

Classificação de Imagens Radiográficas Utilizando Rede Neural Convolutacional

Michel Motta da Silva

Faculdade de Computação

Universidade Federal de Mato Grosso do Sul

Campo Grande, Brasil

michel.silva@aluno.ufms.br

Thiago Machado

Faculdade de Computação

Universidade Federal de Mato Grosso do Sul

Campo Grande, Brasil

tmachado.ufms@gmail.com

Abstract—RADIOGRAPHIC IMAGES CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK: *This paper presents a general view on the potentialities of Artificial Intelligence (AI) in Images analyzes. It illustrates through an example the use of this analyze in radiograph images with MURA Dataset.*

Index Terms—Artificial Intelligence, Radiograph, MURA

I. INTRODUÇÃO

O projeto MURA (Musculoskeletal Radiographs) é um dos maiores conjuntos de dados de imagens radiográficas públicas disponíveis na Internet. Um dos objetivos do projeto MURA é incentivar uma competição de Deep Learning, fornecendo uma grande base de dados de imagens radiográficas de ossos. Essas imagens foram cedidas pela Escola de Medicina da Universidade de Stanford, Estados Unidos.

O projeto MURA disponibiliza, através do preenchimento de um formulário e aceitação dos termos de uso, um grande conjunto de dados de imagens de radiografias de ossos que consistem em 14.863 estudos de 12.173 pacientes. O conjunto de dados possui um total de 40.561 imagens divididas em 7 áreas de estudos da radiografia. São elas: cotovelo, dedos, antebraço, mão, úmero, ombro e punho.

A Radiografia é uma técnica muito utilizada na medicina para identificar problemas estruturais em ossos do corpo humano e em animais. A Radiografia permite a geração de uma imagem com contrastes acentuados geralmente nas cores preto e branco dos ossos que ficam por baixo da pele. Essas imagens permitem identificar possíveis fraturas em ossos e visualizar componentes que são utilizados para ajustar o posicionamento dos ossos em casos de fraturas. Essas imagens são analisadas por especialistas da área médica para definir os procedimentos que devem ser realizados dependendo da situação identificada na imagem radiográfica.

O projeto MURA visa a utilização da Inteligência Artificial para propiciar um modelo capaz de analisar imagens radiográficas de ossos em lugares onde seja difícil o acesso a profissionais capacitados para analisar corretamente as imagens radiográficas. Mesmo em casos de análise humana por profissionais capacitados, os diagnósticos estão sujeitos a falhas humanas, dada a técnica de análise visual utilizada pelos profissionais da Radiologia.

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 como as Redes Neurais Convolucionais para realizar uma classificação binária sendo essa classificação dada como normal e anormal do conjunto de imagens disponibilizadas pelo projeto MURA.

II. PROBLEMA

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

Segundo as 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, com 30 milhões de visitas ao departamento de emergência por ano.

Com base em estudos recentes, o número de pacientes com fraturas ou lesões vem crescendo de forma assustadora, e a razão para tal acontecimento pode estar relacionada a diversos fatores como: à falta de cuidado no trânsito, doenças relacionadas ao funcionamento e fragilidade dos ossos e articulações, doenças relacionadas ao trabalho, entre outros. Com base nisso, a demanda médica para esses casos também cresce de maneira proporcional, o que leva a uma certa perda de tempo para realizar manualmente as análises radiográficas. Em alguns casos não se faz necessário um acompanhamento clínico ou intervenções cirúrgicas.

Sendo assim, uma pré análise dos quadros clínicos realizadas 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. Após isso, a análise poderá ser realizada através da própria máquina que realiza a radiografia, e o paciente já ficaria sabendo ali mesmo se seu quadro é de risco ou não, isso facilitará e melhorará o atendimento em diversos centros médicos e hospitais.

