

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE COMPUTAÇÃO E MATEMÁTICA

FELIPE CABRERA RIBEIRO DOS SANTOS

**RECONSTRUÇÃO DE IMAGENS DE
RESSONÂNCIA MAGNÉTICA COM DADOS
ESPARSOS ATRAVÉS DE INTELIGÊNCIA
ARTIFICIAL**

Ribeirão Preto–SP

2020

FELIPE CABRERA RIBEIRO DOS SANTOS

**RECONSTRUÇÃO DE IMAGENS DE RESSONÂNCIA
MAGNÉTICA COM DADOS ESPARSOS ATRAVÉS DE
INTELIGÊNCIA ARTIFICIAL**

Versão Original

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) da Universidade de São Paulo (USP), como parte das exigências para a obtenção do título de Mestre em Ciências.

Área de Concentração: Computação Aplicada.

Orientador: Luíz Otávio Murta Júnior

Ribeirão Preto–SP

2020

FELIPE CABRERA RIBEIRO DOS SANTOS

**RECONSTRUCTION OF MAGNETIC RESONANCE
IMAGES WITH SPARCE DATA THROUGH ARTIFICIAL
INTELLIGENCE**

Original Version

Dissertation presented to Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) from the Universidade de São Paulo (USP), as part of the requirements to hold the Master of Science degree.

Field of Study: Applied Computing.

Supervisor: Luíz Otávio Murta Júnior

Ribeirão Preto–SP

2020

Felipe Cabrera Ribeiro dos Santos

RECONSTRUÇÃO DE IMAGENS DE RESSONÂNCIA MAGNÉTICA COM DADOS ESPARSOS ATRAVÉS DE INTELIGÊNCIA ARTIFICIAL. Ribeirão Preto-SP, 2020.

65p. : il.; 30 cm.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da USP, como parte das exigências para a obtenção do título de Mestre em Ciências,
Área: Computação Aplicada.

Orientador: Luíz Otávio Murta Júnior

1. Imageamento por ressonância magnética (MRI). 2. Inteligência artificial (AI).
3. Dados esparsos. 4. Espaço K. 5. Reconstrução de imagens.

Felipe Cabrera Ribeiro dos Santos

RECONSTRUÇÃO DE IMAGENS DE RESSONÂNCIA MAGNÉTICA COM DADOS
ESPARSOS ATRAVÉS DE INTELIGÊNCIA ARTIFICIAL

Modelo canônico de trabalho monográfico
acadêmico em conformidade com as normas
ABNT.

Trabalho aprovado. Ribeirão Preto–SP, 21 de novembro de 2020:

Orientador:
Orientador

Professor
Convidado 1

Professor
Convidado 2

Ribeirão Preto–SP
2020

*Dedico este trabalho aos meus familiares e amigos que sempre me apoiaram
Em especial a minha mãe, que me apoiou e confiou em mim até o fim.
Também dedico a todos os divulgadores científicos
que nunca desistiram da ciência e sempre tentam fazer dela um bem de todos.*

Agradecimentos

Agradeço a minha família, minha namorada, amigos, colegas de laboratório, orientador, professores e a todos que me apoiaram e auxiliaram durante todo o mestrado. A vocês, o meu muito obrigado!

*“Um livro é a prova de que os seres humanos
são capazes de fazer magia.”
(Carl Sagan, Cosmos: A Personal Voyage, 1980)*

