



## Projeto Final 2 - Melanoma classification

Disciplina: INF-0619 – Projeto Final

**Grupo:** The Outliers

**Alunos:**

Paula Sampaio Meirelles

Raphael Mendes Motta

Renata Biaggi Barreto

Tailís Tavera Ferreira

## 1. Introdução: entendendo a problemática envolvida no desafio

O câncer é uma doença causada pela reprodução desordenada de células que sofreram algumas mutações genéticas e adquiriram a capacidade de se reproduzirem independentemente da sinalização bioquímica do corpo. Este descontrole da reprodução celular desregula o funcionamento saudável do organismo, invade outros tecidos do corpo (fase conhecida como metástase) podendo levá-lo a óbito.

O câncer de pele é o tipo mais frequente no Brasil (cerca de 30% das ocorrências) segundo o INCA (Instituto Nacional do Câncer) justamente porque a população é muito exposta aos raios ultravioletas que causam as mutações genéticas das células da pele. Dependendo do tipo de célula da pele que sofre a mutação, o câncer é denominado carcinoma basocelular, carcinoma espinocelular e melanoma.

A Figura 1 ilustra esquematicamente um corte transversal da pele, as três camadas que a compõe (epiderme, derme e hipoderme) e a ocorrência dos três tipos de câncer de pele.

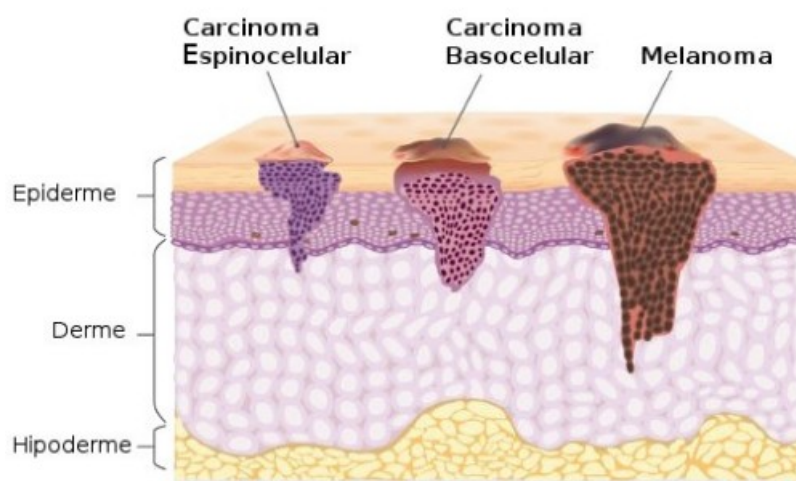


Figura 1: Esquema ilustrativo das camadas (epiderme, derme e hipoderme) da pele e dos tipos de câncer (carcinoma espinocelular, carcinoma basocelular e melanoma).

O carcinoma espinocelular afeta células superficiais da epiderme, como visto na Figura 1, o carcinoma basocelular atinge as células profundas da epiderme e o melanoma ocorre nos melanócitos (células responsáveis pela produção da melanina) que se situam na região basal da epiderme.

Dentre esses tipos, os carcinomas são os mais comuns (cerca de 90% das ocorrências, segundo o INCA) e os menos agressivos. O melanoma é o mais raro (cerca de 3% das ocorrências), porém é o mais agressivo, podendo rapidamente provocar metástase no organismo.

O diagnóstico médico de câncer de pele atualmente é feito pelo dermatologista através de um exame clínico. Para ajudá-lo no diagnóstico, às vezes, faz-se necessário utilizar o dermatoscópio, aparelho que amplia a imagem da lesão em 400 vezes permitindo enxergá-la melhor. Outras vezes, somente é possível o diagnóstico preciso por meio de biópsia e posterior exame histopatológico.

Dado o problema de identificação visual de uma lesão maligna pelo médico, foi demonstrado na literatura que o uso de algoritmos de deep learning com uma base de imagens de treino muito grande é

capaz de obter performances visuais no reconhecimento de objetos maiores que a dos próprios seres humanos [Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. & Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542, 115-118 (2017)].

