

# Universidade Federal do Rio de Janeiro



UNIVERSIDADE FEDERAL  
DO RIO DE JANEIRO

## Redes Neurais

Projeto 2

Aluno	Rodrigo Malta Esteves
Professora	Carlos Tadeu Pagani Zanini
Horários	Sáb - 13:00-17:00

Rio de Janeiro, 2022

# 1 Descrição do problema

Serão analisadas as imagens obtidas pela nave espacial Magellan da Nasa em 10 de Agosto de 1990. O objetivo principal da missão foi mapear a superfície de Venus e determinar a topografia do planeta. Analizando os dados obtidos pela nave, foram encontrados vulcões na superfície de Venus. Esses vulcões podem ser usados para criar uma forma de detecção automática de vulcões.

Os conjuntos de treino e teste já foram separados previamente. Cada imagem tem dimensão 110x110, com pixels entre 0 e 255, onde cada imagem é uma linha das 12100 colunas. A imagem pode ter entre 0 e 4 vulcões. Aqui são apresentados os atributos das imagens:

- Volcano?: se na imagem tem algum vulcão (objetivo principal de classificação), 1 ou 0. Para Volcano?=0 os outros três atributos são NaN.
- Type: 1= definitivamente um vulcão, 2= provavelmente, 3= possivelmente, 4= só um buraco é visível
- Radius: O raio do vulcão no centro da imagem, em pixels
- Number Volcanoes: Número de vulcões na imagem

Os dados foram obtidos pelo site:

<https://www.kaggle.com/fmena14/volcanoesvenus>.

# 2 Análise exploratória

Primeiramente, observamos a proporção da existência de vulcões no conjunto de treino. Apenas 1000 das 7000 imagens, aproximadamente 14% (0.142857) possuem algum vulcão nelas, enquanto 6000, cerca de 85% (0.857143), não possuem nenhum. Entre as imagens com vulcões, 105 possuem quatro vulcões, 187 possuem três vulcões, 346 possuem dois vulcões e 362 possuem apenas um vulcão. Já no conjunto de teste, das 2734 imagens temos que apenas 434 possuem algum vulcão. Apesar da diferença grande entre os valores absolutos em cada conjunto, a proporção entre as duas categorias é muito semelhante. A figura 1, a seguir, mostra os gráficos de barra referentes à essas proporções.

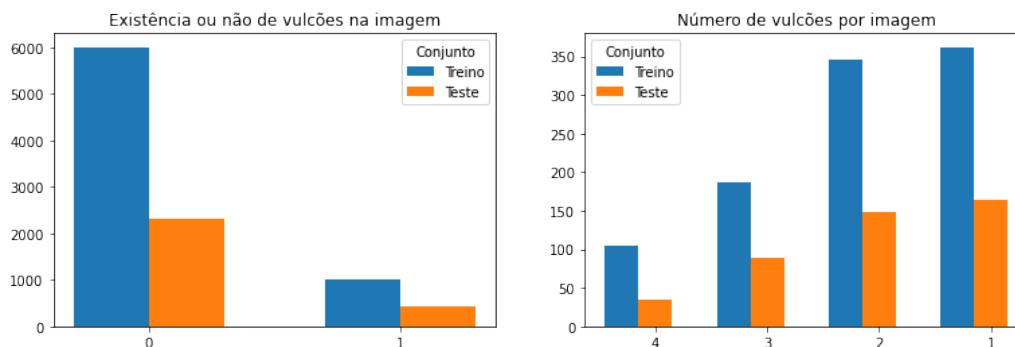


Figura 1: Proporções

Além disso, é interessante observar algumas das imagens. A figura 2 mostra na primeira linha alguns exemplos de imagens com pelo menos um vulcão e na segunda imagens que não possuem nenhum.

