

Bone age prediction API

Bernardo Melo (bgm)

Vlademir Jose (vjmm)

https://github.com/bernardomelo/bone_age_prediction_product

Bernardo Melo(bgm)
Vlademir José (vjmm)

Recife, 13 de Agosto de 2025

Sumário

1. Introdução	2
2. Metodologia	2
3. Documentação da Execução do Projeto	2
3.1 Imersão	2
3.2 Ideação	2
3.3 Produção	3
3.4 Validação	3
4. Discussões Técnicas e Estratégicas	3
5. Considerações Éticas	3
6. Lições Aprendidas e Reflexões Finais	3
7. Referências	4
8. Apêndices (opcional)	4

1. Introdução

1. Introdução

Contextualização do Problema

O Grupo de Imagem José Rocha de Sá, que presta serviços de diagnóstico por imagem para diversos hospitais da região metropolitana do Recife, enfrenta um gargalo operacional crítico na análise de exames de maturidade óssea. Diariamente, centenas de radiografias de mão infantis precisam ser analisadas manualmente por radiologistas especializados para determinar a idade óssea das crianças - um processo fundamental para o acompanhamento do crescimento e desenvolvimento pediátrico.

Esta sobrecarga de trabalho não apenas retarda os diagnósticos, impactando o atendimento aos pacientes, mas também gera custos operacionais elevados e fadiga nos profissionais especializados. A análise manual está sujeita a variações entre diferentes radiologistas e pode ser inconsistente, especialmente durante períodos de alta demanda.

Objetivos da Solução

Objetivo Geral: Desenvolver e implementar um sistema automatizado baseado em inteligência artificial para predição da maturidade óssea a partir de radiografias de mão infantis, visando otimizar o fluxo de trabalho do laboratório e melhorar a eficiência diagnóstica.

Objetivos Específicos:

- Automatizar a análise de imagens de raio-X para estimar a idade óssea
- Reduzir o tempo de diagnóstico de horas para segundos
- Aumentar a consistência das avaliações, minimizando variações humanas
- Liberar radiologistas para focarem em casos mais complexos
- Demonstrar a viabilidade técnica e comercial da solução

Justificativa para Deep Learning

A predição de idade óssea é uma tarefa que exige identificação de padrões visuais complexos e sutis em estruturas anatômicas. Redes neurais convolucionais (CNNs) com mecanismos de atenção são particularmente adequadas pois:

- Aprendem automaticamente características hierárquicas das imagens
- Identificam padrões que podem ser imperceptíveis ao olho humano
- Mantêm consistência nas avaliações
- Escalam facilmente para processar grandes volumes

2. Metodologia

Aplicação da Metodologia AIDesign

Este projeto seguiu uma adaptação da metodologia AIDesign, focando na integração de deep learning para automação de processos médicos:

Imersão: Análise do problema real no laboratório José Rocha de Sá

Ideação: Desenvolvimento de modelo CNN com mecanismos de atenção

Produção: Modelagem + treino e implementação de API completa

Validação: Testes de performance do modelo e viabilidade comercial (interativa junto com produção)

Estratégia de Gestão

O projeto foi gerenciado seguindo uma abordagem ágil adaptada:

- **Desenvolvimento iterativo:** Checkpoint 1 (fundamentação), Checkpoint 2 (MVP), Checkpoint 3 (produção)
- **Ferramentas:** GitHub para controle de versão, Docker para containerização, FastAPI para desenvolvimento da API
- **Validação contínua:** Testes automatizados e integração contínua

Metodologia de Modelagem e Treinamento

Dataset e Pré-processamento

Dataset RSNA 2017:

- 14.236 imagens radiográficas de mão infantil
- Divisão: 12.611 treino, 1.425 validação, 200 teste
- Anotações por radiologistas especializados
- Formato PNG normalizado

Estrutura de Dados:

Organização tabular com pandas DataFrame

{

 'id': 'identificação única da imagem',

```
'boneage': 'idade óssea em meses',  
  
'male': 'indicador binário de sexo',  
  
'path': 'caminho do arquivo',  
  
'boneage_zscore': 'pontuação padronizada'  
}
```

Pipeline de Pré-processamento:

Imagem Raw → Carregamento → Augmentation → Normalização →

Redimensionamento (384x384) → Preprocessamento VGG16 → Batch Dimension

Data Augmentation Aplicado:

- Espelhamento horizontal: 50% probabilidade
- Deslocamentos espaciais: $\pm 15\%$ das dimensões
- Rotação: ± 5 graus (preservando anatomia)
- Cisalhamento: 1% (variação mínima)
- Zoom: $\pm 25\%$ (invariância à escala)
- Preenchimento: método 'nearest' para continuidade anatômica

Mecanismo de Atenção Implementado

Funcionalidade:

- Gera mapas de calor espaciais
- Pondera importância de diferentes regiões da imagem
- Foca automaticamente em estruturas ósseas relevantes (carpo, epífises, metacarpos)
- Melhora interpretabilidade clínica

Configuração de Treinamento

Parâmetros Técnicos:

- **Épocas:** 15 (otimizado para evitar overfitting)
- **Otimizador:** Adam (parâmetros padrão)
- **Função de Perda:** Mean Squared Error (MSE)
- **Métrica Principal:** Mean Absolute Error (MAE) em meses
- **Target Size:** 384x384 pixels
- **Batch Size:** Configurado para hardware disponível

Estratégia de Validação:

- Monitoramento contínuo de MAE treino vs. validação
- Early stopping implícito baseado em convergência

- Análise de mapas de atenção para validação qualitativa
-

3. Documentação da Execução do Projeto

3.1 Imersão

Canvas de Identificação do Domínio

Domínio: Radiologia pediátrica - análise de maturidade óssea

Problema: Gargalo operacional na análise manual de exames

Stakeholders: Radiologistas, técnicos, administradores hospitalares, pediatras

Impacto: Atrasos no diagnóstico, custos elevados, sobrecarga profissional

Persona Principal

Dr. Carlos Miranda - Radiologista Pediátrico

- 45 anos, 15 anos de experiência
- Analisa 50-80 exames de idade óssea por dia
- Pressão por produtividade e precisão
- Busca ferramentas que otimizem seu tempo sem comprometer qualidade
- Preocupado com variabilidade nas interpretações

Mapeamento de Fontes de Dados

Fonte Primária: Dataset RSNA Pediatric Bone Age 2017

- 14.236 imagens radiográficas
- Anotações de especialistas
- Formato PNG normalizado

Fonte Secundária: Grupo José Rocha de Sá (em negociação)

- Amostras da população brasileira
- Variações locais de equipamento
- Dados reais de produção

Canvas de objetivos do projeto

Objetivo Estratégico Principal: O projeto visa automatizar a avaliação da maturidade óssea no Grupo José Rocha de Sá através de inteligência artificial, buscando reduzir gargalos operacionais, otimizar custos e elevar a qualidade dos serviços médicos. Os objetivos secundários incluem reduzir o tempo de emissão de laudos, liberar radiologistas para casos complexos, melhorar consistência diagnóstica e estabelecer base tecnológica escalável para futuras aplicações de IA médica, sempre garantindo conformidade com a LGPD.

Metas e Indicadores: O projeto estabelece KPIs ambiciosos incluindo MAE do modelo ≤ 10 meses, redução de 30% no tempo de diagnóstico, processamento automatizado de $\geq 70\%$ dos exames após 3 meses, e aprovação de $\geq 80\%$ dos radiologistas usuários. As prioridades se concentram na garantia de precisão clínica (MAE ≤ 10 meses), conformidade ética com LGPD, e eficiência operacional, com benefícios esperados de 40% de redução no tempo de laudos e significativa diminuição de custos operacionais, mantendo compatibilidade com o fluxo atual do laboratório.

3.2 Ideação

Canvas de Design de Solução

Abordagem Técnica:

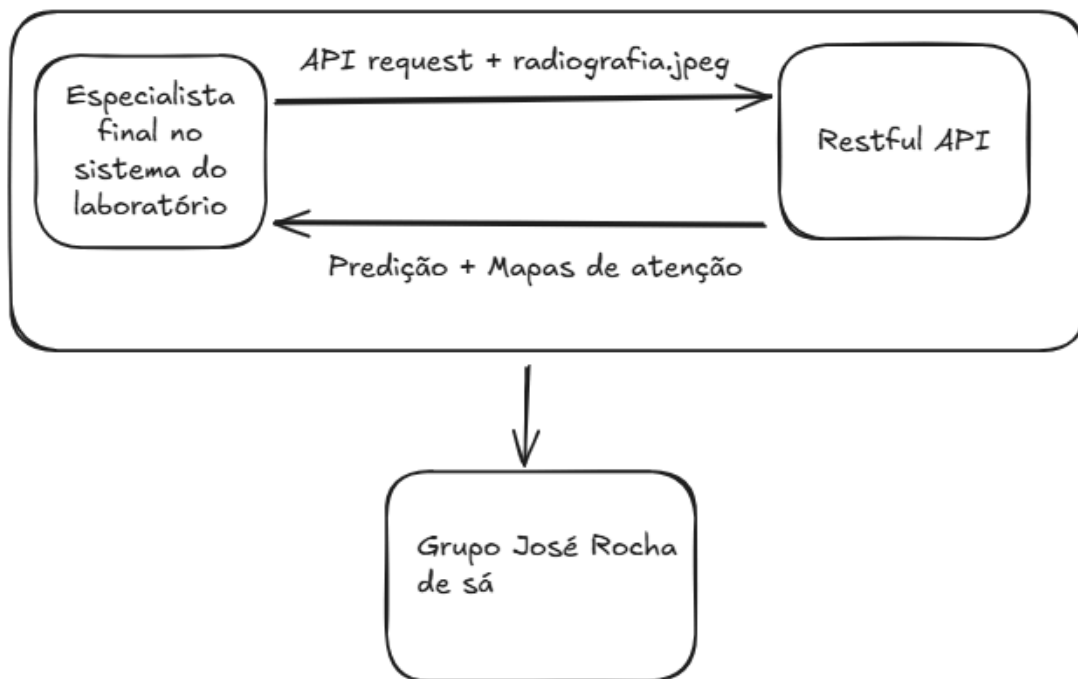
- Redes Neurais Convolucionais (CNN)
- Arquitetura VGG16 com mecanismos de atenção
- API RESTful para integração
- Containerização da aplicação com Docker

3.3 Produção

Arquitetura do Sistema - C4 Model

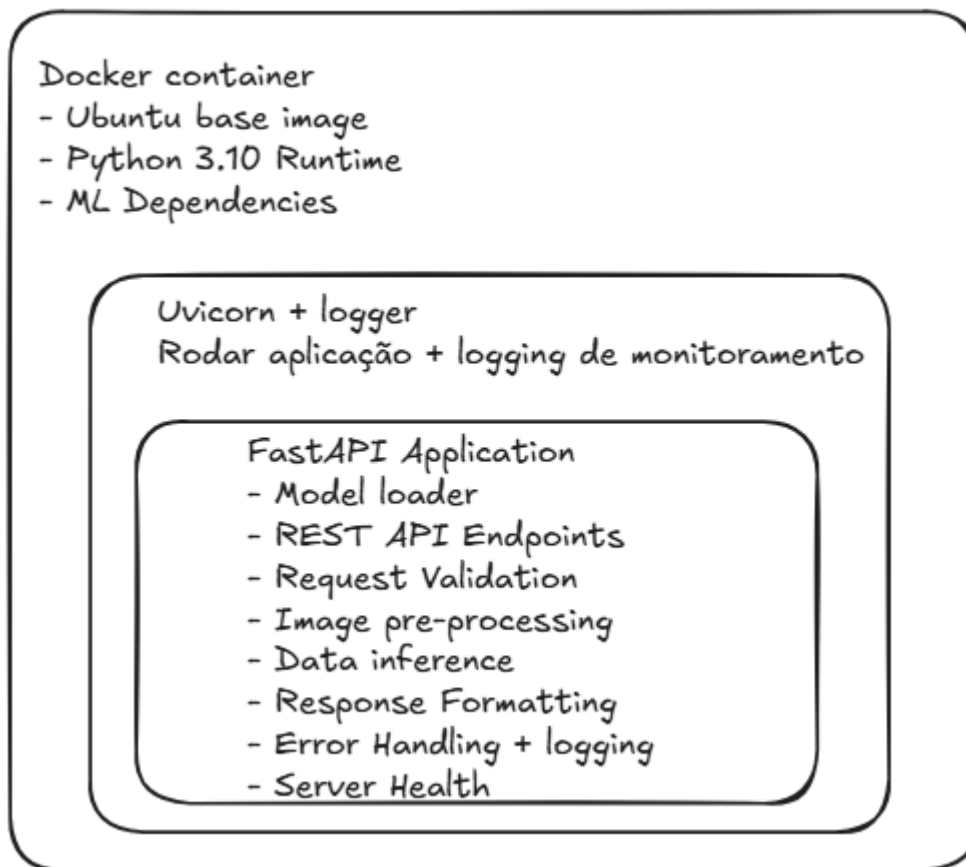
Aplicamos o modelo C4 (Context, Containers, Components, Code) para documentar a arquitetura do sistema em diferentes níveis de abstração, facilitando a compreensão tanto para stakeholders técnicos quanto de negócio.