O presente projeto se insere dentro dessa perspectiva. Utilizamos técnicas de deep learning para tentar identificar a malignidade de certas imagens médicas do conjunto de imagens fornecidas no site do Kaggle.

## 2. Datasets envolvidos

Obtidos no site do Kaggle:

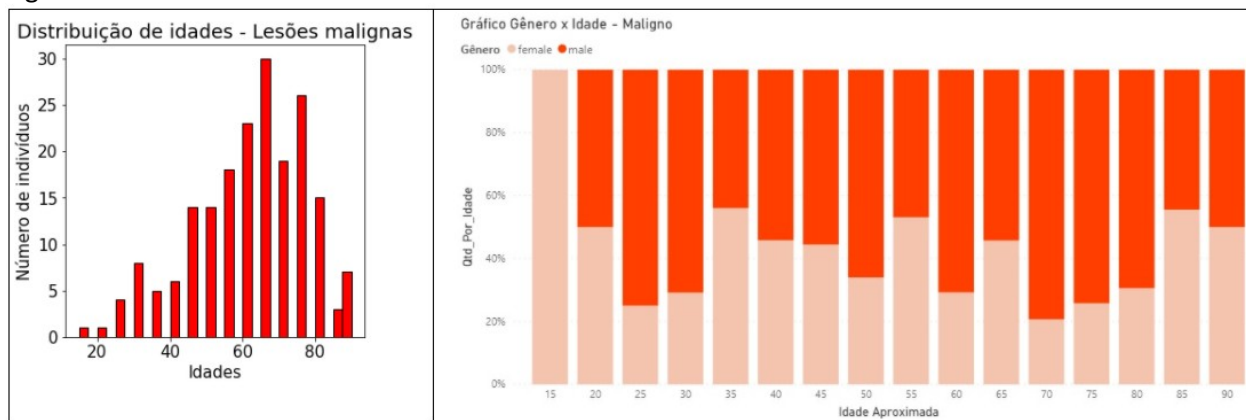
- Treino – Arquivo train.csv e 32.542 imagens benignas / 584 imagens malignas jpg de dimensões e resoluções variadas, no tamanho total de 23.9 GB.
- Teste – Arquivo test.csv e 10.982 Imagens jpg de dimensões e resoluções variadas, no tamanho total de 6,97 GB.

## 3. Features/Columns Originais dos datasets fornecidos pelo Kaggle

- image\_name - Nome da imagem
- patient\_id - Identificação única do paciente
- sex - Gênero do paciente
- age\_approx - Idade aproximada
- anatom\_site\_general\_challenge - Local da lesão
- diagnosis - Diagnóstico do paciente (presente apenas na base de treino)
- benign\_malignant - Indica se a lesão é maligna ou benigna (presente apenas na base de treino)
- target - 0 para lesão benigna / 1 para lesão maligna (presente apenas na base de treino)

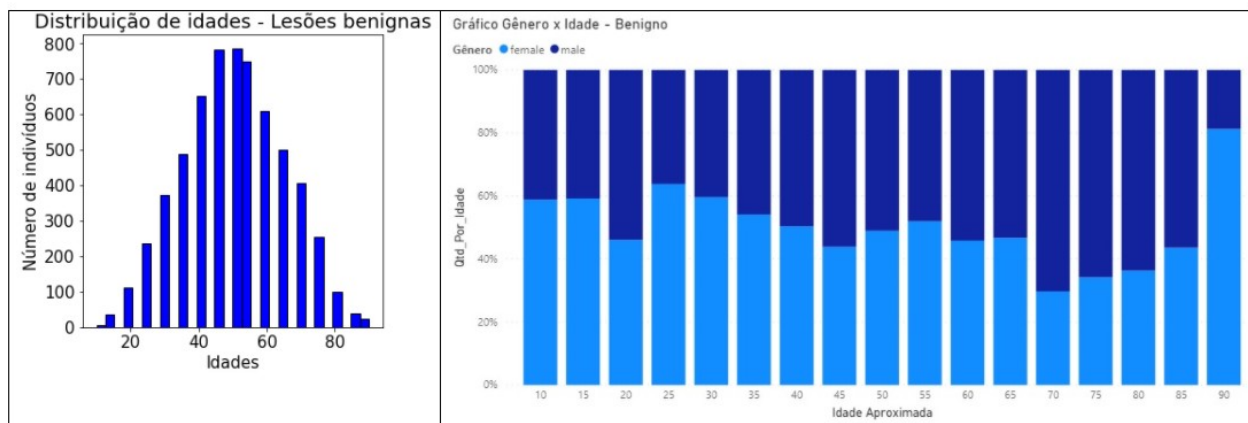
## 4. Análises Exploratórias

Considerando todas as imagens começando a analisando a distribuição por tipo de lesão, idade e gênero.