Resumo

A modalidade de imagens por ressonância magnética nuclear (magnetic resonance imaging - MRI) tem tomado um papel cada vez mais relevante na medicina moderna, apesar dos custos desse tipo de exame permanecerem altos quando comparados aos custos decrescentes de outros exames de imagens. Apesar dos esforços e investimentos em tecnologias, a redução dos custos desses exames ainda é modesta nos últimos anos. Esta proposta de pesquisa estudou a reconstrução de imagens por MRI com dados subamostrados no espaço de frequências (espaço K) através da inteligência artificial (IA), reduzindo o tempo de exame e consequentemente os custos. Este estudo comparou a viabilidade e eficiência desse método em diferentes porcentagens de sub-amostragem. Para isso, foram usados dados brutos de ressonância magnética, os quais foram subamostrados no espaço K, com isso, reduzindo o número de informações da imagem. Em seguida, através de técnicas de IA, como Deep Learning (DL), a imagem esparsa foi passada por um processo de reconstrução. Neste trabalho foi utilizada uma técnica de divisão da imagem em linhas e colunas separadas e trabalhadas como sequências únicas. Tais sequências foram usadas como entrada para uma rede neural e reconstruídas para formar a sequência original, antes da subamostragem. Por fim, foi usado um algoritmo para juntar tais sequências e formar as novas linhas e colunas da imagem reconstruída. Os resultados de tal processo foram comparados com a imagem original, analisando a eficiência da imagem obtida através de medidas quantitativas da qualidade das imagens reconstruídas, no caso, o índice de similaridade estrutural (structural similarity index measure - SSIM) e erro quadrático médio (mean squared error - MSE), para as diferentes ponderações de subamostragens no espaço K. Assim, com esse estudo, foi possível obter um método mais eficaz que o método convencional, reduzindo o tempo necessário para esse processo.

Palavras-chave: imageamento por ressonância magnética (MRI), inteligência artificial (AI), dados esparsos, espaço K, reconstrução de imagens.

Abstract

The medical image modality of magnetic resonance imaging (MRI) plays an increasingly important role in modern medicine, despite the high costs of these exams when compared to the decreasing costs of other imaging exams. Despite advances and investments in technologies, the attempt to take advantage of these studies in cost reduction is still modest in recent years. This research proposal aims to study and evaluate the reconstruction of MRI images with subsampled data in the frequency space (K space) through artificial intelligence (AI), with reduction of the examination time and consequently the exam costs. This study compares the feasibility and efficiency in different percentages of the sample. The raw data in K space was sampled in a controlled way to reconstruct the images, resulting in an image with sparse data reducing the number of image information. Then, the sparse data undergoes a reconstruction process through AI techniques such as Deep Learning (DL). The process resultss was compared to the original image, analyzing the image efficiency through reconstructed quantitative measures. In this work, the division of the image into separate rows and columns was used and worked as single sequences. These sequences were used as input to a neural network and reconstructed to form the original sequence, before subsampling. Finally, an algorithm was used to join these sequences and shapes to the new rows and columns of the reconstructed image. The results of this process were compared with the original image, analyzing the image efficiency obtained through quantitative measures of the quality of the reconstructed images, in this case, the structural similarity index (SSIM) and the mean square error (mean square error) - MSE), for the different subsampling weights in the K space. Therefore, with this study, it was possible to obtain a more effective method than the conventional method, reducing the time required for this process.

Keywords: magnetic resonance imaging (MRI), artificial intelligence (AI), sparse data, K-space, image reconstruction.