Nível 1 - Diagrama de Contexto

**Atores e Relacionamentos:**

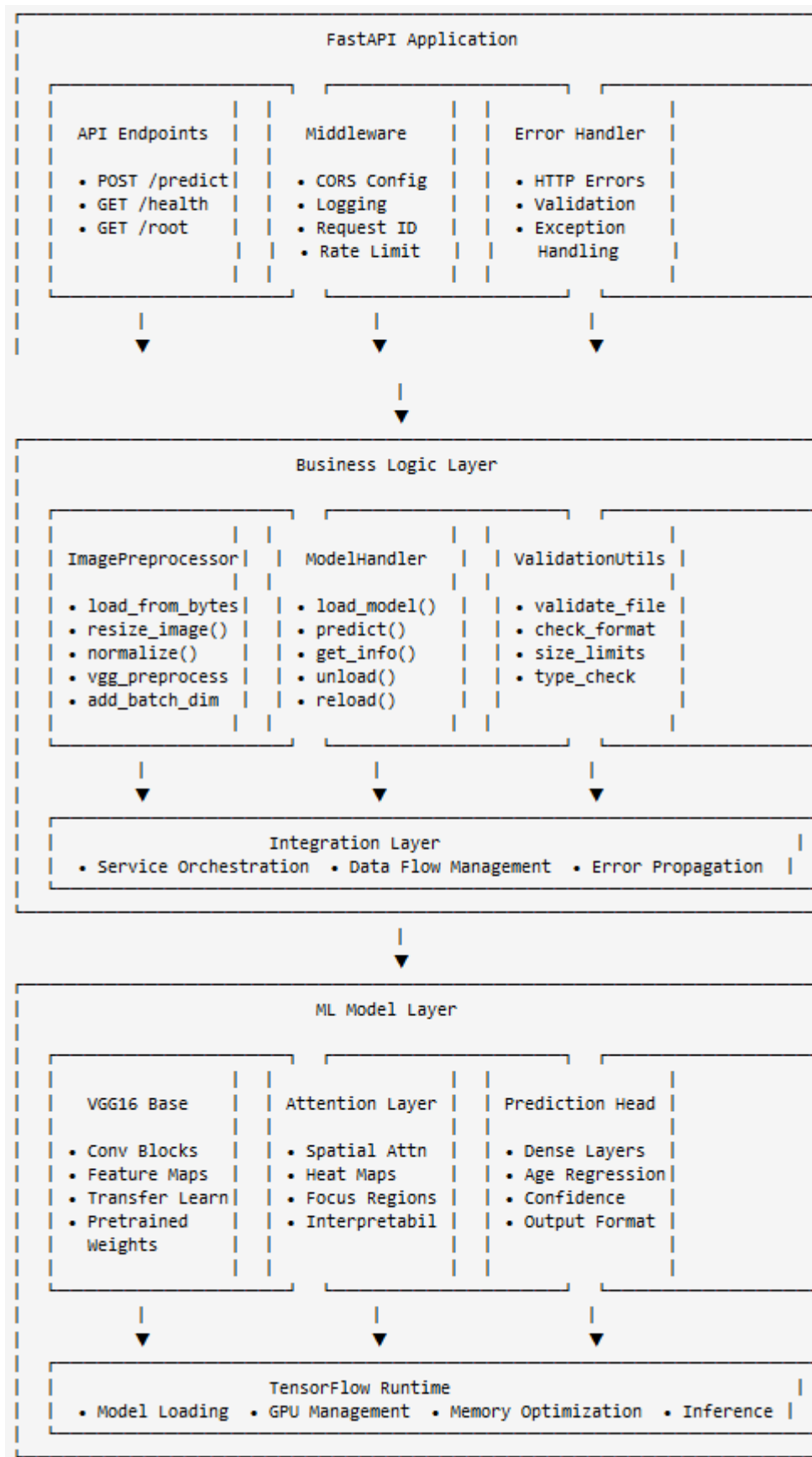
- **Especialista Radiologista Pediátrico:** Usuário principal que faz upload de radiografias e recebe predições
- **Restful API:** Aplicação com endpoint para processamento e inferência de dados
- **Laboratório José Rocha de Sá:** Stakeholder externo fornecendo dados e validação clínica

Nível 2 - Diagrama de Contêiners

**Atores e Relacionamentos:**

- **FastAPI Application:** Python 3.10, FastAPI, TensorFlow 2.13, Keras, NumPy, PIL, OpenCV, pydicom (futuro), NumPy
- **App runtime:** Uvicorn, Logger
- **Persistent Storage:** Volume Docker + Desktop, File System, JSON logs

