



Fundação Universidade Federal do ABC

Pró reitoria de pesquisa

Av. dos Estados, 5001, Santa Terezinha, Santo André/SP, CEP 09210-580

Bloco L, 3º Andar, Fone (11) 3356-7617

iniciacao@ufabc.edu.br

Relatório final do projeto de Iniciação Científica submetido no Edital: 01/2024
(PIC/PIBIC/PIBITI/PIBIC-AF)

Nome do aluno: Sarah Fernandes de Souza

Nome do orientador: John Andrew Sims

Título do projeto: Desenvolvimento de um Software para Avaliação da Homogeneidade e Relação Sinal/Ruído em Scanners de Imagens por Ressonância Magnética.

Palavras-chave do projeto: Determination of SNR in MRI, Hhomogeneity in MRI, Uniformity testing of MRI, Periodic testing of MRI, Quality assurance for MRI systems.

Área do conhecimento do projeto: Engenharia biomédica.

Bolsista: Não.

Sumário

1	Resumo	3
2	Introdução	3
2.1	Bobinas de Radiofrequência	3
2.2	Controle de qualidade	3
2.3	Ruído em imagens de ressonância	4
2.4	Relação sinal ruído (RSR)	5
2.5	Distorções geométricas	6
2.6	Transformada de Hough	7
3	Objetivos	7
4	Metodologia	7
4.1	Materiais	7
4.1.1	Imagens disponíveis para análise	7
4.1.2	Ambiente e linguagem de programação	8
4.2	Fluxograma	8
4.2.1	Leitura das imagens	9
4.2.2	Visualização das imagens	9
4.2.3	Análise do ruído	9
4.2.4	Filtragem das imagens	10
4.2.5	Limiarização da imagem	10
4.2.6	Transformada de Hough - Detecção de círculos	10
4.2.7	Medições para avaliação da Qualidade	10
4.3	Pipeline completo	11
4.3.1	Entradas	11
4.3.2	Saídas	12
5	Resultados e Discussão	12
6	Conclusão	14
7	Referências	15
8	Anexos	17

1 Resumo

Este projeto teve como principal objetivo o desenvolvimento de um pipeline em Python para a avaliação da qualidade de imagens de ressonância magnética (RM). As imagens médicas são suscetíveis a problemas como ruído, inomogeneidade e artefatos, o que pode comprometer a qualidade do diagnóstico. O software realiza a leitura de imagens no formato NIfTI e calcula algumas métricas que podem ser utilizadas como parâmetros de qualidade. As imagens foram lidas dentro do ambiente de programação do Google collaboratory, avaliadas, filtradas, segmentadas e as medidas realizadas foram de distorção geométrica, relação sinal-ruído (RSR), homogeneidade, contraste, coeficiente de variação, se o ruído é o não estacionário e a esfericidade do fantoma nas imagens. Um dos principais benefícios do trabalho é a padronização da análise e a redução das medições manuais. O programa fornece ao usuário imagens para inspeção visual e duas tabelas com a extensão ".csv" contendo os resultados das métricas, as quais podem ser baixadas ao final da avaliação. Uma das tabelas contém a distorção do maior diâmetro, e a outra, os parâmetros para todos os cortes válidos. Os valores podem ser utilizados para comparar os cortes ao longo do tempo.

2 Introdução

2.1 Bobinas de Radiofrequência

As bobinas de radiofrequência (RF) utilizadas em MRI são componentes que transmitem potência de RF (excitar o núcleo do ^1H no corpo) e recebem (detectar o sinal emitido do núcleo) sinais de RF. Dentre os tipos de bobinas de RF para MRI, podemos classificá-las como transmissoras, receptoras e transceptoras (transmissoras e receptoras). Além disso, as bobinas podem ser categorizadas de acordo com a sua geometria e região alvo do corpo como de volume, superfície e multicanal. (GRUBER et al., 2018)

2.2 Controle de qualidade

Ressonância magnética, assim como as outras modalidades estão sujeitas a problemas de deterioração da qualidade, o que pode estar relacionada com limitações de hardware, duração do exame, movimentação do paciente, interações do paciente com os componentes elétricos, Eddy current e movimentação estocástica de elétrons livres presentes nas bobinas de RF e as consequências desse problemas são a adição de ruídos e artefatos às imagens. (AJA-FERNÁNDEZ; VEGAS-SÁNCHEZ-FERRERO, 2016)

O controle de qualidade (CQ) de imagens médicas na clínica ou hospital é essencial para garantir que a instrumentação está operando de forma adequada, produzindo imagens com qualidade suficiente para fins de diagnóstico ou encaminhamento do paciente para tratamento apropriado. (DOMINGOS, 2020; ACR, 2015) Normas nacionais e internacionais existem para determinar quais testes devem ser realizados na instrumentação, com qual frequência, e os resultados

mínimos para garantir sua conformidade. (NEMA, 2008; ANVISA)

2.3 Ruído em imagens de ressonância

O Ruído Gaussiano está presente nos dados antes da obtenção da “imagem magnitude”, portanto o espaço-k, que é a matriz de números complexos codificada em fase e frequência, apresenta ruído gaussiano. Quando a “imagem magnitude” é obtida por meio da Transformada de Fourier e do cálculo do módulo dos valores complexos, o ruído deixa de ser gaussiano. (CAMPOS, 2020)

O arquivo final visualizado pelo médico para diagnóstico poderá apresentar duas possíveis distribuições: Rician, nas regiões com sinal, e Rayleigh nas regiões sem sinal. Essas distribuições ocorrem em imagens com alto RSR (Relação sinal ruído) obtidas com uma bobina de um canal. (CAMPOS, 2020).