Lista de Figuras

Figura 1 – Espaço K e a imageamento por ressonância magnética (MRI) correspondente após a aplicação da transformada de fourier bidimensional. (MAZZOLA, 2009).	19
Figura 2 – Números Gerados pela AC-GAN. Fonte: (ODENA; OLAH; SHLENS, 2017)	25
Figura 3 – Cavalos sendo transformados em Zebras pela CycleGAN. Fonte: (ODENA; OLAH; SHLENS, 2017)	26
Figura 4 – Usos diversos da CycleGAN. Fonte: (ODENA; OLAH; SHLENS, 2017)	26
Figura 5 – Tradução de imagens de satélite para imagem de superfície via Pix2Pix. Fonte: (ISOLA et al., 2018)	27
Figura 6 – Tradução de desenho para prédio real usando a Pix2Pix. Fonte: (ISOLA et al., 2018)	28
Figura 7 – Imagem melhorada (esquerda) obtida a partir de uma imagem reconstruída (direita) tomando uma amostra esparsa do espaço K da imagem original. Fonte do autor.	36
Figura 8 – Representação da Imagem Original (completa) sendo sub-amostrada.	38
Figura 9 – Representação da subamostragem da imagem original (completa) se tornando a imagem sub-amostrada.	39
Figura 10 – Imagem sub-amostrada sendo separada em linhas.	40
Figura 11 – Linhas da imagem sub-amostrada sendo reconstruídas.	42
Figura 12 – Linhas reconstruídas formando a imagem reconstruída inicial (apenas linhas reconstruídas).	43
Figura 13 – Imagem com apenas as linhas reconstruídas sendo inseridas em uma imagem de tamanho igual a imagem original, evidenciando a necessidade de reconstruir mais dados.	44
Figura 14 – Imagem transposta da imagem reconstruída inicial (apenas linhas reconstruídas).	45
Figura 15 – Imagem transposta tendo suas linhas separadas.	46
Figura 16 – Linhas da imagem transposta sendo reconstruídas.	46
Figura 17 – Linhas reconstruídas da imagem transposta sendo transformada em imagem.	47
Figura 18 – Imagem transposta reconstruída gerando a imagem reconstruída final através da transposta inversa.	47
Figura 19 – Imagem original (completa) à esquerda, imagem sub-amostrada (entrada da rede neural) ao centro e imagem final (reconstruída) à direita.	48

Figura 20 – Erro quadrático médio da rede neural, onde Epoch representa a época atual, step a etapa desta época, time o tempo para executar tal etapa e mse o erro quadrático médio.	50
Figura 21 – Imagem com os resultados obtidos na segunda abordagem para todas as subamostragens realizadas.	53
Figura 22 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 10% (90% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.	54
Figura 23 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 50% (50% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.	54
Figura 24 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 50% (50% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.	56
Figura 25 – Três imagens adquiridas do dataset "NYU machine learning data". . .	57
Figura 26 – Três imagens adquiridas do dataset "Stanford Fullysampled 3D FSE Knees".	57
Figura 27 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 50% (50% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.	59

Lista de Tabelas

Tabela 1	– Resultados da similaridade estrutural média entre as imagens reconstruídas em diferentes porcentagens em relação a imagem original . . .	51
Tabela 2	– Resultados do erro quadrático médio para as diversas porcentagens de subamostragem	52
Tabela 3	– Resultados da similaridade estrutural média entre as imagens reconstruídas em diferentes porcentagens em relação a imagem original . . .	55
Tabela 4	– Resultados do erro quadrático médio para as diversas porcentagens de subamostragem	55
Tabela 5	– Dados da captura das imagens de cada um dos datasets utilizados . . .	57
Tabela 6	– Resultados da similaridade estrutural média entre as imagens reconstruídas em diferentes porcentagens em relação a imagem original . . .	58
Tabela 7	– Resultados do erro quadrático médio para as diversas porcentagens de subamostragem	58
Tabela 8	– Resultado da Média do SSIM para cada execução do algoritmo, com o dataset oficial do NYU machine learning data e Stanford Fullysampled 3D FSE Knees, tanto treinado quanto não treinado.	59