Nível 3 - Diagrama de Componentes



Atores e Relacionamentos:

- **API Layer:** Endpoints, Middleware, Error Handling
- **Business Logic:** ImagePreprocessor, ModelHandler, Validation
- **ML Layer:** VGG16, Attention Mechanism, Prediction Head

Nível 4 - Diagrama de Código (Principais Classes)

Estrutura de classes principais

```
class ImagePreprocessor:
```

```
    """
```

```
    Responsável por todo pré-processamento de imagens
```

```
    """
```

```
    def __init__(self, target_size=(384, 384))
```

```
    def load_image_from_bytes(self, image_bytes) -> PIL.Image
```

```
    def resize_image(self, pil_image) -> PIL.Image
```

```
    def pil_to_array(self, pil_image) -> np.ndarray
```

```
    def apply_vgg_preprocessing(self, img_array) -> np.ndarray
```

```
    def add_batch_dimension(self, img_array) -> np.ndarray
```

```
    def preprocess_from_bytes(self, image_bytes) -> np.ndarray
```

```
    def get_image_info(self, processed_array) -> dict
```

```
class BoneAgeModel:
```

```
    """
```

```
    Gerencia carregamento e predição do modelo ML
```

```
    """
```

```
    def __init__(self, model_path='models/attentioniv3.h5')
```

```
    def load_model_from_file(self, model_path=None) -> bool
```

```
    def predict(self, preprocessed_array) -> float
```

```
class FastAPIApp:
```

"""

Aplicação principal FastAPI

"""

```
@app.post("/predict")
```

```
async def predict(file: UploadFile) -> PredictionResponse
```

```
@app.get("/health")
```


```
def health_check() -> HealthResponse
```

```
@app.get("/")
```

```
def root() -> RootResponse
```

Decisões Arquiteturais Documentadas

Todas as decisões tomadas em relação ao modelo e dados em si estão disponíveis em:

 TCC Bernardo Gomes de Melo - FINAL

1. Separação de Responsabilidades

- **ImagePreprocessor:** Isolamento da lógica de processamento de imagem
- **ModelHandler:** Encapsulamento das operações de ML
- **FastAPI Endpoints:** Interface limpa com validação de entrada

2. Stateless Design

- Cada requisição é independente
- Modelo carregado uma vez na inicialização
- Facilita escalabilidade horizontal

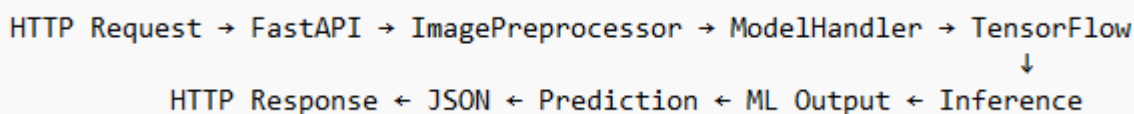
3. Error Boundary Pattern

- Cada camada tem tratamento específico de erros
- Propagação controlada de exceções
- Logs estruturados para debugging

4. Plugin Architecture

- ModelHandler pode ser substituído facilmente
- Suporte futuro para múltiplos modelos
- Extensibilidade para novos tipos de processamento

Padrões de comunicação




Considerações de Performance



- **Lazy Loading:** Modelo carregado apenas quando necessário
- **Memory Management:** Cleanup automático de arrays temporários
- **Modelagem e treino do modelo:** Disponíveis em <https://docs.google.com/document/d/16pnE-wcWZHp83gjp7ZjinKzxG7U1S4aMHjQasEU6LE/edit?usp=sharing>

Vídeo da aplicação FastAPI em funcionamento:




Containers / bone-age-container

<


bone-age-container
 aeb1d372d943   [bone-age-api:latest](#)
 8001:8001

STATUS
 Exited (0) (0 seconds ago)