Uma medida bem utilizada para medir a qualidade da imagem é a relação sinal/ruído (RSR) de RM. Porém, não existe nenhum método aceito universalmente para medir a RSR em todas as situações. O perfil de ruído depois da reconstrução da imagem é uma distribuição Rician. Porém, perto do centro do fantoma, onde a RSR é alta, o ruído tende à distribuição Gaussiana. No ar, fora do fantoma, o ruído pode ser considerado de Rayleigh. (AJA-FERNÁNDEZ; VEGAS-SÁNCHEZ-FERRERO, 2016) Análise das distribuições de ruído em partes diferentes da imagem poderia mostrar possíveis problemas na sua aquisição. As normas de National Equipment Manufacturers Association mostram vários métodos para determinar RSR a partir de imagens de fantoma, e dois métodos foram considerados para o presente projeto, porém, considerando que não há mais de uma aquisição para cada um dos conjuntos de imagens, não foi possível aplicar o método 1, que envolve a subtração de dois cortes iguais obtidos com os mesmos parâmetros. Algumas outras medidas também foram realizadas, como, homogeneidade, contraste, borramento e o coeficiente de variação (CV), sendo todos formas de quantificar a inhomogeneidade do campo B1 calculado a partir dos pixels dentro do campo B1.

Homogeneidade Avalia a proximidade dos valores de pixels da imagem à diagonal da matriz GLCM, que é uma análise de textura que pode ser utilizada para obter características de contraste e de homogeneidade das imagens. O valor da homogeneidade aumenta à medida que os valores apresentam poucas variações entre si. (SCIKIT-IMAGE; LANCASTER; HASEGAWA, 2020).

$$Homogeneidade = \sum_{i,j=0}^{N-1} \frac{P(i,j)}{1+(i-j)^2} \quad (1)$$

Equação 1: Expressão para cálculo da homogeneidade. (SCIKIT-IMAGE).

O contraste é definido como a diferença na intensidade ou exposição da imagem entre regiões adjacentes e refere-se à capacidade de diferenciar tecidos ou objetos vizinhos (SCIKIT-IMAGE; LANCASTER; HASEGAWA, 2020).

$$Contraste = \sum_{i,j=0}^{N-1} P(i, j) \cdot (i - j)^2 \quad (2)$$

Equação 2: Expressão para cálculo do contraste. (SCIKIT-IMAGE).

O Coeficiente de variação avalia o quanto de variação há entre os valores dos pixels da imagem. Considerando que o fantoma é feito de material homogêneo, espera-se que a variação seja a menor possível.

$$CV = \frac{\sigma}{\mu} \cdot 100\% \quad (3)$$

Equação 3: Expressão para cálculo do coeficiente de variação.

2.4 Relação sinal ruído (RSR)

A avaliação do ruído nas imagens foi realizada por meio da seleção de 4 regiões de ruído na imagem, que podem ser chamadas de “NMROI”, que são ROIs que não possuem sinal, apenas ruído. (NEMA, 2008) Nessas regiões a distribuição do ruído é "Rayleigh", mas apenas se estiver sendo analisadas imagens de um canal. (CAMPOS, 2020) A figura 1 mostra como ficou a seleção das quatro regiões. E as distribuições que são as de RAYLEIGH e o RICIANO conforme pode ser observado na figura 2.

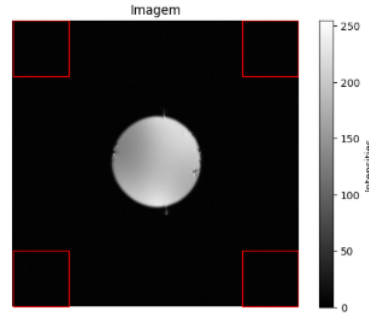


Figura 1: Imagem do fantoma obtida com a bobina millipede de um canal. Na imagem é possível observar em vermelho 4 regiões selecionadas, cada uma delas equivale a uma ROI que possui apenas ruído. (Fonte: Autor.)

- **Distribuição do ruído das regiões:** Considerando um sistema com apenas um canal o ruído deverá ter a mesma distribuição em todas as regiões e é um ruído estacionário, ou seja, o ruído varia com a intensidade do sinal, mas não varia com a posição do pixel, portanto, o desvio padrão e a média das regiões devem ser próximos. (AJA-FERNÁNDEZ; VEGAS-SÁNCHEZ-FERRERO, 2016).
- **Seleção dos pixels:** Os pixels selecionados não podem abranger a região sem sinal e as normas na NEMA para testes de qualidade, aconselha a seleção de ao menos 1000

pixels. Desta forma, cada região possui 250 pixels de área, que somados, dão os 1000 pixels necessários. (NEMA, 2008)

- **Ruído na região de sinal (MROI):** O ruído na ROI que engloba o sinal deve ter uma distribuição chama de RICIANA. (NEMA, 2008)
- **Ruído na região de ruído (NMROI):** O ruído nas NMROI é um caso particular do ruído RICIANO, chamado de RAYLEIGH. (NEMA, 2008)

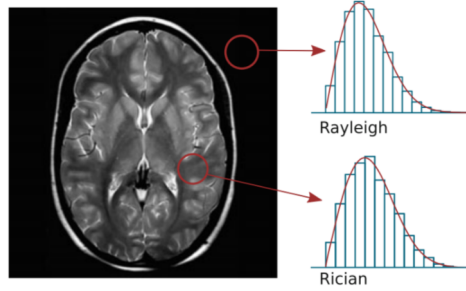


Figura 2: Imagem mostrando a distribuição de ruído esperada. Na região de sinal é a distribuição Riciana e a região de ruído de Rayleigh. (Fonte: AJA-FERNÁNDEZ; VEGAS-SÁNCHEZ-FERRERO, 2016).

Um importante parâmetro para avaliar a qualidade da imagem é a RSR e para o seu cálculo utilizou-se a equação 1. O valor “S” da equação equivale ao valor médio dos pixels da MROI, que é uma ROI que engloba ao menos 75 por cento da região de sinal do Fantoma, na equação esse valor também está representado como “mean(MROI)”. O valor “N” é o desvio padrão do ruído e também está representado na equação como “std(NMROI)”. (NEMA, 2008)

$$RSR = \frac{\text{mean}(MROI)}{\text{std}(NMROI) \cdot 0,66} = \frac{S}{N \cdot 0,66} \quad (4)$$

Equação 4: Expressão utilizada para o cálculo da relação sinal ruído da imagem.