Sumário

1	INTRODUÇÃO	16
1.1	Princípios da Ressonância Magnética Nuclear	17
1.2	Imagens de Ressonância Magnética (MRI)	18
1.3	Espaço K	18
1.4	Sensoriamento Comprimido	19
1.5	Inteligência Artificial	20
1.6	Aprendizagem profunda (deep learning – dl)	20
1.7	Redes Generativas Adversarias	24
1.7.1	Utilizações e Casos de Uso	24
1.7.1.1	AC-GAN	25
1.7.1.2	CycleGAN	25
1.7.1.3	Pix2Pix	26
1.8	Índice de Similaridade Estrutural e Erro Quadrático Médio	28
1.9	Revisão Bibliográfica	29
2	METODOLOGIA	31
2.1	Imagens de ressonância magnética	31
2.2	Técnicas de inteligência artificial	31
2.3	Avaliação e análise dos resultados	31
2.4	Ambiente para testes	32
2.5	Tecnologias utilizadas	32
2.6	Aquisição das imagens	33
2.7	Trabalho com o espaço-k	33
2.8	Abordagem inicial	33
2.8.0.1	Redes GAN's	34
2.8.0.2	Redes de Super Resolução	34
2.8.0.3	Funcionamento padrão da SRGAN	34
2.8.1	Fluxo de uso da rede neural inicial	35
2.8.2	Dificuldades	36
2.9	Abordagem final	37
2.9.1	Preparação dos dados	37
2.9.2	Pré-processamento dos dados	38
2.10	Arquitetura da Rede	40
2.11	Fluxo de Reconstrução	41
2.12	Análise de qualidade da imagem	48

3	RESULTADOS	49
3.0.1	Validação com dataset secundário	55
3.0.2	Discussão dos Resultados	60
4	CONCLUSÃO	62
	Referências Bibliográficas	63

Introdução

As imagens por ressonância magnética nuclear (magnetic resonance imaging - MRI) é um método de imageamento médico com alto contraste em relação ao ruído permitindo o diagnóstico de forma não invasiva. A MRI tem se tornado uma ferramenta diagnóstica poderosa através de desenvolvimento de imagens de alta qualidade, graças aos avanços tecnológicos, oferecendo excelentes detalhes anatômicos devido ao seu alto contraste de tecidos moles e a possibilidade de aprimorar diferentes tipos de tecidos usando diferentes protocolos de aquisição, de forma que a MRI serve para a pesquisa e análise de doenças neurológicas, ortopédicas, abdominais, cervicais e cardíaca, podendo ser uma grande evidência para diagnosticar diversas acometimentos, como, esclerose múltipla, câncer, infartos, fraturas e até infecções. (CONSTANTINIDES, 2016). E, em muitos casos, prefere-se a MRI a Tomografia Computadorizada (TC) quando o médico precisa obter mais detalhes sobre os tecidos moles, por exemplo, para obter imagens de alterações no cérebro, na medula vertebral, nos músculos e no fígado. A MRI é particularmente útil para identificar tumores nesses tecidos (KOCAK, 2019).

Apesar disso, o paciente no momento da realização do exame, deve permanecer imóvel, suportando os ruídos fortes da máquina com protetores auriculares, e, às vezes, ainda precisa segurar a respiração em alguns momentos, enquanto ocorre a varredura para aquisição de uma MRI que pode levar cerca de 20 a 60 minutos (KOCAK, 2019). Essa experiência pode ser mais traumática para àqueles pacientes que são claustrofóbicos e/ou ansiosos.

Por outro lado, existem os altos os custos do exame de MRI que são impactados por vários fatores, como o próprio equipamento, um dos custos mais altos da medicina moderna, além dos custos de instalação, sala apropriada, blindagem magnética e manutenção. Algumas iniciativas foram empregadas em termos de reduções de custo do equipamento (FOO et al., 2018) (SANTINI; BIERI; DELIGIANNI, 2018), entretanto, estas iniciativas ainda não foram capazes de surtir significativo impacto ainda nos custos dos exames.

A outra maneira de reduzir os custos de exames é a redução de tempo de exame

sem prejuízo da qualidade das imagens. Considerando a demanda crescente pelo exame de MRI, em grande parte em virtude do desenvolvimento tecnológico em torno dos exames, pressiona-se a minimização dos custos dos exames. Há algumas iniciativas de se melhorar o workflow dos pacientes e otimizar o tempo por paciente (RECHT et al., 2019). Para além da otimização do workflow dos pacientes, a redução do tempo de aquisição seria crucial para a redução do tempo médio dos exames e, portanto, dos custos dos exames.