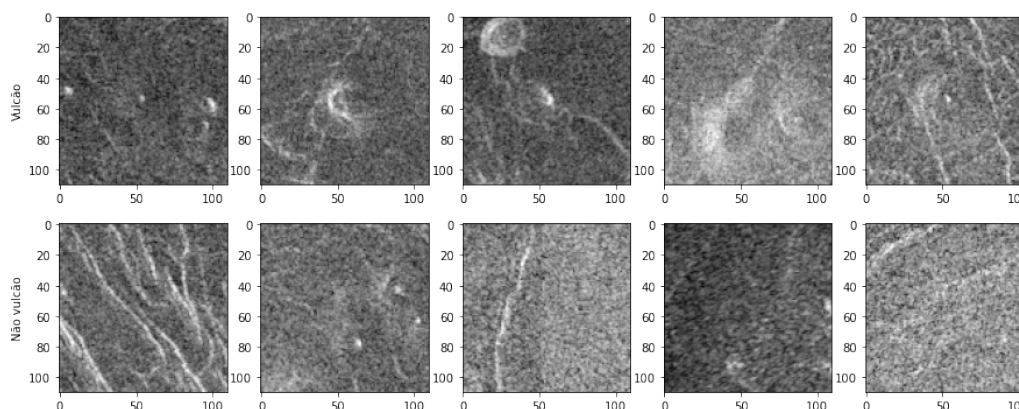


Figura 2: Vulcões?

### 3 Pré-processamento

Nesta etapa alteramos a escala dos pixel para que os valores ficassem entre 0 e 1. Para isso dividimos os os conjuntos de treino e teste por 255 (referente ao valor total de pixels possíveis).

O conjunto de treino também foi dividido utilizando o *train\_test\_split* do *sklearn*, para que 20% fosse reservado para um conjunto de validação.

## 4 Modelagem

Dada a baixa capacidade de processamento disponível (meu notebook é uma batedeira), apenas variações de um modelo de CNN (Convolutional Neural Network) e um modelo de Regressão Logística Simples foram treinados nesse trabalho.

Os modelos de redes neurais foram treinados para 100 épocas, batch size 64 e regularização Dropout com taxa 0.5. Para tentar evitar alguma forma de sobreajuste e diminuir o tempo de processamento, também foi utilizado EarlyStopping com patience igual a 15 (Número de época sem melhoras até o treinamento parar).

### 4.1 Regressão Logística Simples

As Tabela 1 e Tabela 2 apresenta os resultados da regressão para os dados de validação e teste. Observamos uma precisão boa de 93% e uma alta capacidade (98%) em detectar falsos positivos, mas o recall é de apenas 64% no conjunto de validação. Como nosso objetivo é ser capaz de detectar os vulcões nas imagens, parece interessante buscar um modelo que tenha uma taxa maior de detecção de verdadeiros positivos. Além disso, um ponto positivo desse modelo foi o tempo de processamento de apenas 8 segundos.

Tabela 1: Validation Report

	precision	recall	f1-score	support
0	0.94	0.98	0.96	1199
1	0.86	0.64	0.73	201
Accuracy			0.93	1400
Macro Avg	0.90	0.81	0.85	1400
Weighted Avg	0.93	0.93	0.93	1400

Tabela 2: Testing Report

	precision	recall	f1-score	support
0	0.94	0.98	0.96	2300
1	0.84	0.65	0.73	434
Accuracy			0.93	2734
Macro Avg	0.89	0.81	0.84	2734
Weighted Avg	0.92	0.93	0.92	2734

Para ilustrar os resultados, podemos observar, a seguir, os plots comparativos entre a precisão e o recall.