2.5 Distorções geométricas

A distorção geométrica pode ser uma consequência do deslocamento dos pixels de uma imagem em relação à posição em que deveria se encontrar e um dos principais fatores de distorção é a falta de homogeneidade do campo magnético. (DOMINGUES, 2020). As distorções são um problema para as imagens médicas, visto que elas devem representar a estrutura analisada com a maior precisão possível, para permitir um bom diagnóstico. (ACR, 2015) Considerando a importância da correta representação das estruturas do corpo, o código irá avaliar as dimensões do fantoma e comparar com as dimensões reais fornecidas pelo usuário.

Além disso, também relacionado ao diâmetro, pode-se calcular a esfericidade das imagens, que seria o quanto elas se distanciam de um círculo perfeito.

2.6 Transformada de Hough

Uma forma eficiente de aplicar uma máscara e de obter informações da imagem é com a transformada de Hough para círculos. Para o seu cálculo são necessários três parâmetros, sendo eles duas coordenadas “x” e “y” e o raio da circunferência.(OPENCV) Conforme pode ser observado na equação 3.

$$C_{Hough}(x_{centro}, y_{centro}, r_{raio}) \quad (5)$$

Equação 5: Expressão utilizada para representar os círculos que são calculados por meio da transformada de Hough.

3 Objetivos

Os objetivos principais deste projeto de IC é avaliar a qualidade de imagens de ressonância magnética. Imagens médicas são frequentemente corrompidas por ruído e podem apresentar problemas de homogeneidade, contraste e artefatos. O presente trabalho irá realizar a leitura dessas imagens e devolver de forma automática parâmetros e métricas que podem ser utilizadas para garantir a qualidade das imagens.

4 Metodologia

4.1 Materiais

4.1.1 Imagens disponíveis para análise

As imagens disponibilizadas foram feitas com o uso de três tipos de bobinas de RF:

- Volumétrica - tipo milipede com 30 mm de diâmetro interno (**Figura 1a**);
- Volumétrica - tipo solenóide com 30 mm de diâmetro interno (**Figura 1b**);
- Multicanal com 8 canais exclusivos para recepção com base nos protocolos sugeridos pela ACR e pelo NEMA (**Figura 1c**).

O fantoma utilizado possui formato esférico com diâmetro de 25 mm e foi preenchido com solução aquosa de $CuSO_4$ (1 g/L) e NaCl (3,6 g/L), apresentando distribuição homogênea. Este foi posicionado no isocentro de cada bobina. As imagens foram coletadas a partir de um scanner pré-clínico de 9,4T (Bruker Biospec 94/30, Ettlingen, Germany), localizado no Department of Neurobiology, University of Pittsburgh, EUA.

As imagens do fantoma foram realizadas com as três bobinas, solenóide, millipede e 8 canais, foram adquiridas com ponderação em T1, e sequência de pulsos spin-eco com TR = 500 ms e TE = 20 ms. O campo de visão das aquisições (Field of View, FOV) foi de 50 mm × 50 mm e 1 mm de espessura de fatia, com tamanho de matriz 256 × 256 e 30 fatias no total.

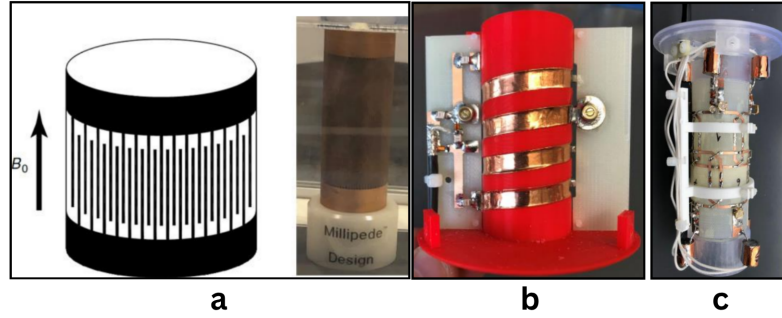


Figura 3: Imagens das três bobinas utilizadas para aquisição dos três conjuntos de imagens. (a) Bobina volumétrica millipede (esquerda), (b) Bobina volumétrica solenóide (central) e (c) Bobina multicanal com 8 canais receptora (direita).

4.1.2 Ambiente e linguagem de programação

Todo o código desenvolvido em Python 3 utilizando bibliotecas científicas e utilitárias padrão, como: skimage, pandas, glob, OpenCV, imageio, matplotlib, pydicom e numpy. OpenCV.HoughCircles implementa o HCT. O ambiente ou IDE utilizada para testar e rodar os códigos foi o Google colaboryatory.

4.2 Fluxograma

O fluxograma das funções pode ser observado na **figura 4** e cada uma delas será explicada a seguir.

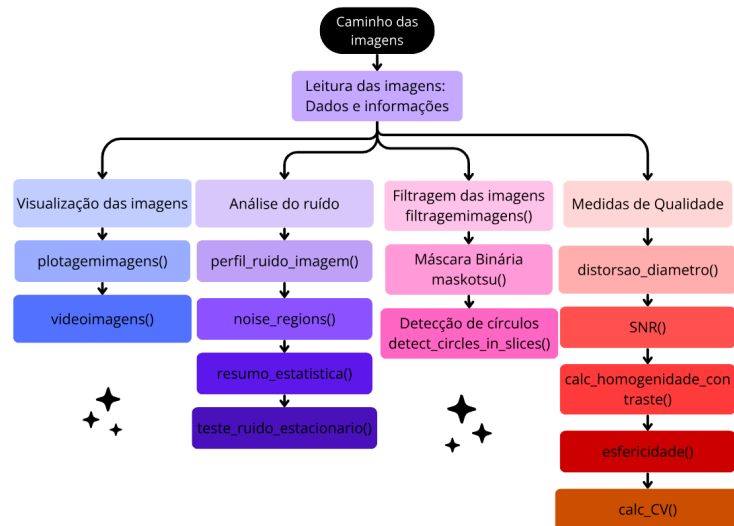


Figura 4: fluxograma das funções mostrando de forma geral todo o funcionamento do código, desde a leitura das imagens até o retorno do resultado final ao usuário

4.2.1 Leitura das imagens

A função para leitura das imagens utiliza a biblioteca Nibabel, que pode ser utilizada para a leitura de arquivos Nifti. Ela permite acesso aos dados da imagem em formato de um array e também permite que se acesse as informações contidas no cabeçalho do arquivo. (NIBABEL)

A entrada dessa função são duas Strings com os caminhos do arquivo da imagem e do ruído e a saída são os dados e informações das imagens e do ruído.