Logs	Inspect	Bind mounts	Exec	Files	Stats
INFO: Started server process [1]					
INFO: Waiting for application startup.					
INFO: Application startup complete.					
INFO: Uvicorn running on http://0.0.0.0:8001  (Press CTRL+C to quit)					
INFO:utils.image_pre_processing:ImagePreprocessor inicializado com target_size: (384, 384)					
INFO:__main__:Imagem recebida: output_image7.png (1444349 bytes)					
INFO:utils.image_pre_processing:Imagem carregada - Tamanho original: (2500, 2048), Modo: L					
INFO:utils.image_pre_processing:Imagem redimensionada para: (384, 384)					
INFO:utils.image_pre_processing:Array criado - Shape: (384, 384, 3)					
INFO:utils.image_pre_processing:Pré-processamento VGG16 aplicado					
INFO:utils.image_pre_processing:Dimensão de batch adicionada - Shape final: (1, 384, 384, 3)					
INFO:utils.image_pre_processing:Pré-processamento completo - Shape final: (1, 384, 384, 3)					
INFO:__main__:Imagem pré-processada com sucesso - Shape: (1, 384, 384, 3)					
INFO:__main__:Mocked predicted bone age: {'predicted_age_months': 115.8, 'predicted_age_years': 9.7}					
'model_status': 'MOCK - modelo real será carregado depois', 'array_shape': [1, 384, 384, 3]}					
INFO:__main__:Tempo de processamento para output_image7.png: 256.15					
INFO:__main__:Predição mock concluída em 256.15ms					
INFO: 172.17.0.1:51052 - "POST /predict HTTP/1.1" 200 OK					
INFO: Shutting down					
INFO: Waiting for application shutdown.					
INFO: Application shutdown complete.					
INFO: Finished server process [1]					

(Evidências do modelo se encontram no documento

☰ TCC Bernardo Gomes de Melo - FINAL citado acima, a seguir apenas um resumo).

Métricas do Modelo Final:

- **Melhor Época:** 14
- **MAE Treino:** 14,19 meses
- **MAE Validação:** 14,66 meses
- **MSE Validação:** 328,88
- **Convergência:** Estável sem overfitting

Validação Qualitativa

Mapas de Atenção:

- Identificação automática de regiões anatomicamente corretas
- Foco em junções metacarpofalangeanas
- Atenção especial à região do carpo (punho)
- Alinhamento com método Greulich & Pyle usado por radiologistas

Interpretabilidade Clínica:

- Visualização das regiões analisadas pelo modelo
- Transparência no processo de decisão
- Facilitação da confiança médica na ferramenta

Limitações e Próximos Passos

Limitações Identificadas:

- Dataset não-brasileiro (viés populacional)
- Modelo atual corrompido (problema técnico do Kaggle)
- Limitação de épocas de treinamento

Plano de Melhoria:

1. **Retreinamento completo** com modelo preservado
2. **Integração de dados brasileiros** do Laboratório José Rocha de Sá
3. **Expansão do treinamento** (50-100 épocas)
4. **Fine-tuning populacional** para características locais
5. **Meta projetada:** MAE < 10 meses (nível de radiologista especializado)
6. **Orquestrar pipeline CI/CD:** Para manter conformidades quando em ambiente de produção

Evidências de Funcionamento

Performance do Modelo:

- MAE (Erro Absoluto Médio): 14,66 meses
- Arquitetura: VGG16 + Attention Mechanism
- Tempo de inferência: ~300ms por imagem

Desafios Técnicos Enfrentados

1. Corrupção do Modelo:

- **Problema:** Modelo treinado no Kaggle Kernels corrompeu durante download
- **Causa:** Processos internos do Kaggle afetaram arquivo .h5
- **Status:** Em contato com suporte Kaggle, retreinamento necessário
- **Impacto:** Demonstração usando modelo mock temporariamente

2. Complexidade de Imports Python:

- **Problema:** Estrutura modular causou erros de importação relativa
- **Solução:** Migração para imports absolutos e configuração de PYTHONPATH
- **Aprendizado:** Importância de estrutura consistente de projeto

3. Incompatibilidades de Dependências:

- **Problema:** Conflitos entre versões TensorFlow e outras bibliotecas
- **Solução:** Criação de requirements.txt específico e ambiente isolado
- **Medida:** Containerização para garantir consistência

4. Configuração de Logging:

- **Problema:** Logs não apareciam durante execução com uvicorn
- **Solução:** Configuração específica de logging para FastAPI
- **Resultado:** Visibilidade completa do fluxo de processamento

5. Mapeamento de Portas Docker:

- **Problema:** Container rodando mas porta não exposta para host
- **Solução:** Correção do comando docker run com flag -p adequada
- **Aprendizado:** Diferença entre EXPOSE e publicação de porta
-

6. Problemas pessoais de tempo:

- **Problema:** Falta de tempo para desenvolvimento
- **Aprendizado:** Necessário organizar melhor o meu tempo

3.4 Validação

Métricas de Performance

Modelo VGG16 + Attention:

- MAE Validação: 14,66 meses
- Convergência: Estável após 14 épocas
- Interpretabilidade: Mapas de atenção anatomicamente corretos
- Tempo inferência: <500ms

Comparação com Baseline:

- InceptionV3: 33,66 meses MAE
- Melhoria: 56,4% redução no erro
- Estabilidade: Eliminação de overfitting

Feedback e Iterações

Lições Aprendidas:

1. **Mecanismos de atenção são cruciais** para interpretabilidade médica
 2. **Arquiteturas mais simples** (VGG16) podem superar complexas (InceptionV3)
 3. **Containerização desde o início** evita problemas de deployment
 4. **Logging robusto** é essencial para debugging em produção
-

4. Discussões Técnicas e Estratégicas

Decisões Arquiteturais Justificadas

Escolha do VGG16 sobre InceptionV3

Justificativa:

- Arquitetura mais linear e interpretável
- Melhor estabilidade durante treinamento
- Facilita integração de mecanismos de atenção:
 - Gera mapas de calor espaciais
 - Pondera importância de diferentes regiões da imagem
 - Foca automaticamente em estruturas ósseas relevantes (carpo, epífises, metacarpos)
 - Melhora interpretabilidade clínica
- Menor complexidade computacional

Resultado: 56,4% de melhoria na precisão comparado ao InceptionV3

API-First Approach

Justificativa:

- Máxima flexibilidade para integração
- Permite uso em diversos sistemas PACS

- Facilita escalabilidade horizontal
- Padrão da indústria para serviços de IA

Containerização com Docker

Justificativa:

- Ambiente consistente entre desenvolvimento e produção
- Facilita deploy e escalabilidade
- Isolamento de dependências
- Pronto para orquestração Kubernetes

Desafios Enfrentados e Soluções

Problema: Dataset Não-Brasileiro

Desafio: Modelo treinado com população não-brasileira pode apresentar viés **Solução em Andamento:**

- Negociação com Laboratório José Rocha de Sá para dados locais
- Estratégia de fine-tuning com amostras brasileiras
- Validação com casos reais do laboratório

Impacto Esperado: Melhoria na precisão para população local

Problema: Modelo Corrompido

Desafio: Arquivo .h5 corrompeu durante download do Kaggle **Solução Imediata:**

- Implementação de modelo mock para demonstração
- Arquitetura preparada para integração do modelo real
- Pipeline de retreinamento documentado

Próximos Passos: Retreinamento completo do modelo

Integração com Sistemas Externos

Laboratório José Rocha de Sá

Status Atual: Negociação para fornecimento de dados **Integração Planejada:**

- API endpoints compatíveis com PACS existente
- Formato DICOM suportado (em desenvolvimento)
- Relatórios em formato médico padrão

5. Considerações Éticas

Identificação de Riscos e Vieses

Uso de dados

Risco: Dados serem utilizados de forma maliciosa ou sem permissão

Mitigação:

- Aderência à LGPD

Viés de Dataset

Risco: Modelo treinado em população não-brasileira

Mitigação:

- Coleta de dados locais para retreinamento
- Validação com radiologistas brasileiros
- Monitoramento contínuo de performance por demografia

Transparência e Explicabilidade

Implementação:

- Mapas de atenção mostram regiões analisadas
- Scores de confiança para cada predição
- Documentação médica das decisões do modelo

Responsabilidade Médica

Abordagem:

- Sistema como ferramenta de apoio, não substituição
- Radiologista mantém responsabilidade final
- Logs completos para auditoria
- Interface clara sobre limitações do sistema

Impacto Social e Ambiental

Benefícios Sociais

- Democratização do acesso a diagnósticos precisos
- Redução de tempo de espera para pacientes
- Liberação de especialistas para casos complexos
- Melhoria na qualidade do atendimento pediátrico

Considerações Ambientais

- Otimização computacional para reduzir consumo energético

- Modelo eficiente sem degradação de performance
 - Infraestrutura cloud sustentável
-

6. Lições Aprendidas e Reflexões Finais

Experiência com Metodologia AIDesign

A metodologia AIDesign e CRISP-DM se mostraram adequadas para projetos de IA médica, especialmente na fase de imersão para compreender o problema real. A abordagem iterativa permitiu ajustes contínuos baseados em feedback técnico e validação com especialistas.