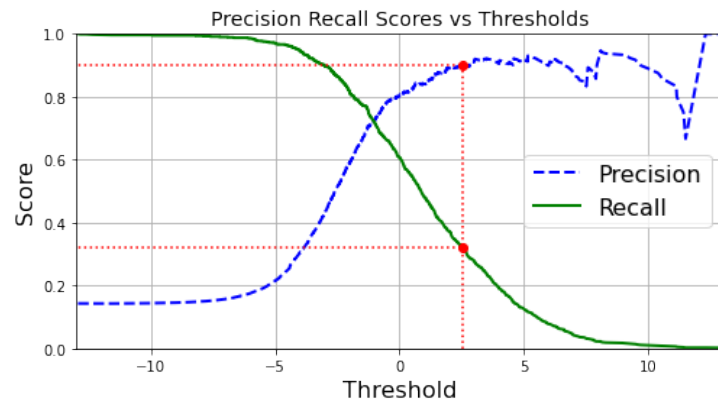


Figura 3: Precision Recall Scores vs Thresholds

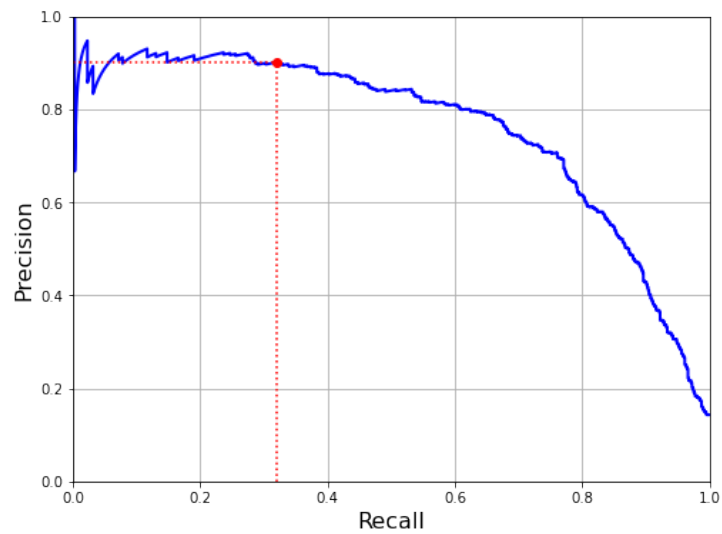


Figura 4: Recall x Precision

## 4.2 CNN com 2 camadas latentes

Observamos melhoras nesse modelo em relação ao modelo de regressão. Nossa acurácia de validação subiu para 95% e houve melhora tanto na precisão quanto no recall. A Figura 5 mostra que os erros de treino e teste estão próximos e diminuindo conforme o tempo aumento, indicando que não é sobreajuste ou subajuste. A acurácia dos dois conjuntos também parece aumentar com o passar das épocas.

Tabela 3: Validation Report

	precision	recall	f1-score	support
0	0.95	1.00	0.97	1206
1	0.96	0.70	0.81	194
Accuracy			0.95	1400
Macro Avg	0.96	0.85	0.89	1400
Weighted Avg	0.95	0.95	0.95	1400

Tabela 4: Testing Report

	precision	recall	f1-score	support
0	0.95	0.99	0.97	2300
1	0.95	0.72	0.82	434
Accuracy			0.95	2734
Macro Avg	0.95	0.86	0.90	2734
Weighted Avg	0.95	0.95	0.95	2734

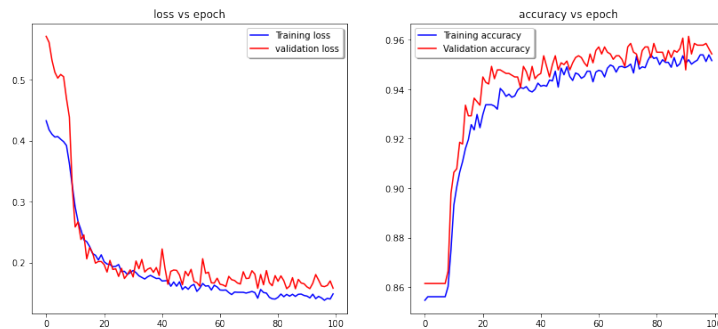


Figura 5: Modelo 1

### 4.3 Balanceando as classes

Na primeira etapa de análise exploratória percebemos que as classes não estão balanceadas. Existem quase quatro vezes mais imagens com um vulcão do que com quatro vulcoes. Aqui foram atribuidos pesos diferentes para essas classes a fim de levar esse desbalanceamento em conta.

Podemos observar que houve uma melhora no conjunto de validação em relação à detecção de verdadeiros positivos. Nosso recall aumento de 0.7 para 0.8, enquanto mantivemos a mesma acurácia.