4.2.2 Visualização das imagens

A etapa de visualização das imagens pode ser realizada de duas formas, por meio do plot de todos os cortes em um grid contendo vários subplots, com a função "plotagemimagens", ou pode ser feita com a função "videoimagens", que permite a visualização dos dados em formato de vídeo.

4.2.3 Análise do ruído

A análise do ruído envolve algumas funções e análises:

1. Função **"perfil_ruído_imagem"**: Essa função recebe os dados do ruído que foram carregados com a biblioteca Nibabel e devolve três plots, que mostram o histograma geral do ruído (considerando uma média de todos os cortes), a função densidade de probabilidade (PDF) e a função de distribuição acumulada (CDF). O objetivo desse plots é a visualização da distribuição do ruído. Assim como já foi discutido na sessão de introdução, o ruído em ressonância magnética segue uma distribuição chamada Reylight, para o caso de apenas uma bobina. Apartir dos dados do histograma, é possível verificar se a distribuição é igual ou não à esperada. A média do ruído deu 115 e o desvio padrão 6.
2. Função **"noise_regions"**: realiza o que foi descrito no manual do NEMA em que 4 regiões são selecionadas para a análise do ruído. Com a seleção dessas quatro regiões, é possível verificar se o ruído é ou não estacionário, o que depende do número de bobinas da imagem. O ruído possui variância uniforme em todas as quatro regiões selecionadas, se a imagem do fantoma tiver sido realizada com uma bobina de apenas um canal.

Rodou-se o código para cada um dos conjuntos de imagens para verificar se o comportamento estacionário do ruído varia ou não com o número de canais nas bobinas.
3. Função **"resumo_estatistica"**: realiza os calculos de média e desvio padrão de todas as ROIs de cada um dos cortes. Todas elas foram comparadas.
4. Função **"teste_ruído_estacionario"**: É uma função que aplica o teste do quiquadrado nas quatro regiões selecionadas. Se o valor "p" for significativo, então podemos recusar a hipótese nula de que o ruído nas quatro regiões são iguais. E se não for significativo, são os casos em que falhamos em rejeitar a hipótese nula e o ruído é igual nas quatro ROIs.

4.2.4 Filtragem das imagens

A filtragem das imagens foi realizada com o uso de um filtro gaussiano. Esse filtro é baseado na distribuição normal, também chamada de distribuição gaussiana, e é muito utilizado em casos em que a distribuição, mesmo não sendo gaussiana, pode ser aproximada para tal distribuição. O papel da filtragem é permitir a melhor aplicação de outras funções, como a de limiarização da imagem para a obtenção de uma máscara binária, detecção de bordas.

4.2.5 Limiarização da imagem

O método escolhido para a formação das máscaras binárias foi método de Otsu. Esse método baseia-se na identificação de classes no histograma da imagem e determinar um valor intermediário entre as classes que permita a melhor separação possível entre ambas. O seja, ele devolve, de forma automática qual valor pode ser usado para separar a região de sinal da região de ruído.

4.2.6 Transformada de Hough - Detecção de círculos

A detecção de círculos com a transformada de Hough pois os fantasmas utilizados para testar as bobinas são circulares. Supondo que não há tanta distorção no formato circular do fantoma nas imagens, a detecção irá abranger maior parte da região de sinal.

4.2.7 Medições para avaliação da Qualidade

As medições foram realizadas para que se possa ter um parâmetro de como as imagens estão ao longo do tempo. Mais de uma medição pode ser realizada com o código a fim de se detectar uma queda na qualidade das imagens, portanto, o uso das medições sem um parâmetro ou referência pode não ser adequado. As principais medições realizadas e disponibilizadas em formato CVC para download são:

1. **Distorção do maior diâmetro:** Considerando que o maior diâmetro é conhecido, essa medição avalia o quanto próximo do ideal está o maior diâmetro capturado pelas imagens.
2. **SNR - Relação sinal Ruído:** Cálculo da relação entre o ruído e o sinal da imagem. O objetivo é sempre que o sinal seja maior que o ruído na imagem.
3. **Homogeneidade e Contraste:** Avalia o quanto homogêneo são os valores dentro da região de sinal e se o fantoma pode ser bem visualizado. Imagens com bordas mais bem definidas possuem maior contraste.
4. **Limiar utilizado pelo método de Otsu:** Define o valor de ton de cinza utilizado para formação das máscaras em cada corte e também representa um piso para os valores da região de sinal da imagem.

5. **Esfericidade:** Avalia o quão distante o fantoma está de ser uma esfera, como era esperado. O código calcula dois valores de raio, que deveriam ser iguais. A menor e a maior medida do raio são utilizadas para o cálculo.
6. **Coefficiente de Variação:** Relacionado à homogeneidade das imagens, quando mais perto de zero, maior é a homogeneidade dentro da região de sinal.
7. **Estacionaridade do Ruído:** Confere se as imagens apresentam ruído estacionário ou não. Sendo que essa característica varia de acordo com o número de canais da bobina utilizada.
8. **Distribuição do ruído:** Confere se as imagens apresentam ruído com a distribuição de rayleigh.