Desse modo, visando a melhor assistência à saúde do paciente, existe um grande desafio em adquirir a MRI em menor tempo e em menor custo. Para isso, a Inteligência Artificial (artificial intelligence – AI) e o aprendizado de máquina (machine learning – ML) estão sendo largamente aplicada com um número crescente de aplicações em imagens médicas e especificamente em MRI (CHOY et al., 2018)(GIGER, 2018)(YASAKA et al., 2018) . As aplicações vão desde o apoio computacional ao diagnóstico (computer aided diagnosis – CAD) ao apoio às neurocirurgias guiadas por imagens (SENDERS et al., 2018).

1.1 Princípios da Ressonância Magnética Nuclear

Na década de 40, surgiram os primeiros estudos em Ressonância Magnética Nuclear(RMN) (BLOCH; HANSEN; PACKARD, 1946).E, após quase 30 anos, na década de 70, Paul Lauterbour foi o responsável por obter as primeiras MRI (LAUTERBUR, 1973) e Peter Mansfield as primeiras MRI do corpo humano, o que lhes garantiram o prêmio Nobel de Medicina em 2003.

Para a Medicina Contemporânea, a RMN é uma tecnologia extremamente poderosa capaz de gerar imagens radiológicas, conhecida como MRI, que é um importante método de diagnóstico por imagem não-invasivo estabelecido na prática clínica, em virtude da sua alta capacidade de diferenciação de tecidos. O adjetivo "magnético" refere-se ao uso de campos magnéticos e "ressonância" refere-se à necessidade de combinar a frequência de um campo eletromagnético oscilante com a frequência de precessão do spin de alguns núcleos em um átomo de tecido. Esse átomo, na maioria dos casos, é o hidrogênio, o mais simples átomo da tabela periódica, possui apenas um próton - partícula positiva - como núcleo que movimenta-se girando, ou spin, em torno de seu próprio eixo, gerando um pequeno dipolo magnético. Além disso, é o átomo mais abundante no corpo humano cujas características ímpares do próton de hidrogênio ,como possuir um maior momento magnético do que os demais átomos do corpo, tornam a RNM mais sensível e mais precisa para a diferenciação de tecidos normais e patológicos (FOSTER et al., 1985).

