Fundação Getulio Vargas
Escola de Matemática Aplicada

Redes Neurais e Aprensizado profundo -
Mestrado de Matemática Aplicada e Ciência de Dados

Relatório Final do Projeto - SSL para imagens médicas

Guilherme Moreira Castilho
Samuel Corrêa Lima

Rio de Janeiro
Dezembro / 2025

Conteúdo

Resumo

Este é o relatório final do projeto principal da matéria. A seção 2 é remanescente dos primeiros relatórios. As novidades do atual relatório se fazem presentes a partir da seção 3.

1 Introdução

O intuito deste trabalho é aplicar Self-Supervised Learning (SSL) para problemas de segmentação de imagens médicas. Para isso, seguiremos o artigo "Self-supervised learning for label-free segmentation in cardiac ultrasound"[1] como base.

No decorrer desse relatório, explicaremos sobre as opções estudadas, decisões tomadas o que foi feito até aqui e próximos passos. Na seção 2, é apresentada a inspiração do trabalho, resumindo o *paper* base. Na seção 3, discutimos diretamente nosso trabalho.

2 Self-supervised Learning for Label-free Segmentation in Cardiac Ultrasound

2.1 Métodos do Artigo

No estudo foram utilizadas imagens de ecocardiogramas para desenvolver um pipeline de SSL capaz de segmentar as câmaras cardíacas nas vistas apical de 2 câmaras (A2C), apical de 4 câmaras (A4C) e eixo curto médio (SAX). O processo iniciou-se com a geração de rótulos fracos obtidos por técnicas de visão computacional e informações anatômicas estatísticas, que serviram de base para o treinamento de redes neurais profundas. O pipeline seguiu um ciclo de aprendizado progressivo, no qual as redes refinavam suas próprias previsões até atingir segmentações estáveis, utilizadas posteriormente para calcular medidas estruturais e funcionais cardíacas conforme recomendações clínicas.

O conjunto de dados principal foi composto por 8.843 ecocardiogramas desidentificados da UCSF, abrangendo casos clínicos diversos, sem exclusão por qualidade de imagem. Para treinamento e validação, foram usados 2.228 vídeos (93 mil imagens), enquanto 8.393 ecocardiogramas (4,47 milhões de imagens) compuseram o conjunto de teste. Um conjunto externo adicional, oriundo do repositório EchoNet-Dynamic, forneceu 20.060 imagens A4C com anotações manuais do ventrículo esquerdo, permitindo avaliar a generalização do modelo.

Durante o pré-processamento, as imagens DICOM foram normalizadas, redimensionadas e convertidas para escala de cinza padronizada (0-1). As regiões de interesse foram extraídas e preparadas com bibliotecas OpenCV, scikit-image, SciPy e NumPy. Para a segmentação, foi utilizada uma arquitetura UNet modificada, com função de ativação sigmoide e otimizador Adam (taxa de aprendizado de 1e-4). Foram aplicadas diversas técnicas de augmentation, incluindo rotações, zoom, ruído gaussiano e ajustes de contraste. Em paralelo, uma rede de detecção de bordas HED (Holistically Nested Edge Detection) foi ajustada com pesos pré-treinados no ImageNet, a fim de melhorar a precisão das fronteiras das câmaras.

O controle de qualidade das segmentações envolveu análise geométrica das câmaras, eliminando formas incoerentes ou fora dos padrões anatômicos. O treinamento foi monitorado pelo TensorBoard, interrompendo-se automaticamente quando o modelo atingia o ponto de "cotovelo" na curva de perda Dice (momento em que a rede começaria a memorizar ruído). Essa estratégia foi complementada por etapas de autoaprendizagem, nas quais o modelo utilizava suas próprias previsões de maior qualidade para refinar o treinamento subsequente.