III. METODOLOGIA

Este trabalho foi dividido em 4 etapas principais. A primeira etapa foi o ajuste do conjunto de imagens. A segunda etapa

foi o desenvolvimento e treinamento da Rede Neural Convolutacional. A terceira etapa foi a avaliação do modelo. A quarta e última etapa foi o desenvolvimento do ambiente de implantação, que consiste em uma API utilizando Python e uma aplicação FrontEnd para consumir dados da API desenvolvida. O modelo foi desenvolvido utilizando a linguagem de programação Python 3 e os Frameworks de aprendizado de máquina TensorFlow e Keras.

A. Ajuste do Conjunto de Imagens

A primeira etapa do trabalho foi o ajuste dos conjuntos de imagens disponibilizado. O conjunto de imagens estava dividido em duas pastas. Na primeira pasta estavam as imagens relativas ao conjunto de treinamento. Esse conjunto de treinamento estava dividido em pastas representando as principais áreas de estudo da radiologia, que são cotovelo, dedos, antebraço, mão, úmero, ombro e punho. Na segunda pasta estavam as imagens relacionadas ao conjunto de teste, que seguia a mesma divisão do conjunto de treinamento. A principal diferença entre o conjunto de treinamento e o conjunto de teste era a quantidade de imagens em cada conjunto.

O conjunto de treinamento possuía 36.808 imagens e o conjunto de teste possuía 3.197 com um tamanho total de aproximadamente 3.5 GB. O conjunto de imagens estavam devidamente rotulados através de um arquivo de extensão CSV (Comma-separated values). Cada linha do arquivo CSV continha o caminho para a pasta das imagens, assim como a classificação das imagens. As imagens possuíam a classificação "0" para as imagens que foram consideradas normais pelos especialistas de Radiologia e a classificação "1" para as imagens que foram consideradas anormais.

O conjunto de imagens possuía um grande número de imagens e tivemos algumas dificuldades durante o processo de treinamento da Rede Neural Convolutacional. O processo de aprendizagem com todas as imagens estava demorando aproximadamente 5 horas para processar cada época. Além disso, tivemos alguns problemas relacionados a capacidade de Hardware. Muitas das tentativas de desenvolvimento do modelo resultavam de mensagens de erros por pouca memória disponível.

Desta forma, decidimos trabalhar apenas com uma parte do conjunto de dados. Escolhemos apenas um subconjunto de imagens da área de estudo da Radiografia. O subconjunto escolhido foi o pulso. Nós criamos um programa em Python que carregava o arquivo CSV e dividia todas as imagens radiográficas de pulso em duas pastas. A pasta dos casos positivos e a pasta dos casos negativos. Esse processo de divisão foi executada tanto para o conjunto de dados de treinamento quanto para o conjunto de dados de teste.

B. Desenvolvimento e Treinamento da Rede Neural Convolutacional

O projeto MURA descreve que a Rede Neural Convolutacional que foi desenvolvido por eles possui 169 camadas. Um dos grandes problemas enfrentados durante o desenvolvimento deste trabalho foi a limitação de Hardware que foi utilizado

para o treinamento da Rede Neural Convolutacional desenvolvida. O treinamento de uma Rede Neural Convolutacional requer um equipamento de Hardware robusto capaz de lidar com os constantes cálculos e operações realizadas no processo de treinamento da rede. O modelo gerado neste trabalho foi treinado utilizando uma Rede Neural Convolutacional composta por 13 camadas.

Dada a limitação de Hardware e a necessidade de processamento do problema, o treinamento da Rede Neural Convolutacional durou aproximadamente 20 horas contínuas. O equipamento utilizado para o treinamento possuía as seguintes características:

- Sistema operacional Linux Ubuntu 18.10 - 64 bits
- Memória RAM de 8 GB
- Processador Intel Core I5
- Placa de vídeo NVIDIA GeForce 940MX com 2 GB de memória dedicada
- Disco Rígido com 1 TB de memória

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.