1.2 Imagens de Ressonância Magnética (MRI)

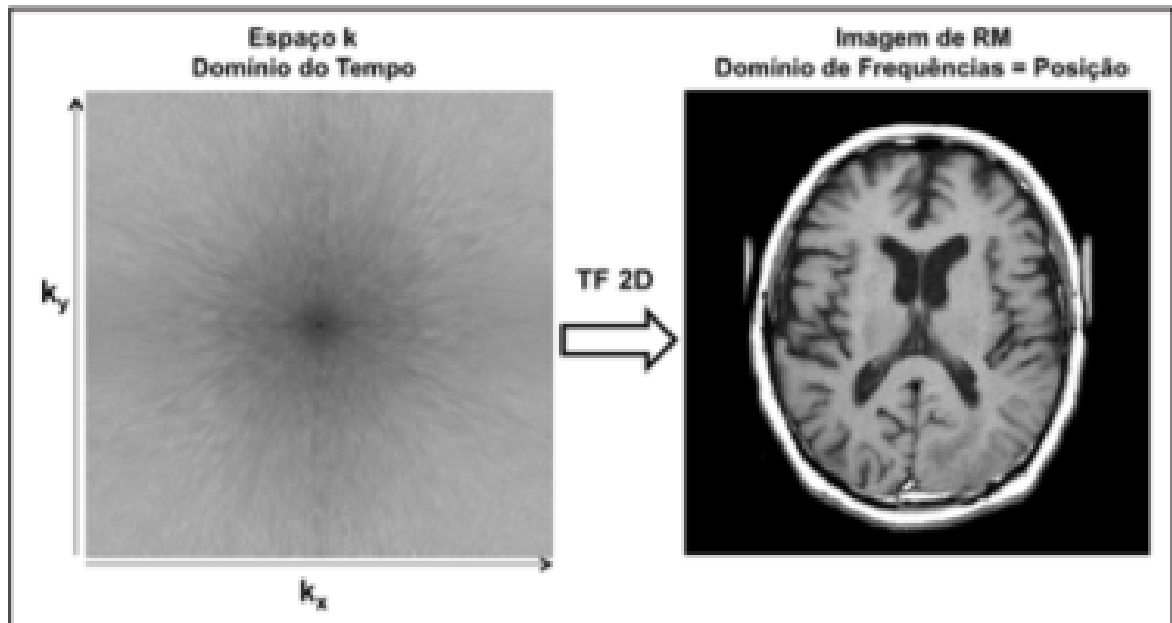
No momento do exame de ressonância magnética, o paciente é posicionado no interior da máquina magnética e fica sob ação do campo magnético, cujos os prótons de hidrogênios presentes no tecidos do paciente irão se orientar de acordo com a direção do campo aplicado, de forma que os prótons de hidrogênios apontam tanto paralelamente - nível de mais baixa energia- quanto antiparalelamente - nível de mais alta energia. Esse processamento para a obtenção da MRI depende da codificação espacial do sinal através do uso de gradientes de campo magnético. O ciclo de sequência de pulso deve ser repetido até que o espaço K seja preenchido de maneira adequada para formar a imagem final.

Após a codificação de frequência e fase mediante o uso de gradientes de campos magnéticos, um sinal contínuo emitido pelo tecido humano a ser estudado é recebido pela bobina receptora e amostrado pelo equipamento, tornando-se um sinal discreto. O sinal amostrado é convertido em tensão e organizado no espaço K , que contém os dados brutos da imagem. Como o sinal é codificado em fase e frequência, ambos são as coordenadas do espaço K (Figura 1). A transformada inversa de Fourier da magnitude do sinal do espaço K é a imagem de ressonância magnética usual, portanto, uma boa cobertura do espaço K é necessária para garantir uma qualidade de imagem razoável. A transformada de Fourier, de forma simplificada, é uma teoria que diz que qualquer imagem ou sinal de frequência pode ser expresso como uma soma de sinusóides (senos e cossenos). É bem utilizada na ressonância magnética para transformar imagens reais (espaço natural) em imagens no espaço- k e vice-versa.

1.3 Espaço K

O espaço K é um conceito abstrato que auxilia no entendimento de sequências de pulso modernas e metodologias de aquisição, que permite a visualização do espaço K como uma matriz de tons de cinza, cujas linhas são preenchidas com um sinal da RMN. Cada ponto nesta matriz corresponde a uma intensidade de sinal (tom de cinza) e a uma posição no tempo, e representa a amplitude do sinal recebido pela bobina naquele dado instante. Os eixos de coordenadas (x e y ou k_y e k_x) deste espaço são, respectivamente, o gradiente de codificação de frequência e o gradiente de codificação de fase (MAZZOLA, 2009), como mostra a Figura 1. Ou seja, o espaço- k é o espaço de frequência que é obtido nas máquinas de ressonância magnética e que através da transformada de Fourier podem se tornar imagens reais. À vista disso, podemos entender o espaço- k como o espaço de frequência.

Figura 1 – Espaço K e a imageamento por ressonância magnética (MRI) correspondente após a aplicação da transformada de Fourier bidimensional. (MAZZOLA, 2009).



