

# Classificação de imagens radiográficas utilizando Rede Neural Convolucional

Michel Motta  
Thiago Machado


# Resumo

Este trabalho tem por objetivo, mostrar de maneira geral as potencialidades da Inteligência Artificial na análise de imagens. Demonstraremos isto através de um exemplo com análise em imagens radiográficas disponibilizadas pelo Dataset MURA.



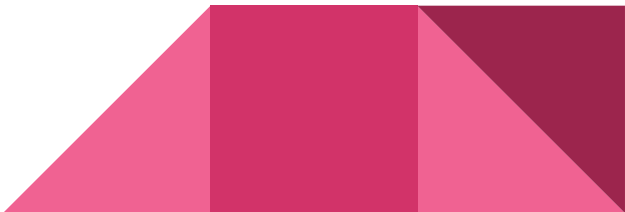
# Introdução


O projeto MURA (***M******u******s******c******u******l******o******s******k******e******t******a******l*** ***R******a******d******i******o******g******r******a******p******h******s***) é um dos maiores conjuntos de dados de imagens radiográficas públicas disponíveis na Internet. Um dos objetivos deste projeto é incentivar uma competição de *Deep Learning*, fornecendo uma grande base de dados de imagens radiográficas de ossos, que estão divididas em 7 áreas de estudos da radiografia: *cotovelo*, *dedos*, *antebraço*, *mão*, *úmero*, *ombro* e *punho*.



# Introdução

As Redes Neurais Convolucionais são muito utilizadas para análise e extração de características em imagens e podem ser utilizadas para identificação de características em imagens de radiografia. O objetivo principal deste trabalho é buscar utilizar técnicas de aprendizado de máquina para realizar uma classificação binária, sendo essa classificação dada como normal e anormal do conjunto de imagens disponibilizadas pelo projeto MURA.






Como a inteligência artificial  
pode auxiliar nas análises de  
quadros clínicos de pacientes?

# Problema

Segundo o informações fornecidas na descrição do projeto MURA, as condições musculoesqueléticas afetam mais de 1,7 bilhão de pessoas em todo o mundo e são a causa mais comum de dor e incapacidade severa a longo prazo. Lesões ou fraturas podem estar relacionadas à falta de cuidado no trânsito, doenças relacionadas ao funcionamento e fragilidade dos ossos e articulações, doenças relacionadas ao trabalho, entre outros.



# Problema

Uma pré análise dos quadros clínicos realizada através da computação, por meio da inteligência artificial, pode filtrar os casos onde realmente existe um problema que deve ser averiguado. Como é apresentado no projeto MURA, a taxa de acerto de um algoritmo pode ser melhor ou tão precisa quanto à análise visual de um médico, bastando apenas ter uma base de dados consistente para o seu devido treino.



# Metodologia

O trabalho foi dividido em 4 etapas principais

- Ajuste do conjunto de imagens
  - Desenvolvimento da Rede Neural Convolutacional
  - Avaliação do modelo
  - Desenvolvimento do ambiente de implantação
-



# Ajuste do conjunto de imagens

- ❑ Dividido em 2 pastas
  - ❑ Conjunto de treinamento (36.808)
  - ❑ Conjunto de testes (3.197)
- ❑ Tamanho total 3,7 GB
- ❑ Rotulados em um arquivo CSV
  - ❑ “0” para normais
  - ❑ “1” para anormais
- ❑ Todas as imagens levavam 5 horas para processar 1 época
- ❑ Limitação do hardware devido a quantidade de imagens

Baseado nestes fatores, decidimos trabalhar com apenas uma parte conjunto de imagens, o subconjunto relacionado ao **pulso**. Criamos então um aplicativo em Python que carregava o arquivo CSV e dividia todas as imagens radiográficas de pulso em duas pastas: casos **positivos** e casos **negativos**.

# Desenvolvimento da Rede Neural Convolucional

- ❑ 13 camadas
- ❑ 20 horas ininterruptas de processamento
- ❑ Equipamento utilizado
  - ❑ SO: Ubuntu 18.10 64 bits
  - ❑ Memória RAM: 8 GB
  - ❑ Processador: Core I5
  - ❑ GPU: NVIDIA 940MX 2 GB
  - ❑ HD: 1 TB

Ao final do treinamento foi gerado um arquivo com extensão H5 (Hierarchical Data Format), que é um tipo de arquivo utilizado para organização e armazenamento de grandes quantidades de dados numéricos. Esse arquivo de modelo pode ser carregado em uma aplicação como, por exemplo, uma API (Application Programming Interface) para classificar imagens radiográficas.