As segmentações finais foram empregadas para calcular volumes, áreas e frações de ejeção por métodos clínicos padronizados (biplano de discos e área-comprimento). Os quadros de sístole e diástole foram determinados automaticamente por ajuste senoidal das variações de área ao longo dos vídeos, garantindo maior consistência entre as medidas. Por fim, a avaliação estatística foi conduzida

com correlação de Pearson, regressão linear e análises de Bland-Altman, além de métricas como coeficiente de determinação (r^2), Kappa de Cohen e Dice score. Todos os testes foram realizados com o pacote SciPy, com intervalos de confiança de 95% estimados por bootstrapping.

2.2 Resultados do Artigo

Os autores desenvolveram um pipeline de SSL para segmentação de ecocardiogramas, capaz de gerar medições cardíacas precisas sem necessidade de rótulos manuais. O modelo foi treinado com 93 mil imagens e testado em mais de 4 milhões, incluindo um conjunto externo e um subconjunto com ressonância magnética cardíaca (padrão ouro). O pipeline utiliza rótulos fracos gerados por visão computacional e os aprimora progressivamente, alcançando coeficientes de determinação (r^2) entre 0,53 e 0,81, além de forte correlação com as medições clínicas em todas as câmaras cardíacas.

Os resultados mostraram que as medidas obtidas pelo método autossupervisionado apresentaram desempenho semelhante ao aprendizado supervisionado e à variabilidade clínica humana, com pontuação Dice média de 0,89 e precisão de 0,85 para detecção de câmaras anormais. Na comparação com a ressonância magnética, o pipeline manteve correlações consistentes (r^2 entre 0,60 e 0,73), evidenciando boa generalização e robustez. Assim, o estudo demonstra o potencial do aprendizado autossupervisionado para automatizar análises cardíacas complexas de forma escalável, reduzindo a dependência de anotações manuais.

3 A Escolha do Problema

Como dito, o Self-Supervised Learning é uma ferramenta poderosa, que pode ser muito bem usada na medicina. Com isso em mente, nosso trabalho se baseia no artigo para utilizarmos SSL em alguma outra gama de imagens médicas.

O artigo base, possuia grandes bancos de imagens para poder treinar e testar os modelos. Para este trabalho, buscamos bases de dados onde pudemos trabalhar com os recursos que temos.

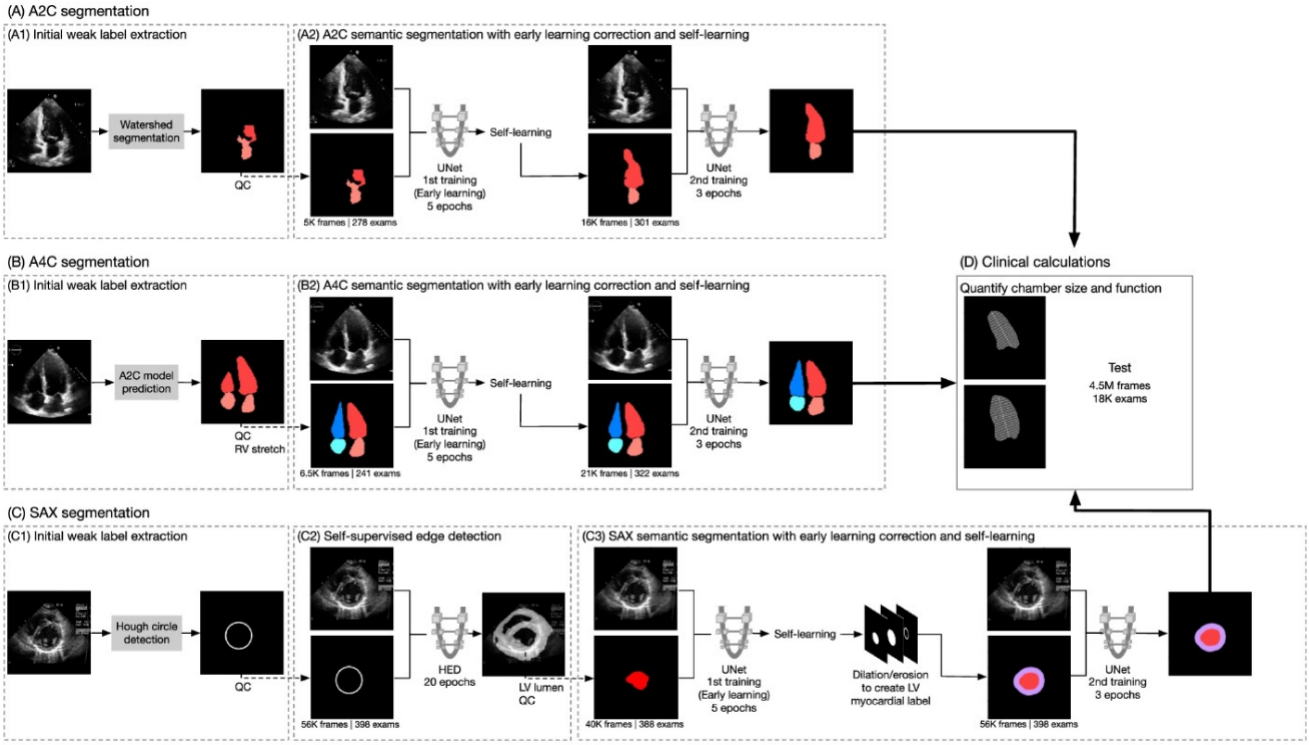
Buscamos diversos modalidades de imagens médicas, principalmente as utilizadas no Grand Challenge. Considerando as limitações de maquina/nuvem, além do contexto do trabalho com SSL, o problema escolhido foi a segmentação de imagens de raio X do tórax, com foco nos pulmões. A ideia inicial foi de usar um dataset de 40992 imagens **Chest X-Ray Images (Pneumonia)**[2] disponível no kaggle, que possuia imagens dos pulmões de pessoas com e sem pneumonia (como explicado no primeiro relatório).