4.3 Pipeline completo

Este pipeline tem como objetivo executar uma análise completa da qualidade de imagens de ressonância magnética. O processo inicia-se com a leitura dos dados de entrada e avança até o cálculo e a compilação de métricas de qualidade, como SNR (Relação Sinal-Ruído), distorção e esfericidade. Para tal, ele foi separado em oito etapas:

- ETAPA 1: Leitura das imagens.
- ETAPA 2: Visualização das imagens.
- ETAPA 3: Análise do ruído.
- ETAPA 4: Filtragem das imagens.
- ETAPA 5: Criação de máscaras binárias.
- ETAPA 6: Detecção de círculos.
- ETAPA 7: Cálculo de métricas.
- ETAPA 8: Compilação dos resultados.

4.3.1 Entradas

Para a execução do pipeline, os seguintes parâmetros são necessários:

- **Caminho da Imagem e do Ruído:** O caminho para os arquivos NIfTI da imagem e do ruído, representados por `imagem_path` e `ruído_path`, respectivamente.
- **Detalhes Geométricos:** O Campo de Visão (FOV) em milímetros, o diâmetro real do objeto em milímetros e a dimensão da matriz em pixels.
- **Dimensão das ROIs:** A dimensão das Regiões de Interesse (ROIs) do ruído, em pixels.

- **Configurações de Saída:** Duas variáveis booleanas controlam a geração de elementos visuais: `mostrar_plots` para exibir gráficos intermediários e `gerar_videos` para criar vídeos do processo.

4.3.2 Saídas

Ao final da execução, o pipeline retorna dois DataFrames do Pandas contendo os resultados da análise:

- **DataFrame Final (`df_final`):** Um DataFrame completo que reúne todas as métricas de qualidade calculadas e organizadas de forma limpa.
- **DataFrame de Diâmetro (`pd_diametro`):** Um DataFrame específico com os resultados detalhados sobre a distorção do diâmetro.

5 Resultados e Discussão

Os resultados serão apresentados na ordem em que as funções foram sendo implementadas no pipeline completo do código. A primeira etapa consistiu no recebimento de dois arquivos NIfTI, e na apresentação das informações da imagem, como dimensões, tipo de dado e cabeçalho.

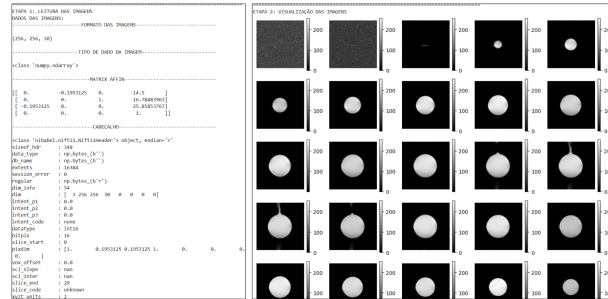


Figura 5: Enter Caption

A partir da leitura desses dados, a próxima etapa é a visualização dos cortes, que pode ser em um grid ou em vídeo. O grid é mostrado direto na saída do código, mas o vídeo, quando habilitado, pode ser baixado pelo usuário.

As próximas etapas envolvem a análise do ruído e algumas operações antes dos cálculos das métricas. Primeiro, o desvio padrão e a média do ruído das imagens são calculados, sendo que o vetor utilizado foi uma média geral de todos os cortes do arquivo "ruído_data". E 3 gráficos relacionados ao ruído foram plotados: um histograma, uma função de densidade de probabilidade (PDF) e uma função de distribuição acumulada (CDF).

Depois da análise visual do ruído, as imagens foram passadas por um filtro gaussiano para suavizar o ruído e preparar os cortes para uma melhor segmentação da região de sinal. O método utilizado para a montagem da máscara foi o de Otsu. E, por último, a Transformada de Hough foi usada para detectar os círculos nas imagens. O cálculo das métricas é então iniciado e os

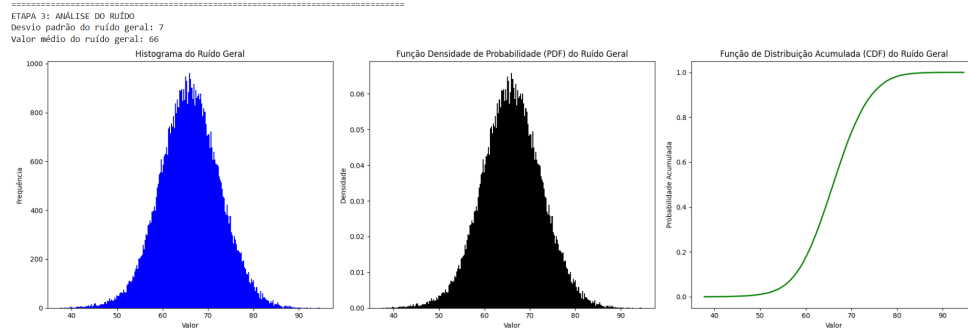


Figura 6: Informações sobre as imagens e cabeçalho (Etapa 1) e cortes das imagens plotadas em um grid (Etapa 2).

resultados vão sendo salvos em dataframes. A saída indicando a passagem por essas etapas pode ser visualizada na figura 7.

```

=====
ETAPA 4: FILTRAGEM DAS IMAGENS - Filtro Gaussiano
Vídeo das imagens filtradas
(256, 256, 30) <class 'numpy.ndarray'>
Maior valor de cinza: 255.0 e menor valor de cinza: 0.7
=====
ETAPA 5: MÁSCARAS BINÁRIAS - Método de Otsu
Vídeo das máscaras
(256, 256, 30) <class 'numpy.ndarray'>
Maior valor de cinza: 1.0 e menor valor de cinza: 0.0
=====
ETAPA 6: DETECTANDO CÍRCULOS - Transformada de Hough
Vídeo da detecção dos círculos
(256, 256, 24) <class 'numpy.ndarray'>
Maior valor de cinza: 252.0 e menor valor de cinza: 0.0
=====
ETAPA 7: CALCULANDO MÉTRICAS
1) Distorção do maior diâmetro
=====DISTORÇÃO DO MAIOR DIÂMETRO=====
Tamanho do pixel: 0.1953125
FOV: 50.00 mm
Diâmetro Real: 25.00 mm
Dimensão do pixel: 0.20 mm/pixels
Diâmetro máximo calculado = 24.61 mm
Porcentagem de Distorção do diâmetro = 1.59 %
=====
2) SNR
3) Homogeneidade e Contraste
4) Homogeneidade e Contraste
5) Esfericidade
6) Coeficiente de Variação

```

Figura 7: Imagem com os gráficos do ruído (Etapa 3): Da esquerda para a direita, há um histograma, a função de densidade de probabilidade (PDF) e a de distribuição acumulada (CDF).