# Avaliação do modelo

Foram utilizadas 3 formas de avaliação do modelo treinado:

- ❑ Plotagem e análise dos gráficos de acurácia e perda que são fornecidas pelos históricos do Framework Keras
- ❑ Geração da Matriz de confusão do classificador
- ❑ Plotagem do gráfico ROC

# Desenvolvimento do ambiente de implantação

## API Python:

- ❑ Entrada de dados no formato JSON
- ❑ Processamento dos dados
- ❑ Retorno em formato JSON com a classificada

## Aplicação FrontEnd:

- ❑ Aplicação HTML e CSS
- ❑ Utilização da biblioteca JQuery de JavaScript
- ❑ Envio dos dados da imagem em Base64 para a API
- ❑ Processamento assíncrono
- ❑ Apresentação do resultado

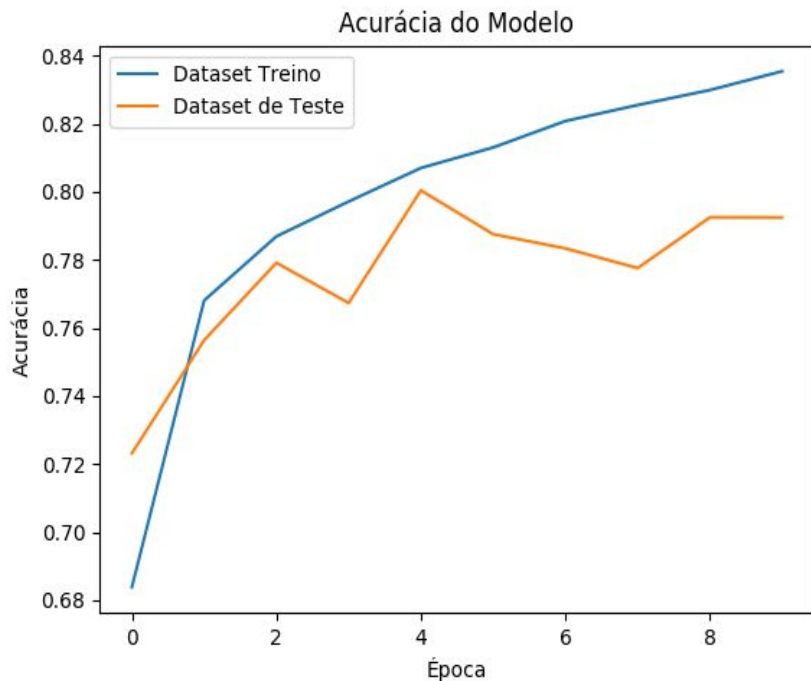
# Resultados

O processo de análise dos resultados do modelo treinado foi dividido em 3 etapas

- Taxa de acurácia e taxa de perda
- Matriz de confusão
- Gráfico ROC

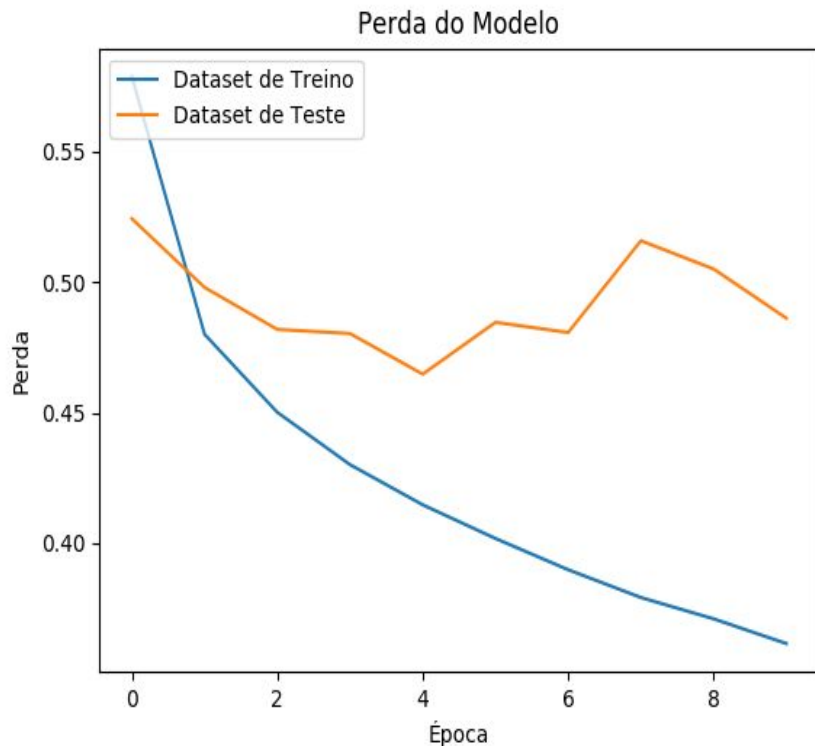
---

# Taxa de acurácia



A diferença entre o conjunto de dados de treinamento e teste é devido a diferença entre a quantidade de exemplos presentes em cada conjunto. O conjunto de treinamento possui um número maior de exemplos do que o conjunto de teste. Em um total de 10 épocas de processamento, a taxa de acurácia do conjunto de treinamento obteve uma taxa próxima a 0.84 e o conjunto de teste obteve uma taxa entre 0.78 e 0.80 de acurácia.