Nosso objetivo é aplicar SSL para que o modelo aprenda a identificar estruturas anatômicas pulmonares, destacar regiões opacas ou infiltradas (padrões de pneumonia) e gerar mapas de calor indicando áreas suspeitas.

4 Arquitetura

Seguimos com as ideias dos relatórios anteriores para a implementação. O foco principal foi seguir a pipeline apresentada no *paper* (explicação na seção seguinte), então nossa implementação se baseia em UNets.

4.1 Pipeline SSL do *paper*



O pipeline é organizado em três grandes blocos (A2C, A4C e SAX), mas todos seguem a mesma lógica: primeiro extrair *weak labels* iniciais, depois refiná-las usando early-learning, e por fim melhorar a segmentação via self-learning.

A2C (A1) – Watershed Segmentation:

- Aplica watershed para separar as principais regiões ecográficas.
- Gera um rótulo aproximado das câmaras cardíacas.
- Realiza um processo de controle de qualidade (QC) antes de seguir.

A4C (B1) – Predição usando modelo A2C:

- Treina um modelo rápido usando as *weak labels* geradas na etapa A2C.
- Usa este modelo para predizer rótulos iniciais para as imagens A4C.
- Filtra regiões incorretas (ex.: alongamento anômalo do ventrículo direito).

SAX (C1) – Hough Circle Detection:

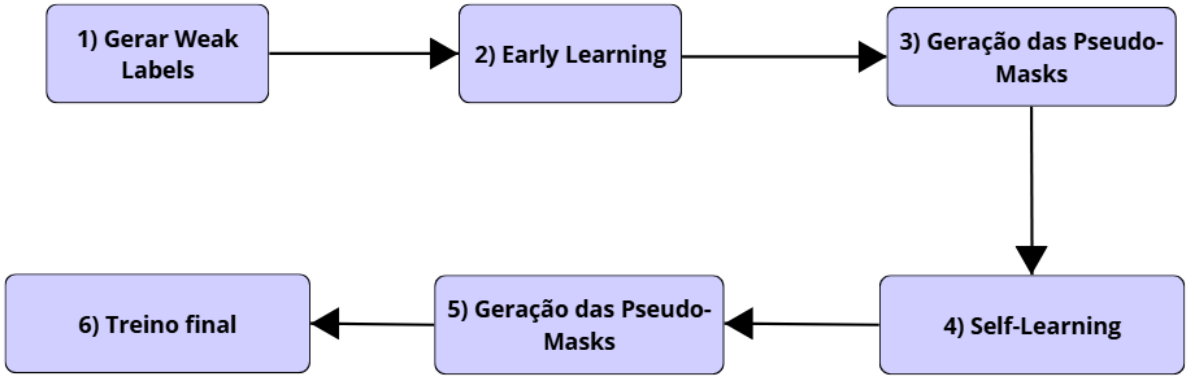
- Usa transformada de Hough para detectar círculos, representando o ventrículo esquerdo em cortes transversais.
- Isso produz uma máscara grosseira do contorno cardíaco.
- Aplica QC para remover falhas óbvias.