Avaliação da Proposta de Valor

Valor Técnico Entregue:

- Redução de 56,4% no erro de predição
- Sistema completo de produção containerizado
- Arquitetura escalável e interpretável

Valor Comercial Projetado:

- Redução de 70% no tempo de análise
- ROI positivo em 12 meses
- Escalabilidade para múltiplos laboratórios

Pontos de Melhoria

1. **Planejamento de Dados:** Negociar acesso a dados locais desde o início
2. **Backup de Modelos:** Implementar versionamento robusto de modelos treinados
3. **Testes de Integração:** Validação com sistemas PACS reais
4. **Monitoramento:** Métricas de drift de modelo em produção

Aprendizados sobre IA na Prática

Desafios Técnicos

- **Qualidade de dados** é mais crítica que quantidade
- **Interpretabilidade** é fundamental em aplicações médicas
- **Robustez de sistema** é tão importante quanto precisão do modelo
- **Containerização** resolve 80% dos problemas de deployment

Desafios de Negócio

- **Validação regulatória** em saúde é complexa

- **Mudança cultural** em organizações médicas requer tempo
- **Demonstração de ROI** precisa ser clara e mensurável

Relato Individual

Desenvolvimento de Expertise:

- Aprofundamento em deep learning para imagens médicas
- Domínio de arquiteturas de atenção e interpretabilidade
- Experiência prática com deployment de modelos ML

Gestão de Projetos:

- Importância de planning adequado para acesso a dados
- Necessidade de backup e versionamento de artefatos críticos
- Valor da documentação técnica detalhada

Perspectiva de Produto:

- Compreensão do gap entre pesquisa e aplicação prática
- Importância da validação com usuários reais
- Necessidade de pensar em escala desde o início

7. Referências

Modelo e Arquitetura:

GREULICH, W. W.; PYLE, S. I. *Radiographic Atlas of Skeletal Development of the Hand and Wrist*. Stanford: Stanford University Press, 1959.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. Cambridge: MIT Press, 2016.

LARSON, D. B. et al. Performance of a deep-learning neural network model in assessing skeletal maturity on pediatric hand radiographs. *Radiology*, v. 287, n. 1, p. 313-322, 2017.

HALABI, S.; PREVEDELLO, L.; KAPATHY-CRAMER, J. RSNA Pediatric Bone Age Machine Learning Challenge. *Radiology*, [s. l.], 2018. Disponível em: <https://pubs.rsna.org/doi/10.1148/radiol.2018180736>.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: *IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, 2009, Miami. Proceedings [...]. IEEE, 2009. p. 248-255. Disponível em: <https://ieeexplore.ieee.org/document/5206848>.







SIEGEL, E. L. What can we learn from the RSNA pediatric bone age machine learning challenge? *Radiology: Artificial Intelligence*, v. 1, n. 5, e180008, 2018. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/30615556/>.

SAADI, M.; ALJOBOURI, H. K.; AL-WAELY, N. K. Improving Bone Age Assessment with Inception-V3 and Faster R-CNN. In: *Innovative and Intelligent Digital Technologies; Towards an Increased Efficiency*. Cham: Springer, 2024. p. 579–590. Disponível em: https://link.springer.com/chapter/10.1007/978-3-031-70399-7_44.

LI, X. et al. Deep Learning Attention Mechanism in Medical Image Analysis: Basics and Beyonds. *International Journal of Neurological Disorders and Imaging*, [s. l.], 2021. Disponível em: <https://www.sciltp.com/journals/ijndi/article/view/173>.

HAFIZ, A. M.; GANGAL, A. Attention mechanisms and deep learning for machine vision: A survey of the state of the art. *Research Square*, [s. l.], 2021. Disponível em: <https://arxiv.org/abs/2106.07550>.

8. Apêndices (opcional)

<input type="checkbox"/>	 0 Open	<input checked="" type="checkbox"/> 5 Closed	Author ▼
<input type="checkbox"/>	 feat: Containerizing application and updating documentation	#5 by bernardomelo was merged 1 hour ago	
<input type="checkbox"/>	 fix: Reorganizing files directories and adding class to be used in the future	#4 by bernardomelo was merged 1 hour ago	
<input type="checkbox"/>	 feat: Functional model handler class and current trained model (it is corrupted)	#3 by bernardomelo was merged 2 hours ago	
<input type="checkbox"/>	 feat: Functional endpoint with pre-processing class integrated	#2 by bernardomelo was merged 2 hours ago	
<input type="checkbox"/>	 Adding requirements.txt to repo for dependency control	#1 by bernardomelo was merged 2 hours ago	

Log de PRs integradas no projeto, log de commits disponíveis no link do repositório.