Apesar de algumas iniciativas importantes na redução do tempo do exame com os procedimentos de aquisição convencionais e otimizados, recentemente abriu-se a possibilidade de aquisições subamostradas no espaço de frequências ou espaço K (K-space) e a reconstrução das imagens através de AI. Nesse processo de reconstrução, as informações adquiridas no espaço K não são reconstruídas através de algoritmos de transformada rápida de Fourier (fast Fourier transform -FFT) inversa para chegar no espaço natural das imagens, mas sim através de algoritmos de AI especificamente desenhados e configurados para esta aplicação. Nesse contexto, podemos ter algoritmos capazes de lidar com o espaço K subamostrado, e sem a necessidade de amostrar todo o espaço K, podemos reduzir o tempo de aquisição (HYUN et al., 2018) (QUAN; NGUYEN-DUC; JEONG, 2018) (YANG et al., 2017) (ZHU et al., 2018).

1.4 Sensoriamento Comprimido

O esquema de reconstrução subamostrado descrito tem sido chamado na literatura de “imageamento de ressonância magnética por sensoriamento comprimido” (compressed sensing magnetic resonance imaging – CS-MRI) (SHRIVIDYA; BHARATHI, 2016) (SANDILYA; NIRMALA, 2017). A técnica de subamostragem baseia-se em pegar um conjunto completo de dados e retirar uma parte (amostra desse conjunto). Desse jeito, teremos como resultado final (sobra), apenas uma parte do conjunto original, ou seja, uma su-

bamostra. Quando nos referimos ao espaço- k , o nosso espaço de frequência, ao executar uma subamostragem no mesmo, os dados remanescentes serão considerados dados esparsos. Em resumo, dados esparsos são os dados subamostrados.

1.5 Inteligência Artificial

De acordo com Barr e Feigenbaum (1981), "a Inteligência Artificial (Artificial Intelligence - AI) é a parte da ciência da computação preocupada com o design de sistemas de computadores inteligentes", isto é, sistemas que exibem as características associadas à inteligência no comportamento humano - compreensão, linguagem, aprendizado, raciocínio, solução de problemas e assim por diante.

Um dos ramos importantes da IA é a Rede Neural Artificial (RNAs). RNAs mimetizam as redes neurais biológicas. Os neurônios recebem sinais por meio de dendritos, processam os pulsos elétricos recebidos no corpo celular e transmitem sinais através do axônio. De maneira análoga, o modelo elétrico de um neurônio biológico típico consiste em um ativador linear, seguido por uma função inibidora não-linear. A função de ativação produz a soma da excitação de entrada ponderada, enquanto a função de inibição não linear tenta capturar os níveis de sinal da soma. O sinal resultante, produzido pelo neurônio elétrico, é assim limitado (amplitude limitada) (KONAR, 2018). Uma RNA é uma coleção desses neurônios elétricos conectados em diferentes topologias. A aplicação mais comum de uma RNA é no aprendizado de máquina (MELLIT; KALOGIROU, 2008).

1.6 Aprendizagem profunda (deep learning – dl)

A tecnologia de aprendizado de máquina capacita muitos aspectos da sociedade moderna: de buscas na web a filtragem de conteúdo em redes sociais, as recomendações em sites de comércio eletrônico, e está cada vez mais presente em produtos de consumo como câmeras e smartphones. Os sistemas de aprendizado de máquina são usados para identificar objetos em imagens, transcrever fala em texto, combinar notícias, publicações ou produtos com interesses de usuários e selecionar resultados relevantes da pesquisa. Cada vez mais, essas aplicações fazem uso de uma classe de técnicas chamada aprendizagem profunda (deep learning – DL) (LECUN; BENGIO; HINTON, 2015).

As técnicas convencionais de aprendizado de máquina eram limitadas em sua capacidade de processar dados naturais em sua forma bruta. Durante décadas, a construção de um sistema de reconhecimento de padrões ou de aprendizado de máquina exigiu enge-

nharia cuidadosa e considerável especialização de domínio para projetar um extrator de recursos que transformasse os dados brutos, por exemplo, como os valores de pixel de uma imagem em uma representação interna adequada ou vetor de recursos. A partir do qual o subsistema de aprendizagem, muitas vezes, um classificador, pode detectar ou classificar padrões na entrada.