C. Avaliação do Modelo

Foram utilizadas 3 formas de avaliação do modelo treinado. A primeira forma de avaliação foi a plotagem e análise dos gráficos de acurácia e perda que são fornecidas pelos históricos do Framework Keras. Essas imagens podem ser visualizadas no tópico Resultados deste relatório. A segunda forma de avaliação foi a geração da Matriz de confusão do classificador. A terceira e última forma de avaliação do classificador foi a plotagem do gráfico ROC.

D. Desenvolvimento do Ambiente de Implantação

A última etapa deste trabalho foi a criação de um ambiente para implantação e utilização do modelo treinado, sempre tendo em vista uma utilidade para este modelo. Após analisar algumas possibilidades de implantação do modelo, nós desenvolvemos uma API utilizando a linguagem de programação Python 3 e o Framework Flask.

Essa API foi desenvolvida com o objetivo de receber dados de imagens em formato JSON (JavaScript Object Notation), que é uma forma de estrutura de dados, analisar essas imagens através do modelo treinado e devolver uma resposta no mesmo formato com a imagem e a classificação dessa imagem como "normal" ou "anormal". Também desenvolvemos uma aplicação visual (FrontEnd) utilizando HTML, CSS, Javascript e o plugin Javascript JQuery. Essa aplicação FrontEnd roda no navegador e os usuários podem selecionar uma quantidade de imagens para serem analisadas pela API. A aplicação FrontEnd é responsável por converter as imagens selecionadas pelo usuário em BASE64, que é um método para codificação de dados para transferência na Internet.

Após realizada a conversão das imagens os dados são estruturados no formato JSON e submetidos a API de forma assíncrona. Quando a API retorna a resposta, a aplicação FrontEnd converte os dados em formato JSON em HTML e mostra o conteúdo na tela do usuário.

IV. RESULTADOS

O processo de análise dos resultados do modelo treinado foi dividido em 3 etapas. A primeira etapa foi a plotagem e análise dos gráficos de acurácia e perda que são salvos pelo histórico de treinamento do Framework Keras. Na segunda etapa foi desenvolvido um programa em Python que selecionava aleatoriamente 100 imagens radiográficas positivas e 100 imagens radiográficas negativas do conjunto de dados de teste. Essas imagens foram submetidas e classificadas pelo modelo treinado.

Com a classificação das imagens e comparação com a classe real de cada imagem radiográfica, foi possível gerar a Matriz de Confusão e calcular as taxas de verdadeiros positivos e de falsos positivos, assim como a taxa de acerto e a taxa de erro. Na terceira e última etapa foi realizado a plotagem do gráfico ROC do classificador.

A. Taxa de Acurácia e Taxa de Perda

A Rede Neural Convolutiva de 13 camadas que foi desenvolvida neste projeto, apesar de não apresentar resultados excelentes, por causa da limitação de Hardware, obteve alguns resultados interessantes.

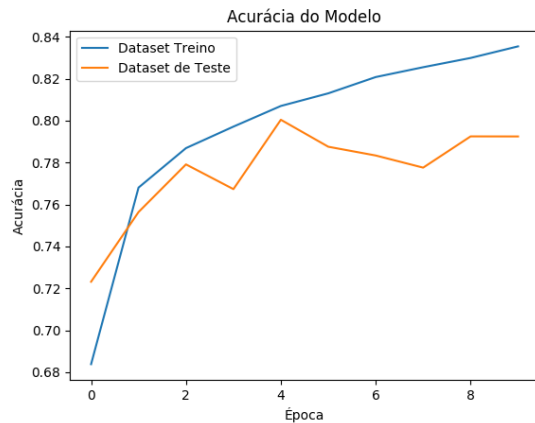


Fig. 1. Taxa de acurácia do modelo.

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.

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.