O resultado final pode ser visualizado na figura 8. Há dois dataframes, um com as métricas dos cortes e um com as informações de distorção do maior diâmetro. Além disso, há também um exemplo de como os dados ficam ao serem disponibilizados no formato .csv. Os dados ficam separados por vírgulas.

index,Corte,Média,Image_Noise,SNR,SNR_db,Limiar,homogeneity,contrast,CV,Esfericidade,ESTACIONÁRIO											
5	6	188.23	74.85	2.51	8.01	94.13	0.66	19.69	3.02	98.24	Sim
6	7	206.44	92.73	2.23	6.95	103.1	0.64	32.07	3.36	98.72	Sim
7	8	216.0	105.67	2.04	6.21	108.08	0.63	46.2	3.74	98.49	Sim
8	9	220.25	114.63	1.92	5.67	110.07	0.61	57.26	4.06	98.55	Sim
9	10	196.06	106.86	1.83	5.27	98.12	0.66	56.74	4.33	98.57	Sim
10	11	120.71	120.71	1.77	4.97	107.09	0.63	73.41	4.46	98.62	Sim
11	12	193.58	112.04	1.73	4.75	97.12	0.65	67.59	4.77	98.7	Sim
12	13	209.46	123.41	1.7	4.59	105.09	0.61	83.34	4.95	98.46	Sim
17	18	212.19	125.73	1.69	4.55	106.08	0.62	90.89	4.66	98.48	Sim
18	19	216.1	126.3	1.71	4.67	108.08	0.62	87.92	4.43	98.5	Sim
19	20	183.02	104.6	1.75	4.86	92.14	0.66	58.25	4.16	98.49	Sim
20	21	223.14	124.05	1.8	5.1	112.06	0.63	77.76	3.94	98.65	Sim
21	22	208.17	111.3	1.87	5.44	104.09	0.66	59.71	3.67	98.49	Sim
22	23	211.7	107.62	1.97	5.88	106.08	0.66	52.05	3.51	98.61	Sim
23	24	232.84	110.66	2.1	6.46	117.04	0.63	47.65	3.34	98.89	Sim
24	25	182.5	78.94	2.31	7.28	91.14	0.71	24.08	3.21	98.77	Sim
25	26	218.8	82.18	2.66	8.51	110.07	0.65	23.19	3.09	98.43	Sim

index	tamanho_pixel_mm	FOV_mm	diametro_real_mm	diametro_max_calculado_mm	distorcao_diametro_percentual_%
0	0.1953125	50	25	24.609375	1.5873015873015872

Visualização de como fica o arquivo CVC

Figura 8: Saídas no console referentes iniciando com a etapa 4 e terminando com a etapa 7.

6 Conclusão

O pipeline pode ser utilizado para a avaliação da qualidade de imagem, no que diz respeito às métricas de ruído, contraste e homogeneidade, e o seu principal benefício, é a automação da extração dessas métricas, reduzindo a necessidade de medições manuais. Além disso, o código é composto por um conjunto de funções independentes, que podem ser ajustadas ou rodadas separadamente, se necessário.

Algumas limitações relacionadas ao projeto são o tipo de arquivo, que precisa ser em formato NIFTI, outros tipos de arquivo podem exigir outras funções para a leitura das imagens. O formato do fantoma precisa ser circular, pois os códigos de detecção de círculos e cálculo dos valores de raio levam em consideração apenas essa geometria. E a transformada Hough fornece apenas uma estimativa de onde há uma estrutura circular na imagem, mas os objetos imagiados não são círculos perfeitos, algumas regiões de sinal podem acabar sendo excluídas da delimitação, e regiões de ruído, incluídas.

Apesar de o projeto fornecer métricas quantitativas para avaliação da qualidade, a interpretação delas requer conhecimento especializado em ressonância magnética e processamento de imagens médicas, a fim de validar a relevância clínica dos valores e indicar possíveis problemas com as bobinas ou com o equipamento.

Não foi possível construir um site nem implementar uma função para validar os tipos de distribuições presentes nas imagens, o que poderia ser realizado em projetos futuros. Os próximos passos para uma implementação segura em ambientes hospitalares seriam validar e testar as funções com mais imagens para garantir que todos os passos estão funcionando como esperado, além da refatoração do código, para remover redundâncias e deixá-lo mais eficiente.