Gráficos 1 | 2 – Lesões malignas por idade | por gênero x idade

Observando os gráficos 1 e 2 podemos notar que temos maior incidência de casos malignos na faixa de 50 a 80 anos, em relação ao gênero nos chama atenção haver uma predominância masculina nas idades de 25 e 70 anos.



Gráficos 3 | 4 – Lesões benignas por idade | por gênero x idade

Observando os gráficos 3 e 4 podemos notar que temos maior incidência de casos benignos na faixa de 40 a 70 anos, em relação ao gênero ao comparar com os casos malignos é perceptível um aumento de casos no gênero feminino e diminuição no masculino.

Na sequência estudamos a distribuição das lesões no corpo (anatomia).

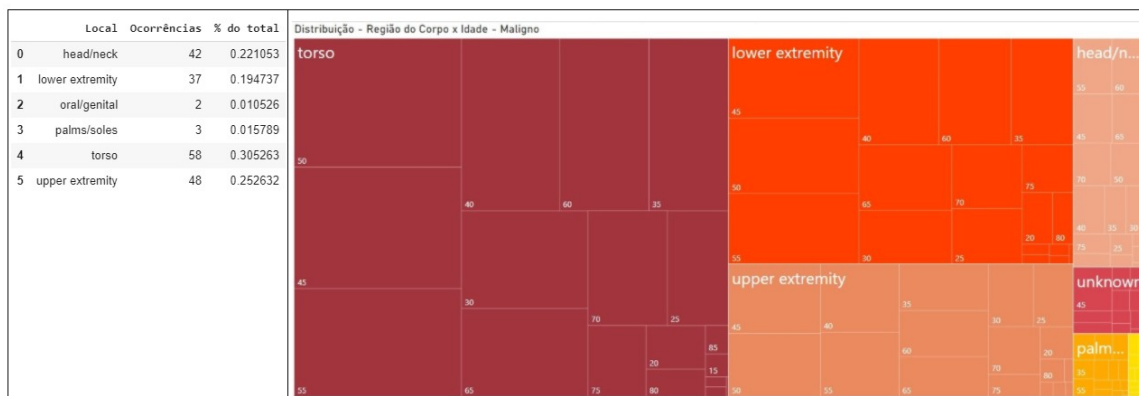


Tabela 1 | Gráfico 5 – Distribuição de lesões malignas pelo corpo

Visualizando o gráfico 5 notamos que a maioria das lesões malignas se concentram no “torso”, no tronco do corpo, seguido de baixas extremidades (regiões das pernas, joelhos, tornozelos e pés) e alta extremidades ( regiões do pescoço, ombros , braços e mãos) sendo a faixa de idade de 40 a 50 anos presentes como maioria de casos nessas regiões.



Tabela 2 | Gráfico 6 – Distribuição de lesões benignas pelo corpo

No gráfico 6 notamos o mesmo comportamento identificado para lesões malignas, ou seja, as lesões benignas em sua maioria se concentram no tronco do corpo, seguido de baixas extremidades e alta extremidades, porém, a faixa de idade com predominância em comum se concentra entre 60 e 65 anos.

## 5. Base de Treino x Amostragem de Treino

Notamos que as informações do dataset de treino carregados do arquivo train.csv, em alguns casos, possuíam um ou mais features com valores ausentes, de forma que retiramos do dataset de treino as referências de imagens que tivesse essa situação, com esse tratamento, da quantidade original da de 33.126 passamos a considerar 32.531 imagens.