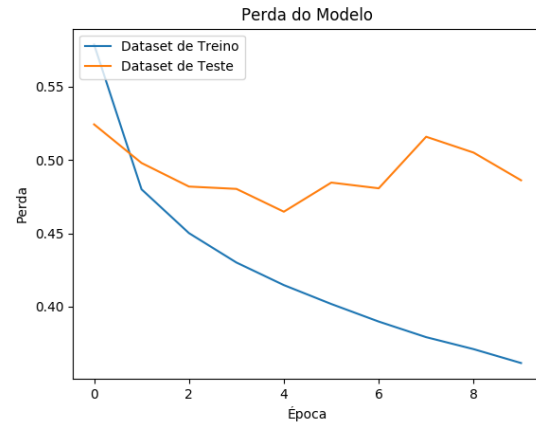


Fig. 2. Taxa de perda do modelo.

A taxa de perda do modelo para o conjunto de treino obteve um resultado abaixo de 0.40 e a taxa de perda do modelo para o conjunto de teste obteve um resultado que variou entre 0.45 e 0.55. Essas métricas de taxa de acurácia e perda são métricas que são fornecidas pelo Framework Keras, sendo calculadas e atualizadas em cada época de treinamento.

B. Matriz de Confusão

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.

TABELA I
MATRIZ DE CONFUSÃO

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

Após a construção da Matriz de Confusão, foram realizados os cálculos necessários para encontrar a taxa de verdadeiros positivos, a taxa de falsos positivos, a taxa de acerto e a taxa de erro do modelo.

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

C. 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.

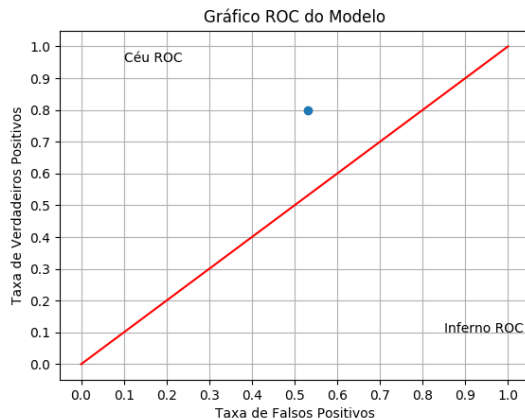


Fig. 3. Gráfico ROC do classificador.

A taxa de verdadeiros positivos (TVP) obteve o valor de 0.82 e taxa de falsos positivos (TFP) obteve o valor de 0.53. Esses valores quando plotados no gráfico ROC ficam acima da linha dos classificadores aleatórios e demonstra um resultado interessante, dado todas as problemáticas e limitação durante o desenvolvimento do trabalho.

V. CONCLUSÃO

Produzir este trabalho foi de suma importância para ampliação do conhecimento dos autores sobre o tema que está presente na realidade profissional de sua área de estudo. Investigar como a inteligência artificial pode melhorar os atendimentos médicos, trazendo comodidade e eficiência nas análises dos quadros clínicos, é muito relevante para a compreensão dos pontos que a computação pode auxiliar no dia a dia de muitas áreas.

A análise de uma radiografia muitas vezes pode mostrar ao médico ou especialista que analisa, uma diversidade de pontos que necessitem de atenção. Uma análise realizada de forma equivocada pode gerar diversos transtornos, tanto para o paciente, quanto para o corpo médico que o acompanha. Trazer comodidade e a facilidade de uma pré seleção dos casos que realmente necessitam de uma intervenção médica, é essencial para o bem estar dos pacientes e melhor aproveitamento do tempo em que os especialistas ficam em um determinado hospital ou centro médico.

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.

VI. BIBLIOGRAFIA

- Stanford ML Group. Disponível em: <https://stanfordmlgroup.github.io/competitions/mura/>, acessado em 15/Nov/2018.
- Python. Disponível em: <https://www.python.org/>, acessado em 19/Nov/2018.
- Keras: The Python Deep Learning library. Disponível em: <https://keras.io/>, acessado em 20/Nov/2018.
- TensorFlow. Disponível em: <https://www.tensorflow.org>, acessado em 21/Nov/2018.
- Flask Framework. Disponível em: <http://flask.pocoo.org/>, acessado em 27/Nov/2018.