7 Referências

- GRUBER, B. et al.** RF coils: A practical guide for nonphysicists. *Journal of Magnetic Resonance Imaging*, v. 48, n. 3, p. 590–604, 13 jun. 2018.
- BALLARD, D. H.** Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, v. 13, n. 2, p. 111–122, 1981.
- CANNY, J.** A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. PAMI-8, n. 6, p. 679–698, 1986.
- GUDBJARTSSON, H.; PATZ, S.** The Rician distribution of noisy MRI data. *Magnetic Resonance in Medicine*, v. 34, n. 6, p. 910–914, Dec. 1995. DOI: 10.1002/mrm.1910340618.
- HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I.** Textural features for image classification. *IEEE Transactions on Systems, Man, Cybernetics*, v. 3, n. 6, p. 610–622, 1973.
- MAZZOLA, A. A.** Ressonância magnética: princípios de formação da imagem e aplicações em imagem funcional. *Revista Brasileira de Física Médica*, v. 3, n. 1, p. 117–129, 2009.
- PETROU, M.; SEVILLA, P. G.** *Image Processing: Dealing with Texture*. John Wiley Sons, Ltd, London, 2006.
- THAKUR, K. V.; DAMODARE, O. H.; SAPKAL, A. M.** Poisson Noise Reducing Bilateral Filter. *Procedia Computer Science*, v. 79, p. 861–865, 2016.
- GONZALEZ, R. C.; WOODS, R. E.** *Digital Image Processing*. 4. ed. New York, NY: Pearson, 2018.
- RANGAYYAN, R. M.** Image quality and information content. In: *Biomedical Image Analysis*, chapter 2, p. 39–84. CRC Press, 2005.
- DOMINGUES, C. V. J.** *Controle da qualidade de imagem em ressonância magnética*. Tese (Mestrado) - Universidade de Lisboa, Portugal, 2020.
- CAMPOS, V. P.** *Método baseado em estabilização de variância para filtragem de imagens volúmetricas de ressonância magnética corrompidas por ruído Riciano não-estacionário*. Dissertação UFSCAR, São Carlos, 2020. Disponível em: <https://www.teses.usp.br/teses/disponiveis/18/18152/tde-02022022-151520/pt-br.php>. Acesso em: 29 set. 2024.
- AMERICAN COLLEGE OF RADIOLOGY (ACR).** *Quality Control Manual*. Reston, VA, 2015. Disponível em: https://www.researchgate.net/publication/284188389_2015_ACR_MR_QCManual_Table_of_Contents. Acesso em: 16 fev. 2025.
- BRASIL. Agência Nacional de Vigilância Sanitária (ANVISA).** *Instrução Normativa nº 97, de 03 de dezembro de 2021*. Disponível em: https://bvsms.saude.gov.br/bvs/saudelegis/anvisa/2021/in097_27_05_2021.pdf. Acesso em: 25 maio 2024.

NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. *NEMA Standards Publication MS 1-2008 (R2014, R2020): Determination of Signal-to-Noise Ratio (SNR) in Diagnostic Magnetic Resonance Imaging.* Rosslyn, VA, 2008. Disponível em: www.nema.org. Acesso em: 16 fev. 2025.

ELSTER LLC. Website - MRI Questions: Signal-to-Noise Ratio in MRI, 2024. Disponível em: <https://mriquestions.com/index.html>. Acesso em: 29 maio 2024.

GITHUB. GitHub: Where the world builds software. Disponível em: <https://github.com/>. Acesso em: 16 fev. 2025.

OPENCV. Hough Circle Transform. Disponível em: https://docs.opencv.org/3.4/d4/d70/tutorial_hough_circle.html. Acesso em: 16 fev. 2025.

STREAMLIT. Streamlit Library. Disponível em: <https://streamlit.io/>. Acesso em: 16 fev. 2025.

SCIKIT-IMAGE. Plotting grey-level co-occurrence matrix properties. Disponível em: https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_glcmm.html. Acesso em: 16 fev. 2025.

NIBABEL. *NIfTI Images*. Disponível em: https://nipy.org/nibabel/nifti_images.html. Acesso em: 24 ago. 2025.

8 Anexos

ANEXO I - Documentação de arquitetura e design do código

Pipeline de processamento de imagens e análise de dados para avaliar a qualidade de imagens de ressonância magnética (MRI) no formato NIfTI. O objetivo principal é automatizar a extração de métricas de qualidade, como a Relação Sinal-Ruído (SNR), homogeneidade, contraste, distorção geométrica e esfericidade. As principais bibliotecas utilizadas foram a nibabel (para arquivos NIfTI), numpy (para computação numérica), matplotlib e seaborn (para visualização), scikit-image e opencv (para processamento de imagens) e pandas (para manipulação de dados tabulares).