Utilizamos o Google Colab com configurações padrão ( Ram de 13,3 GB) como ambiente de desenvolvimento dos modelos e ao tentarmos carregar essas 32.531 imagens de treino, tivemos problemas de limitação física, falta de RAM conforme mensagem da imagem 1. Pesquisamos na Internet e testamos alguns procedimentos para extensão de RAM, sem sucesso.

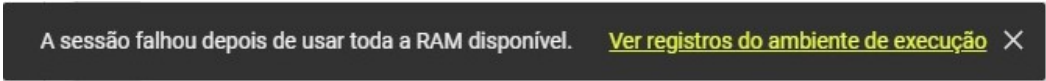


Figura 2 – Mensagem de Erro durante a execução do código Python no Google Colab ao tentar carregar todas as imagens de treino

Dessa maneira, percebemos que não seria possível processar todas as imagens, e traçamos algumas estratégias de amostragem.

### Amostra Estratificada Proporcional

Em todas as amostragens que geramos para as imagens benignas criamos uma feature “chave”, chamada comb\_features, com a combinação das features originais: image\_name, sex, age\_approx, anatom\_site\_general\_challenge, benign\_malignant e target com o intuito de selecionar uma amostra estratificada proporcional.

Ou seja, uma amostra que tivesse as imagens benignas com todas as variações dessas features e na proporção existente entre a quantidade total de imagens benignas considerada em cada caso de amostragem. Por exemplo, se no total tivéssemos 4 imagens na anatomia “torso” e 2 na anatomia “lower extremity” e gerássemos uma amostra de 50%, teríamos como resultado 2 imagens na anatomia “torso” e 1 na anatomia “lower extremity”.

**Amostragem 1** – Notamos que na base de treino havia mais de uma imagem relacionada ao mesmo paciente, em alguns casos referenciando a mesma idade, em outros em idades distintas, optamos por selecionar uma imagem de cada paciente , considerando a imagem em que a maior idade estivesse referenciada, isso fez com que amostra fosse reduzida à 6131 imagens.

Sendo que dessas, apenas 179 eram imagens reconhecidas como malignas na base de treino. Portanto, nesse cenário a quantidade total de imagens benignas é igual 5952. Aplicamos um train split com a feature comb\_features para obter uma amostra estratificada proporcional de 179 imagens benignas.

Logo nessa primeira amostragem nossa base de treino ficou composta por 179 benignas e 179 malignas, totalizando 358 imagens.

**Amostragem 2** - Gerado uma amostra desbalanceada de 537 imagens, 179 imagens malignas e 1321 benignas, ou seja, mantendo as imagens malignas e dobrando a quantidade de imagens benignas, não houve melhora de desempenho nos modelos já testados.

**Amostragem 3** – Procuramos gerar uma amostragem com uma quantidade maior de imagens : 2298, aplicando o tratamento de recorte central das imagens num tamanho maior de 280 x 280 essas foram as configurações máximas de imagem e tamanho que conseguimos executar sem que o Colab apresentasse falha de execução por conta de RAM.

### **Outras Amostragens Geradas x Limitação de RAM na execução do Projeto**

- Gerado uma amostra desbalanceada com todas as imagens malignas (173) + benignas (5893) = 6.072.
  - > Google Colab apresentou falha de RAM ao tentar tratamento de recorte central das imagens no tamanho 400 x 400.
- Gerado uma amostra com 2500 imagens
  - > Proporção: ( 1 img maligno x 13 img benignos ) - 179 img malignos x 2.321 img benignos
  - > Google Colab apresentou falha de RAM após tratamento de recorte central das imagens no tamanho 400 x 400
- Gerado uma amostra com 1500 imagens
  - > Proporção: ( 1 img maligno x 7.37 img benignos ) - 179 img malignos x 1.321 img benignos
  - > Aplicado tratamento de recorte central das imagens no tamanho 400 x 400
  - > Google Colab apresentou falha de RAM ao tentar executar a primeira rede neural que possui duas camadas

## **6. Tratamento de Imagens**

Devido os problemas de limitação de recurso de RAM no Colab, adotamos a estratégia de reduzir as imagens efetuando cortes de forma centralizada nas imagens, para tal geramos 3 tratamentos diferentes com esse mesmo objetivo procurando identificar qual desses tratamentos traria melhor desempenho quando as imagens fossem aplicadas nos modelos.

