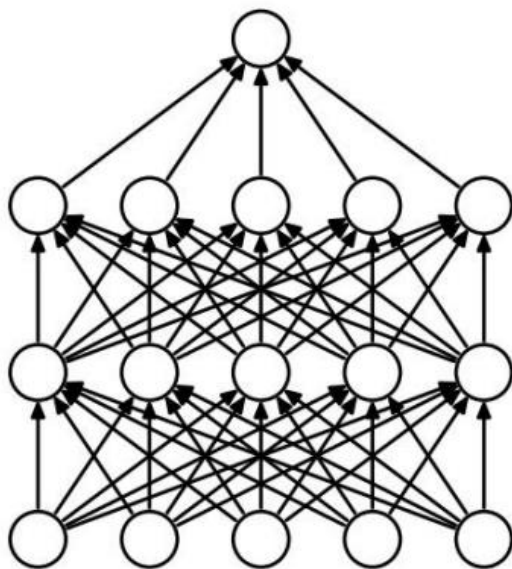
(b) FLOPs



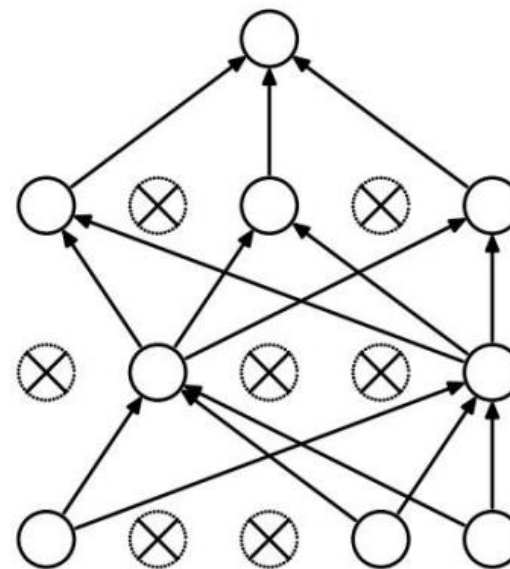
(c) Latência da GPU V100 (lote 16)

MÉTODOS

Dropout



(a) Rede Neural Padrão



(b) Depois de aplicar o abandono.

Modelo de Rede Neural de Dropout. Esquerda: Uma rede neural padrão com 2 camadas ocultas. Direita: Um exemplo de rede afinada produzida pela aplicação de dropout à rede à esquerda. Unidades cruzadas foram descartadas.

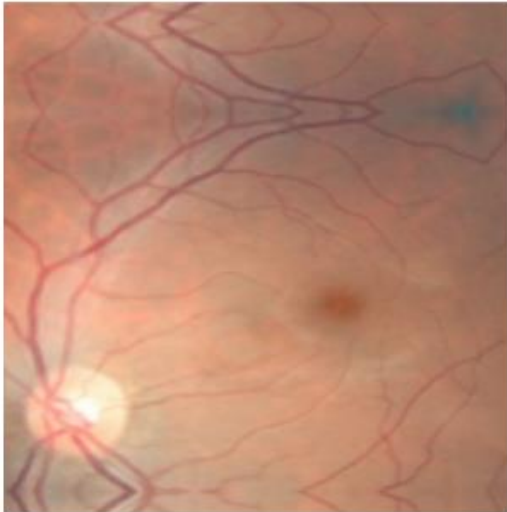
RESULTADOS E DISCUSSÃO

```
hist = model.fit(train_dataset, epochs= 14, steps_per_epoch = steps_per_epoch,  
                 validation_data = test_dataset, validation_steps = validation_steps).history
```

```
Epoch 1/14  
210/210 [=====] - 479s 2s/step - loss: 0.7020 - accuracy: 0.7217 - val_loss: 0.4925 - val_accuracy: 0.8137  
Epoch 2/14  
210/210 [=====] - 146s 693ms/step - loss: 0.6331 - accuracy: 0.7522 - val_loss: 0.4707 - val_accuracy: 0.8185  
Epoch 3/14  
210/210 [=====] - 156s 746ms/step - loss: 0.5748 - accuracy: 0.7781 - val_loss: 0.4472 - val_accuracy: 0.8185  
Epoch 4/14  
210/210 [=====] - 146s 696ms/step - loss: 0.5487 - accuracy: 0.7862 - val_loss: 0.4655 - val_accuracy: 0.7993  
Epoch 5/14  
210/210 [=====] - 149s 710ms/step - loss: 0.5184 - accuracy: 0.7993 - val_loss: 0.4751 - val_accuracy: 0.8149  
Epoch 6/14  
210/210 [=====] - 144s 689ms/step - loss: 0.5224 - accuracy: 0.7987 - val_loss: 0.5615 - val_accuracy: 0.7764  
Epoch 7/14  
210/210 [=====] - 168s 804ms/step - loss: 0.5158 - accuracy: 0.8023 - val_loss: 0.4602 - val_accuracy: 0.8161  
Epoch 8/14  
210/210 [=====] - 163s 781ms/step - loss: 0.5071 - accuracy: 0.8032 - val_loss: 0.4324 - val_accuracy: 0.8341  
Epoch 9/14  
210/210 [=====] - 92s 438ms/step - loss: 0.5198 - accuracy: 0.7996 - val_loss: 0.4003 - val_accuracy: 0.8618  
Epoch 10/14  
210/210 [=====] - 131s 627ms/step - loss: 0.4873 - accuracy: 0.8115 - val_loss: 0.4513 - val_accuracy: 0.8233  
Epoch 11/14  
210/210 [=====] - 132s 633ms/step - loss: 0.4851 - accuracy: 0.8177 - val_loss: 0.3963 - val_accuracy: 0.8474  
Epoch 12/14  
210/210 [=====] - 132s 633ms/step - loss: 0.4780 - accuracy: 0.8169 - val_loss: 0.4160 - val_accuracy: 0.8389  
Epoch 13/14  
210/210 [=====] - 167s 799ms/step - loss: 0.4809 - accuracy: 0.8169 - val_loss: 0.4060 - val_accuracy: 0.8401  
Epoch 14/14  
210/210 [=====] - 94s 450ms/step - loss: 0.4869 - accuracy: 0.8127 - val_loss: 0.4620 - val_accuracy: 0.8125
```

RESULTADOS E DISCUSSÃO

```
plt.imshow(image)  
plt.axis('off');
```



```
image = np.expand_dims(image, axis = 0) #Para enviar para a rede neural, precisamos converter esse formato para o form  
image.shape
```

```
(1, 260, 260, 3)
```

```
prediction = model.predict(image)  
prediction
```

```
1/1 [=====] - 1s 1s/step  
array([[ -0.2681191,  -9.446066 ,  -3.6260016,   5.7117705]], dtype=float32)
```

```
prediction = np.argmax(prediction) #Fazendo a classificação da imagem  
prediction
```

```
3
```

```
classes[3]
```

```
'3_diabetic_retinopathy'
```

RESULTADOS E DISCUSSÃO

Como resultados, foram obtidas algumas métricas oriundas do treinamento realizado, comprovando os resultados obtidos pela rede e observou-se uma de acurácia, precisão, sensibilidade igual a 81%. Apresentando assim um resultado inadequado para a classificação de imagens na triagem de pacientes com deficiência visual

CONSIDERAÇÕES FINAIS

Mesmo com resultado não satisfatório o mesmo será melhorado, pois a consequencia desse projeto abre portas para melhora-lo, como por exemplo testar outros modelos de redes convolucionais, aumentar o número de épocas no treinamento, acrescentar mais imagens no banco de dados e usar outras técnicas em conjunto com as que foram mencionadas.