Funções para leitura e visualização de dados		Entrada	Saída
Leituraimagens()	Carrega arquivos NIfTI de imagem e ruído, extraindo dados, matrizes afins e cabeçalhos	Strings (diretórios para os arquivos NIfTI).	Tupla contendo os objetos e dados de imagem e ruído.
Dadosimagens()	Imprime metadados e informações de forma para verificação.	Tupla contendo os objetos e dados de imagem e ruído.	Imprime informações das imagens no console.
plotagemimagens()	Plota todas as fatias (slices) de uma imagem em uma grade.	imagem_data (numpy.ndarray)	Exibe uma figura com as imagens plotadas.
videoimagens()	Cria um vídeo a partir das fatias de imagem, oferecendo uma forma alternativa de visualização.	imagem_data (numpy.ndarray)	Video com as imagens
Análise de Ruído		Entrada	Saída
perfilruídoimagem()	Calcula estatísticas globais de ruído (desvio padrão, média) e visualiza seu histograma e funções de distribuição (PDF e CDF)	ruído_data (numpy.ndarray)	Tupla: Desvio, média e ruído. Gráficos: Histograma, CDF e PDF do ruído.
noise_regions()	Extrai quatro Regiões de Interesse (ROIs) de ruído nos cantos da imagem.	Imagem: numpy.ndarray Coordenadas do Zoom str: "True" ou "False" (Plotagem)	roi1, roi2, roi3, roi4 (numpy.ndarray)
resumo_estatistica()	Calcula estatísticas (média e desvio padrão) para cada ROI de ruído e para o conjunto total.	roi1, roi2, roi3, roi4 (numpy.ndarray)	df_estatisticas (pd.DataFrame) roi (numpy.ndarray): ROIs concatenadas
teste_ruído_estacionario()	Realiza um teste de Qui-Quadrado para verificar se o ruído nas ROIs é estatisticamente semelhante, ou seja, se é estacionário.	roi1, roi2, roi3, roi4 (numpy.ndarray)	pd.DataFrame
Processamento e Segmentação de Imagem		Entrada	Saída
filtragemimagens()	Aplica um filtro Gaussiano para suavizar as imagens e reduzir ruído	imagem_data (numpy.ndarray)	imagem_data (numpy.ndarray) filtrado
maskotsu()	Cria uma máscara binária usando o método de limiarização de Otsu, que é fundamental para segmentar o objeto principal da imagem (o phantom).	(numpy.ndarray)	mask_otsu (numpy.ndarray) t_otsu (float)
Métricas de Qualidade		Entrada	Saída
detect_circles_in_slices()	Utiliza a Transformada de Hough para detectar o círculo principal na imagem, medindo seu diâmetro.	imagem_data (numpy.ndarray)	Círculos - (numpy.ndarray) diâmetro - (list)
distorsao_diâmetro()	Calcula a distorção geométrica do diâmetro em porcentagem, usando o diâmetro medido em pixels e o diâmetro real.	diâmetro (list), FOV (int), diâmetro_real (int), matrix_dim (int)	pd_diâmetro (pd.DataFrame)
SNR()	Calcula a Relação Sinal-Ruído (SNR) para cada fatia da imagem, com base na média dos pixels da Máscara de Região de Interesse (MROI) e no desvio padrão do ruído.	imagem_data (numpy.ndarray), mostrar_plots (bool)	df_snr (pd.DataFrame)
calc_homogeneidade_contraste()	Calcula homogeneidade e contraste usando as propriedades da Matriz de Co-ocorrência de Níveis de Cinza (GLCM).	imagem_data (numpy.ndarray)	df (pd.DataFrame)
esfericidade()	Calcula a esfericidade, que é a razão entre os eixos menor e maior de um objeto, usando a biblioteca scikit-image.	mask_otsu (numpy.ndarray), Plotar (bool)	df_esfericidade (pd.DataFrame)
calc_CV()	Calcula o Coeficiente de Variação (CV) para avaliar a uniformidade de sinal dentro do objeto.	mask_otsu (numpy.ndarray), imagem_data (numpy.ndarray)	df (pd.DataFrame)
pipeline_completo()		Descrição	
ETAPA 1	Leitura de Dados	Carrega os arquivos NIfTI	
ETAPA 2	Visualização (Opcional)	Plota ou gera vídeos das imagens.	
ETAPA 3	Análise de Ruído	Calcula estatísticas de ruído e testar a estacionariedade do ruído nas ROIs	
ETAPA 4	Filtragem	Aplica o filtro Gaussiano para pré-processar as imagens.	
ETAPA 5	Máscaras Binárias	Cria a máscara binária (Otsu) para isolar o objeto de interesse	
ETAPA 6	Deteção dos círculos	Detecta círculos para calcular a distorção	
ETAPA 7	Métricas de qualidade	Distorção do maior diâmetro, SNR, Homogeneidade e Contraste, Filtragem e Limiarização com Otsu, Esfericidade, Coeficiente de Variação.	
ETAPA 8	Compilação dos resultados e remoção de Outliers (remove_outliers())	Combina todos os resultados em um único DataFrame e salva em um arquivo CSV.	

ANEXO II - Orientações de Uso

Abaixo há algumas orientações para rodar a função que roda todas as outras funções em oito etapas. O código está disponível no link para consulta no link: [LINK!](#)

1. **Instale as bibliotecas necessárias:** O código possui um conjunto de bibliotecas que precisam ser instaladas antes da utilização do código. Para rodar localmente, crie um ambiente virtual e instale as versões mais recentes. Alguns casos podem exigir a instalação de uma versão mais antiga, visto que python é uma linguagem que está em constante aprimoramento. Algumas funções podem ser depreciadas e ou removidas.
2. **Rodando o código no colab:** Conecte-se ao Google Drive para carregar seus arquivos de imagem e ruído diretamente do Google Drive. Rodando localmente, basta criar um diretório com os arquivos.
3. **Execução do Pipeline:** A função principal para rodar todo o processo é a "pipeline_completo()", que inclui todas as etapas da análise. Para rodar a análise, basta chamar a função passando os caminhos dos seus arquivos NIfTI de imagem e ruído.

```
1 imagem_path = "/content/drive/MyDrive/Solenoide/Image_Phantom.nii"
2 ruido_path = "/content/drive/MyDrive/Solenoide/Noise.nii"
3
4 pipeline_completo(imagem_path, ruido_path, FOV=50, diametro_real=25, matrix_dim=256,
5                   dim_roi=50, mostrar_plots=True, gerar_videos=True)
6
7 #ou
8
9 df_final, pd_diametro = pipeline_completo(imagem_path, ruido_path, FOV=50, diametro_real=25, matrix_dim=256,
10                                           dim_roi=50, mostrar_plots=True, gerar_videos=True)
```

Figura 9: Exemplo de como rodar o Pipeline completo

Parâmetros da função:

- **imagem_path (obrigatório):** Caminho do arquivo NIfTI da sua imagem.
- **ruido_path (obrigatório):** Caminho do arquivo NIfTI da sua imagem de ruído.
- **FOV (opcional):** O Campo de Visão da imagem em milímetros (padrão: 50).
- **diametro_real (opcional):** O diâmetro real do objeto (phantom) em milímetros (padrão: 25).
- **matrix_dim (opcional):** A dimensão da matriz de pixels da imagem (padrão: 256).
- **dim_roi (opcional):** Dimensão das ROIs de ruído usadas na análise (padrão: 50).
- **mostrar_plots (opcional):** Defina como True para exibir gráficos de todas as fatias. O padrão é True.

- **gerar_videos (opcional):** Defina como True para gerar vídeos das fatias. O padrão é True.

Após a execução, a função retorna e salva dois DataFrames no diretório atual:

- **resultados_finais.csv:** Contém métricas como SNR, homogeneidade, contraste, esfericidade e CV para cada fatia.
- **distorcao_diametro.csv:** Apresenta a distorção geométrica do diâmetro.