- **Tratamento 1** - Aplicamos as médias entre as diferenças entre tamanho atual e novo tamanho nas dimensões na largura e altura aplicando-as à direita, esquerda, acima e abaixo. Efetuando o comando crop da library PIL para efetuar o recorte da imagem.
- **Tratamento 2** - Aplicamos diretamente o comando crop da library PIL para efetuar o recorte da imagem nas dimensões (400 x 400 x 400 x 400).
- **Tratamento 3** - Aplicamos Keras Target\_Resize com as dimensões (400,400,3) no momento do "load" da imagem.

Abaixo podemos visualizar duas imagens escolhidas de forma aleatória:

- > Imagem 256 - ISIC\_2255411 - IP\_7868912/ male/ 65.0 /upper extremity/ melanoma/ malignant/ 1
- > Imagem 18 - ISIC\_8319711 - IP\_2290360/ male / 450 / lower extremity/ nevus /benign/ 0

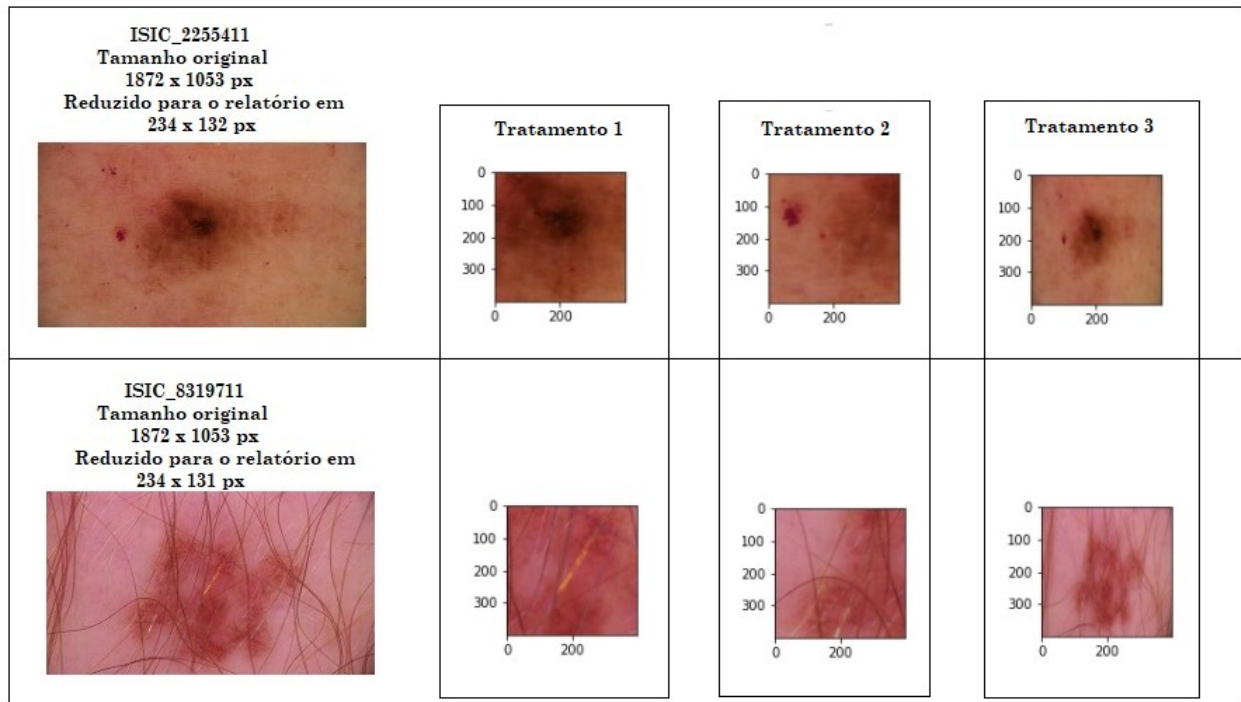


Figura 3 –Imagens de exemplos antes e após os 3 tipos de tratamento de imagens ( recorte central) aplicados

Ao aplicar as imagens nos modelos notamos uma desempenho um pouco melhor quando utilizadas as imagens no qual o tratamento 1 foi aplicado.