# Taxa de perda



O gráfico de perda do modelo plotado também possui uma diferença entre o conjunto de treinamento e o conjunto de teste. Essa diferença também é relacionada a quantidade de exemplos de imagens em cada classe positiva e negativa. O conjunto de treinamento possui uma quantidade maior de exemplos do que o conjunto de teste.

# Matriz de confusão

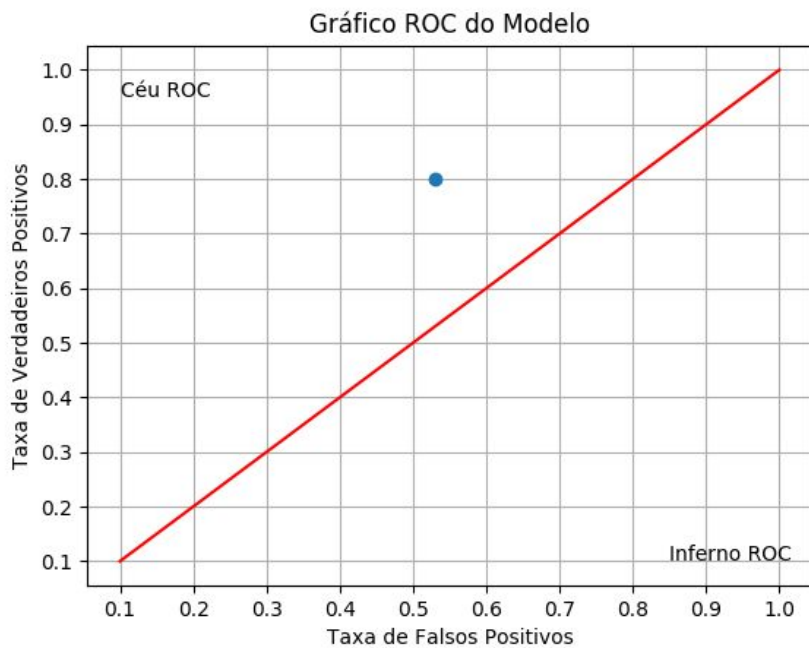
	Predito Positivo	Predito Negativo	Total
Classe Positivo	82	18	100
Classe Negativo	53	47	100
Total	135	65	200

- Taxa de Verdadeiros Positivos (TVP): 82%
- Taxa de Falsos Positivos (TFP): 53%
- Taxa de Acerto: 64,5%
- Taxa de Erro: 35,5%

Foram selecionadas aleatoriamente 100 exemplos de imagens radiográficas positivas e 100 imagens radiográficas negativas do conjunto de teste. Essas imagens foram classificadas pelo modelo treinado. Os resultados da classificação foram coletados e analisados para gerar a Matriz de Confusão. Com base na matriz realizamos os cálculos das taxas relacionadas.



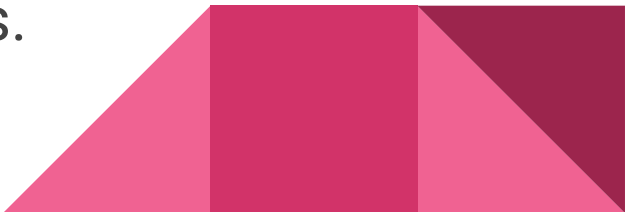
# Gráfico ROC



A última etapa de avaliação dos resultados gerados pelo modelo treinado foi a criação do gráfico ROC. Este gráfico fica mais interessante quando existem mais classificadores para serem comparados, mas com apenas um classificador já é possível ter uma noção sobre o seu desempenho.

# Conclusão

Partindo do objetivo de auxiliar a medicina com a análise automatizada de radiografias, verificou-se que com um algoritmo bem estruturado e uma base de testes bem formada, os resultados de análises automatizadas pode ser tão precisa quanto a de um médico que atua na área há um bom tempo. Por isso, conclui-se que é altamente recomendável, incluir esta abordagem nos centros médicos com o intuito de auxiliar e melhorar a qualidade dos serviços prestados.



# Referências

- ❑ Stanford ML Group. Disponível em: <https://stanfordmlgroup.github.io/competitions/mura/>
- ❑ Python. Disponível em: <https://www.python.org/>
- ❑ Keras: The Python Deep Learning library. Disponível em: <https://keras.io/>
- ❑ TensorFlow. Disponível em: <https://www.tensorflow.org>
- ❑ Flask Framework. Disponível em: <http://flask.pocoo.org/>





Obrigado!