Tabela 5: Validation Report

	precision	recall	f1-score	support
0	0.97	0.97	0.97	1206
1	0.81	0.80	0.80	194
Accuracy			0.95	1400
Macro Avg	0.89	0.88	0.89	1400
Weighted Avg	0.95	0.95	0.95	1400

Tabela 6: Testing Report

	precision	recall	f1-score	support
0	0.97	0.97	0.97	2300
1	0.83	0.82	0.83	434
Accuracy			0.95	2734
Macro Avg	0.90	0.90	0.90	2734
Weighted Avg	0.95	0.95	0.95	2734

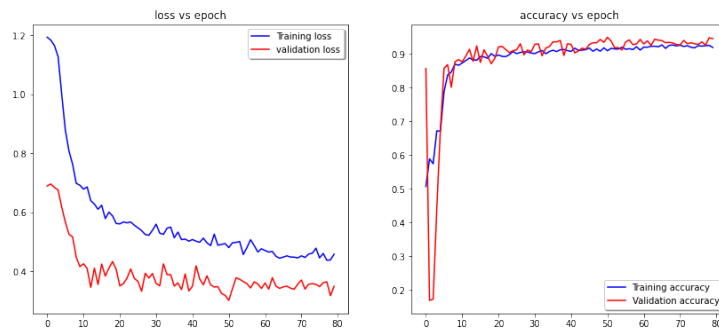


Figura 6: Modelo 1

## 4.4 Utilizando data augmentation

Aqui foi tentada uma implementação de data augmentation como sugerido no *Tema 4* do pdf.

Nesse caso, há um aumento marginal no recall, mas uma perda grande na precisão e na acurácia.

Tabela 7: Validation Report

	precision	recall	f1-score	support
0	0.97	0.94	0.95	1206
1	0.69	0.82	0.75	194
Accuracy			0.92	1400
Macro Avg	0.83	0.88	0.85	1400
Weighted Avg	0.93	0.92	0.93	1400

Tabela 8: Testing Report

	precision	recall	f1-score	support
0	0.97	0.94	0.96	2300
1	0.74	0.87	0.80	434
Accuracy			0.93	2734
Macro Avg	0.86	0.91	0.88	2734
Weighted Avg	0.94	0.93	0.93	2734

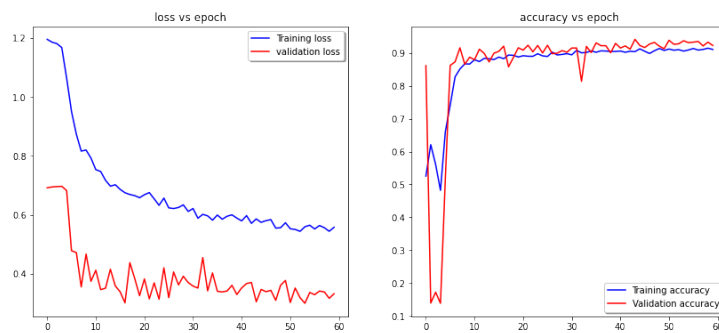


Figura 7: Modelo 1



## 4.5 Aumentando a complexidade do modelo

Nesse caso foi adicionado um layer de complexidade ao problema. Agora utilizamos três camadas latentes, ao invés de apenas duas.

Observamos uma melhora em relação ao recall, onde agora temos quase 90% de precisão, e mantemos a acurácia semelhante às dos casos anteriores por volta de 95%.

Tabela 9: Validation Report

	precision	recall	f1-score	support
0	0.98	0.97	0.97	1206
1	0.81	0.88	0.84	194
Accuracy			0.95	1400
Macro Avg	0.89	0.92	0.91	1400
Weighted Avg	0.96	0.95	0.96	1400

Tabela 10: Testing Report

	precision	recall	f1-score	support
0	0.98	0.97	0.98	2300
1	0.85	0.90	0.87	434
Accuracy			0.96	2734
Macro Avg	0.91	0.94	0.92	2734
Weighted Avg	0.96	0.96	0.96	2734

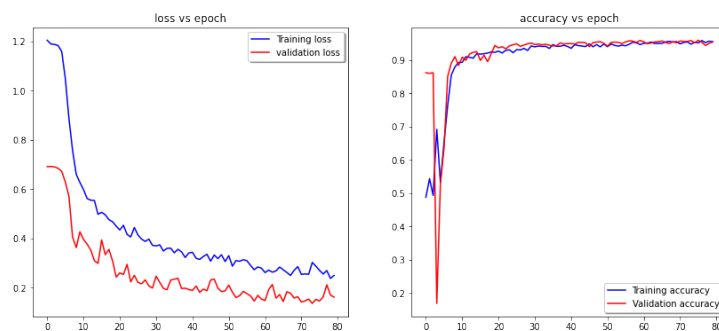


Figura 8: Modelo 1

## 5 Redução de ruído

Para reduzir o ruído das imagens foi utilizado um *Deep Convolutional Autoencoder*. Para aplicar esse Autoencoder foi necessário alterar a dimensão das imagens de 110x110 para 128x128. Por algum motivo, o compilador alterava a dimensão de entrada do conjunto de treino e levava a um erro.