## 7. Modelos

Após a preparação de dados descrita, dividimos a amostra de treino em dois conjuntos: treino e validação. 80% dos dados de (X\_train, y\_train) formam o novo conjunto de treino. Isso resultou em 286 dados do novo treino e 72 dados de validação.

A seguir, faremos um breve resumo do que foi realizado nos modelos e nossas principais estratégias para melhorar as métricas. Nos modelos, utilizamos sempre a cross-entropy e o otimizador SGD com duas diferentes taxas de learning rate de 0.001 e 0.001 e dois valores de 'momentum': 0.5 e 0.9. Além disso, sempre foi utilizado um batch de 32 e 100 épocas. Foram realizados, no total, 13 experimentos.



## Baseline (experimento 1)

Como o nosso baseline, resolvemos escolher uma arquitetura já implementada e pré-treinada na ImageNet. No site <https://keras.io/api/applications/>, escolhemos, inicialmente, a ResNet50 para usar seus pesos. A escolha baseou-se na performance desse modelo pré-treinado com os datasets de validação da Imagenet. Descartamos a saída original e congelamos as camadas. Em seguida, adicionamos uma nova camada de saída com 2 classes.

Nesse experimento, utilizamos um valor de 0.5 para o 'momentum' e as imagens submetidas ao tratamento 1.

O valor obtido da métrica AUC, nessa configuração para o conjunto de validação, foi de 0.63. A curva da figura 2 exibe o comportamento da curva loss versus o número de épocas do modelo baseline.

A escala de 0.025 do gráfico das curvas loss exibe uma grande variação nos valores da loss tanto para o treino quanto para a validação. Se considerarmos essa escala podemos verificar que os níveis da curva azul (treino) e da curva laranja (validação) estão ligeiramente descoladas uma da outra o que poderia indicar algum grau de "overfitting". Entretanto, não podemos fazer com precisão tal afirmação porque a definição formal de overfitting é feita somente através de uma investigação estatística, não através de um gráfico que recai, muitas vezes, em subjetividade.

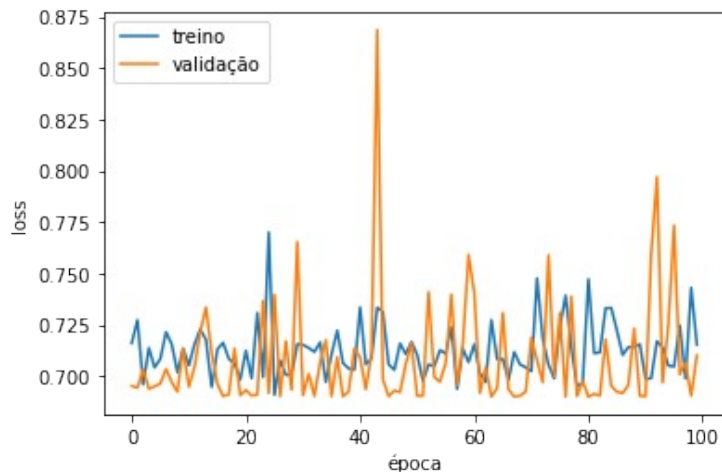


Figura 2: Curva loss x número de épocas do modelo baseline

A Figura 3 exibe os valores de acurácia obtidos na última época de treinamento do modelo que foi aproximadamente 0.52.

```
Epoch 69/100  
8/8 [=====] - 3s 394ms/step - loss: 0.7057 - accuracy: 0.5118 - val_loss: 0.6904 - val_accuracy: 0.5278
```

Figura 3: Acurácias do modelo baseline na última época de treinamento para os conjuntos de treino e validação.

## Experimento 2

Como queríamos investigar o papel dos tratamentos realizados sobre as imagens, resolvemos usar o mesmo modelo baseline, porém com imagens tratadas com o tratamento 3. Observamos que o



comportamento das curvas loss foi parecido com o primeiro experimento (baseline) na mesma escala bem como os valores de acurácia dos dois conjuntos ao final da época 100. Entretanto, o valor de AUC foi menor: 0.44.