4.2 Nossa Pipeline

Como nosso problema se baseia em imagens diferentes do *paper* precisamos fazer algumas mudanças na pipeline apresentado acima.

No geral, mantivemos a mesma estrutura de pipeline, mudando as transformações que geram as weak labels. Além disso o *paper* não foi muito claro quanto ao processo de **Self-learning**. Demais pontos serão abordados nas sub-seções abaixo e na seção de dificuldades.

A seguir uma representação simples da nossa pipeline:



4.2.1 Obtenção das Weak Labels

De acordo com a metodologia proposta no artigo, o objetivo é gerar *weak labels* sem o uso de anotações humanas. Para isso, os autores partem das próprias imagens e aplicam uma sequência de transformações destinadas a realçar aspectos relevantes da anatomia cardíaca. Em seguida, utilizam limiares para identificar regiões fortemente e fracamente representadas, descartando as regiões pouco confiáveis. O processo resulta em vetores binários que representam, de forma aproximada, a anatomia cardíaca em cada imagem. Esses vetores constituem as *weak labels* (ruins), que posteriormente são utilizadas como máscaras auxiliares no treinamento.

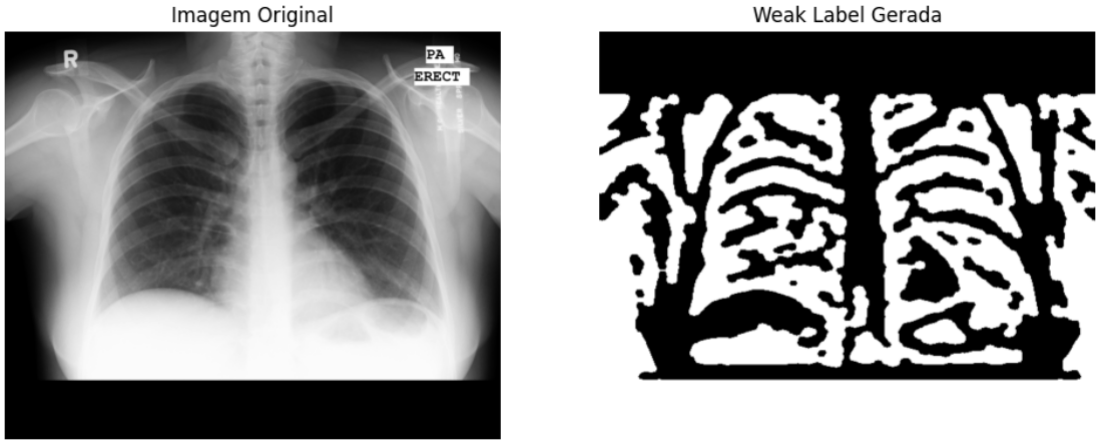
Para as obter as *weak labels* iniciais assim como o *paper* faz, nós criamos a função **generate_weak_mask**, que procura reproduzir essa ideia, adaptando-a ao contexto de radiografias de tórax. O pipeline adotado é o seguinte:

- **Leitura da radiografia em tons de cinza com redimensionamento das imagens para (512x512):** garante um formato adequado para operações (ao fim o tamanho é restaurado).
- **Realce de contraste e suavização:** aplica-se equalização adaptativa de histograma (CLAHE) seguida de um desfoque Gaussiano, de modo a melhorar a visibilidade das estruturas internas e reduzir ruídos.
- **Supressão de linhas:** utiliza-se abertura morfológica com elementos estruturantes lineares em diferentes ângulos para atenuar padrões lineares (como costelas), preservando regiões mais homogêneas.
- **Inversão e filtragem:** a imagem é invertida para destacar opacidades pulmonares e suavizada com filtro bilateral, que reduz ruído mantendo bordas relevantes.
- **Segmentação inicial:** aplica-se um limiar binário simples (threshold) para separar regiões de interesse. Em seguida, são removidas áreas do topo e das laterais que tipicamente não correspondem ao pulmão.

- **Refinamento morfológico:** operações de abertura e fechamento com elemento elíptico suavizam os contornos e eliminam pequenas imperfeições.
- **Seleção de componentes conectados:** calcula-se os componentes conectados e mantém-se apenas regiões internas com área suficiente e posição plausível (descartando objetos pequenos ou muito superiores).
- **Remoção de artefatos:** aplica-se remove_small_objects para eliminar estruturas residuais como ombros ou silhuetas laterais.

As transformações aplicadas atualmente diferem das apresentadas no relatório 2, veja a comparação:

Transformações antigas



Transformações atuais

Apesar da grande melhora, as weak labels geradas por essas transformações são instaveis, se comportando mal em algumas imagens. Essa instabilidade se refleriu também quando mudamos algumas transformações, onde percebiamos que melhorava em imagens coom certas características e piorava em outras. Esse foi um dos principais desafios que enfrentamos.

4.2.2 A UNet Utilizada

A UNet foi utilizada para a tarefa de segmentação na pipeline: A rede foi implementada com a camada de saída de 1×1 (1×1 output layer) ativada por sigmoide.

Os parâmetros de treinamento para a UNet foram definidos da seguinte forma:

- Otimizador: Adam;
- Learning Rate: 10^{-4};
- Loss Function: Dice loss;
- Batch Size: 32;

• Divisão dos Dados: 80% dos dados foram usados para treinamento e 20% para validação, com a divisão feita por paciente.

Todos os passos da pipeline utilizaram essa mesma configuração de UNet, diferindo que a última rede da pipeline foi treinada com 3 épocas e não 5 como as demais.

Para facilitar nosso trabalho utilizamos uma UNet pré-treinada no ImageNET.

4.2.3 Augmentation

Para aumentar a robustez do modelo e reduzir o risco de overfitting, aplicamos técnicas de data augmentation nas radiografias e suas máscaras correspondentes.

Como as imagens de Raio X do tórax seguem um padrão, a ideia foi utilizar pequenas ttransformações, principalmente de posição, para melhorar a robustez sem que as imagens fujam de um possível cenário real.

As transformações feitas foram: **Espelhamento Horizontal**; **Translações leves**; **Rotações de até 5º**; **Alteração de Brilho e Contraste**.

4.2.4 Quality Control (QC)

Para filtrar as weak labels, a fim de melhorar o treinamento, o *paper* utiliza o **QC**. Sua implementação não foi especificada, então resolvemos utilizar algumas métricas simples da imagem, como o **tamanho** dos dois maiores objetos – idealmente os pulmões –, a **ecentricidade**, o **extent** – quanto a mascara preenche a imagem –, o **solidity** – medida de convexidade, que vai indicar o quanto a mascara está fragmentada –, e o **aspect ratio** – controlar o formato dos objetos –, e por fim, o número de objetos na imagem. Todas essas medidas foram avaliadas com valores entre 0 e 1, com o mesmo peso. Por fim o QC Score é a média dessas medidas.

Para saber se uma máscara é aprovada ou não, é passado um threshold no QC Score.

4.2.5 Early Learning

Uma UNet, como já descrevemos, recebe imagens aprovadas do filtro QC e treina por 5 épocas.

Após isso é feita a previsão de todas as máscaras no set completo e aplicado novamente o filtro QC para passar para o **self-learning**

4.2.6 Self-Learning

O *paper* não foi claro quanto a forma que foi feito o self-learning. Em consequencia disso, e a excassez de informações sobre esse método em outros *papers*, essa foi a etapa mais desafiadora.