A Figura 9 mostra alguns exemplos de imagens originais do conjunto de treino, a Figura 10 mostra as mesmas imagens codificadas pelo autoencoder e a Figura 11 mostra as mesmas imagens suavizadas após remover o ruído.

original volcanos

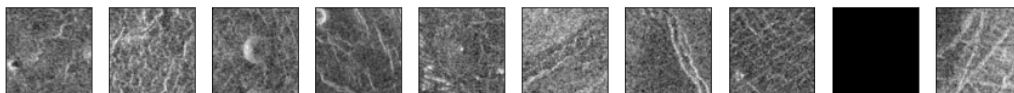


Figura 9: Original Volcanos

encoded volcanos



Figura 10: Encoded Volcanos

reconstructed volcanos

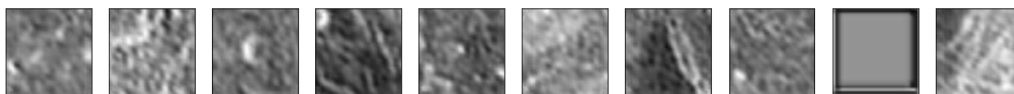


Figura 11: Reconstructed Volcanos

## 6 Modelando com Autoencoder

Rodamos modelo mais complexo utilizado anteriormente para as imagens suavizadas pelo autoencoder. Não foram observados resultados superiores. O recall caiu de .88 para .77 e a acurácia foi de 95% para aproximadamente 93%. Um ponto positivo foi o tempo de processamento que caiu para um terço do original.

Tabela 11: Validation Report

	precision	recall	f1-score	support
0	0.96	0.96	0.96	1206
1	0.77	0.75	0.76	194
Accuracy			0.94	1400
Macro Avg	0.87	0.86	0.86	1400
Weighted Avg	0.93	0.94	0.93	1400

## 7 Comparando os modelos

Comparando os modelos gerados, vemos que a CNN com 3 camadas foi a melhor tanto na acurácia quanto na capacidade de prever verdadeiros positivos e, portanto, deveria ser o modelo escolhido caso desejássemos o melhor ajuste. É importante notar destacar que o tempo de processamento para ele foi extremamente grande, o que poderia ser um impedimento.

Tabela 12: Comparação das acurácias de validação

Modelo	Acurácia	Recall
Regressão Logística Simples	0.9328	0.64
CNN com duas camadas	0.9542	0.70
Classes balanceadas	0.9457	0.80
Data augmentation	0.9228	0.82
CNN com 3 camadas	0.9542	0.88
Redução de ruído	0.935	0.75

## 8 Alguns comentários

1. Tive um problema importante durante o trabalho. Os meus gráficos de 'epoch x loss' apresentaram um tipo de problema que eu nunca tinha visto antes. O meu erro de treino foi maior que meu erro de validação. Eu sei que se o erro de treino é muito menor que o de validação, então tenho um problema de sobre ajuste, e se os dois erros estão próximos e com valor alto eu tenho um problema de subajuste. Nesse caso não faço ideia do que pode ter ocorrido.

Quando busquei soluções para esse problema, li que o Dropout do Keras é aplicado no conjunto de treino ao final de cada época, enquanto no conjunto de validação ele só é aplicado no final de todas as iterações. Aparentemente é um erro normal e que ocorre com alguma frequência (nunca tinha ouvido falar). Gostaria de receber algum feedback em relação a isso se fosse possível.

2. Acho que era esperado que após a redução de ruído a precisão do modelo fosse superior, mas por algum motivo isso não aconteceu. Talvez se eu tivesse rodado mais iterações isso tivesse acontecido, mas não me parece o caso. Algum problema na manipulação do conjunto de treino deve ser o mais provável.

3. Como não consegui rodar muitos modelos, deixei o último tópico que pedia um ensemble dos modelos por fazer.