### Experimento 3 e 4

Resolvemos nesses experimentos usar outra rede muito utilizada na literatura para treinamento. Usamos a VGG16 também com camadas congeladas, porém com imagens com tratamento 1 e 3 respectivamente. Observamos um comportamento interessante em relação ao overfitting conforme ilustrado na figura 4 para o experimento 3. A rede overfita em torno da época 40. Isso pode ter ocorrido por conta da maior complexidade da rede (a VGG16 possui muito mais parâmetros que a ResNet50).

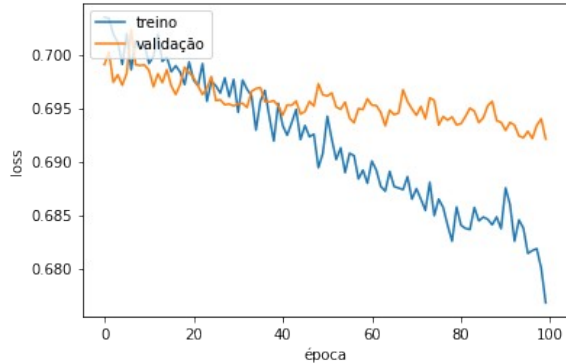


Figura 4: Curva loss x número de épocas do experimento 3.

Tentamos várias outras estratégias, como podem ser vistas no código em anexo, usando aumento de dados, alterando o corte das imagens e testando outros valores para o learning rate e momentum. Nenhuma dessas estratégias performaram melhor que o nosso modelo baseline. A tabela, a seguir, exhibe resumidamente os principais resultados que obtivemos nesse projeto.

Nr. Exp.	Rede Neural	Roc_Auc Score Validação (20% de Treino)	Tratamento Considerado
1	ResNet50 com camadas congeladas	0.63	Imagens com tratamento 1
2	ResNet50 com camadas congeladas	0.44	Imagens com tratamento 3
3	VGG16 com camadas congeladas	0.42	Imagens com tratamento 1
4	VGG16 com camadas congeladas	0.4	Imagens com tratamento 3
5	Usamos VGG16 com fine-tuning com as imagens do tratamento 1, camada densa com função de ativação relu e com uma menor taxa de aprendizado (0.0001) e mesmo momentum.	0.54	Imagens com tratamento 1
6	ResNet152 com camadas congeladas	0.43	Imagens com tratamento 1
7	ResNet50 com camadas congeladas com diferentes taxa de aprendizado e momentum	0.52	Imagens com tratamento 1
...			
13	ResNet50 com camadas congeladas	0.49	Alteramos a quantidade de imagens de amostras de 358 para 2300 imagens recortadas e centralizadas no formato 240 x 240. Chegamos a aumentar o formato da imagem para 260 x 260, mas acurácia ROC foi inferior 0.465. Quando aumentamos para 300 x 300, tivemos problemas de falta de Ram no Colab.

Ressalva: Em relação a submeter um dos modelos à base de teste, encontramos problemas limitação de IO do Colab para efetuar load de todas as imagens de uma vez.

Adotamos uma segunda estratégia : criamos 2 dataset com metade das imagens cada um e unificamos em um único array, porém, ao executar a rede neural com esse array unificado, tivemos problema de falta de RAM no Colab.

A terceira estratégia foi gerar uma amostra de apenas 359 imagens de testes , a mesma quantidade de imagens de treino adotadas em vários modelos, porém, não submetemos essa amostragem em uma rede neural.

Para conhecimento, a submissão dessa competição foi encerrada no Kaggle no dia 17/08/2020, abaixo temos as três primeiras posições com os Scores ROC mais altos.




#	△pub	Team Name	Notebook	Team Members	Score	Entries	Last
1	▲ 885	All Data Are Ext	Mean Columnwise Area Under Receiver Operating Characteristic Curve			116	6d
1	▲ 885	All Data Are Ext			0.9490	116	6d
2	▲ 56	aloe			0.9485	61	6d
3	▲ 264	Deloitte Analytics Spain			0.9484	118	8d

Figura 4 – Ranking e métrica utilizada pelo Kaggle