Nossa implementação de **self-learning** funcionou da seguinte forma:

• É treinada uma UNet com os pesos da anterior com os dados de treino filtrados;

• Uma etapa de geração de pseudo-mascaras é feita com as previsões dessa UNet treinada em todo o set de treino;

• Essa imagens passam pelo filtro QC;

• Entram novamente na UNet com os pesos atualizados e o ciclo se repete.

Esse processo iterativo é feito até um número definido de vezes –escollhemos 5 iterações como máximo – ou até uma certa porcentagem das máscaras geradas serem aprovadas –Escolhemos 66%.

4.2.7 UNet final

Ao final, com mais mascaras de boa qualidade, é treinada uma última UNet, dessa vez apenas com 3 épocas.

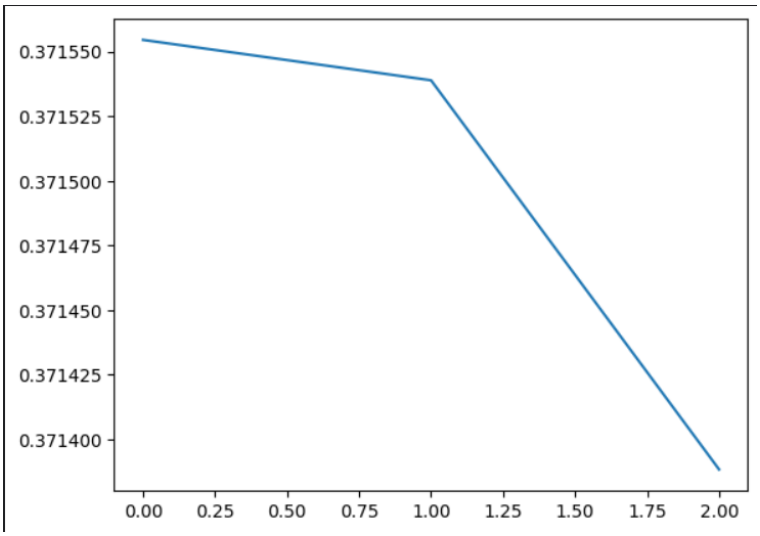
Ao fim do treinamento, o modelo está pronto para gerar máscaras automaticamente com qualidade similar às feitas com supervisão.

5 Resultados

Após rodar a pipeline, tivemos resultados não muito animadores – na seção de dificuldades isso será melhor explicado.

A pipeline foi executada com um subset do nosso dataset – motivs na seção de dificuldades – com 10% do dataset completo.

Veja abaixo a loss da úlltima UNet:

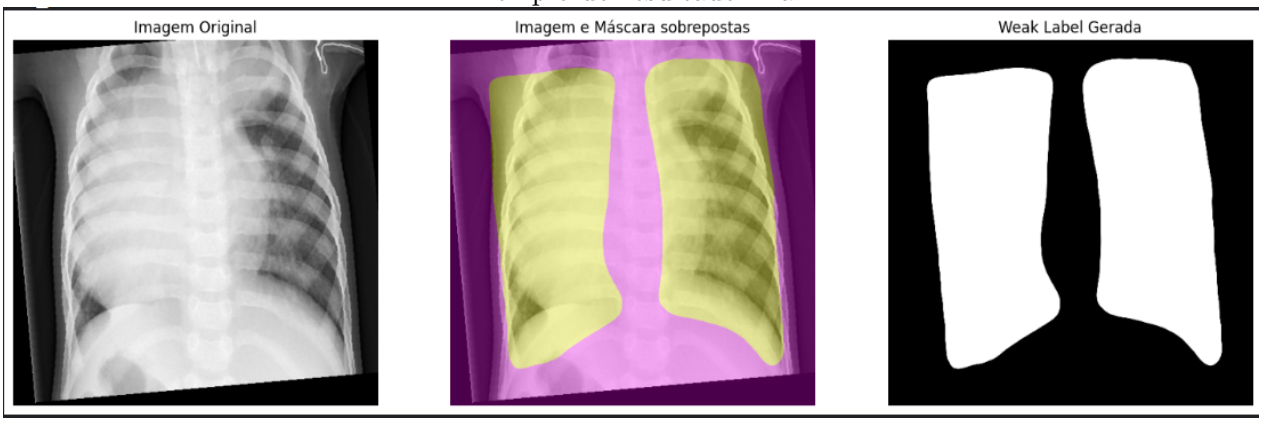


Considerando que a loss média da validação foi aproximadamente 1.17, vemos que o modelo não tem um mal comportamento e funciona estruturalmente falando.

O real problema do modelo é por conta dos parametros do QC Score principalmente – veja a seção de dificuldades.

Veja abaixo como o modelo prediz mascaras bem mais suavez, com contornos com formato bom, porém por conta do alto número de mascaras problematicas que não foram captadas pelo filtro QC, o modelo acaba tendo muito exemplos ruins, que fazem com que a mascara final não seja uma boa segmentação do pulmão. Veja:

Exemplo do resultado final



6 Dificuldades

Como dito anteriormente, tivemos problemas com partes não detalhadas do paper, ou partes que precisariam de auxilio de algum profissional da área.

O nosso trabalho está funcional e teoricamente bem fundado, porém diversos passos precisariam ser polidos para resultados como no paper:

• **Geração das weak labels:** O processo foi bem empirico e dificil de comparar as melhorias, além de que alguém mais especializado na área poderia ajudar a decidir quais transformações seriam feitas;

• **QC Score:** O problema é semelhante ao explicado acima: não sabemos o suficiente de anatomia do pulmão para saber quais métricas deveriam ser usadas ou mais "pesadas"na média;

• **Nº de dados:** Trabalhamos com uma base menor que no *paper* que teria 40992 imagens, porém por limitações de recursos, conseguiriamos treinar apenas um pedaço dessa base. E para agravar mais o problema, houveram erros nos ambientes virtuais que impediram que treinassemos com essa base, por não sobrar mais tempo. Por isso a base final utilizava apenas 10% do dataset;

• **Ainda o volume de dados:** Para gerar as weak labels para as 40000 imagens, houve um grande custo, o que fez com que criassemos uma nova base no kaggle já com as weak labels pré-processadas ((Chest X-Ray Weak Labels Dataset);

Por fim há de se destacar o maior desafio: entender a lógica por trás do **self-learning**. Não encontramos muito sobre este método e o *paper* fala muito brevemente e sem detalhes. Com isso, após muitas buscas, chegamos a conclusão, a partir do pequeno trecho do paper, que seria um processo iterativo utilizando o **QC**.

Houve também coisas que gostariamos de ter implementado (ou incluído) no nosso modelo que por conta dos problemas listados não conseguimos:

• **Self-learning com pesos:** Pensamos que poderia ser uma ideia viável atribuir o QC Score como pesos na loss para que imagens mais confiaveis influenciassem mais no treino. Chegamos a implementar, mas não pudemos testar – mesmo se testassemos, o resultado carregaria o problema do QC Score não estar polido;

• **Comparar com supervisionado:** Gostariamos de ter testado as mascaras geradas após a nossa pipeline com alguma rede de semantic segmentation e comparar com o resultado de usar máscaras já prontas;

• **Pipeline divida:** O *paper* divide o trabalho de segmentação em 3 pipelines, uma para cada parte do coração que definiram como importantes para segmentação. Gostariamos de ter feito isto também, segmentando primeiro um pulmão e depois o outro;

• **Arquiteturas:** Presente na seção de próximos passos do relatório anterior, estava testar outras arquiteturas, como variações de UNet.

Referências

[1] Zaynaf Salaymang Rima Arnaout Danielle L. Ferreira, Connor Lau. Self-supervised learning for label-free segmentation in cardiac ultrasound. *Nature Communications*, 16:4070, 2025.

[2] Paul Timothy Mooney. Chest x-ray images (pneumonia). *Kaggle*, 2018.