O aprendizado de representação é um conjunto de métodos que permite que uma máquina seja alimentada com dados brutos e descubra automaticamente as representações necessárias para detecção ou classificação. Com a composição de tais transformações, funções muito complexas podem ser aprendidas. Para tarefas de classificação, camadas mais altas de representação amplificam aspectos da entrada que são importantes para a discriminação e suprimem variações irrelevantes. Uma imagem, por exemplo, vem na forma de uma matriz de valores de pixel, e os recursos aprendidos na primeira camada de representação normalmente representam a presença ou ausência de bordas em orientações e locais específicos na imagem. A segunda camada, geralmente, detecta motivos, detectando arranjos particulares de bordas, independentemente de pequenas variações nas posições das bordas. A terceira camada pode reunir motivos em combinações maiores que correspondem a partes de objetos familiares, e camadas subsequentes detectariam objetos nas combinações dessas partes. O aspecto principal da DL é que essas camadas de recursos não são projetadas por engenheiros humanos: elas são aprendidas a partir de dados usando um procedimento de aprendizado para fins gerais.

A DL está fazendo grandes avanços na resolução de problemas que resistiram às melhores tentativas da comunidade de inteligência artificial por muitos anos. Acabou sendo muito bom descobrir estruturas intrincadas em dados de alta dimensão e, portanto, é aplicável a muitos domínios da ciência, negócios e governo. Além de bater recordes em reconhecimento de imagem e reconhecimento de fala, ele superou outras técnicas de aprendizado de máquina ao prever a atividade de moléculas de drogas potenciais, analisando dados de aceleradores de partículas, reconstruindo circuitos cerebrais e prevendo os efeitos de mutações na expressão gênica não-codificadora e doença. Talvez mais surpreendentemente, o aprendizado profundo produziu resultados extremamente promissores para várias tarefas no entendimento da linguagem natural, particularmente a classificação de tópicos, a análise de sentimentos, o questionamento e a tradução de idiomas.

Acredita-se que o DL terá muitos mais sucessos no futuro próximo, pois exige muito pouca engenharia à mão, de modo que pode facilmente aproveitar o aumento da quantidade de dados e cálculos disponíveis. Novos algoritmos de aprendizado e arquiteturas que estão sendo desenvolvidos para redes neurais profundas só irão acelerar esse progresso.

A forma mais comum de aprendizagem de máquina, profunda ou não, é a aprendizagem supervisionada. Imagine que queremos construir um sistema que possa classificar

imagens contendo, digamos, uma casa, um carro, uma pessoa ou um animal de estimação. Primeiramente, coletamos um grande conjunto de imagens de casas, carros, pessoas e animais de estimação, cada um rotulado com sua categoria. Durante o treinamento, a máquina exibe uma imagem e produz uma saída na forma de um vetor de pontuações, uma para cada categoria. Queremos que a categoria desejada tenha a pontuação mais alta de todas as categorias, mas é improvável que isso aconteça antes do treinamento. Calculamos uma função objetivo que mede o erro (ou distância) entre as pontuações de saída e o padrão desejado de pontuações. A máquina então modifica seus parâmetros ajustáveis internos para reduzir esse erro. Esses parâmetros ajustáveis, geralmente chamados de pesos, são números reais que podem ser vistos como 'botões' que definem a função de entrada-saída da máquina. Em um sistema típico de aprendizagem profunda, pode haver centenas de milhões desses pesos ajustáveis e centenas de milhões de exemplos rotulados para treinar a máquina.

Para ajustar adequadamente o vetor de peso, o algoritmo de aprendizado calcula um vetor de gradiente que, para cada peso, indica em qual quantidade o erro aumentaria ou diminuiria se o peso fosse aumentado em uma pequena quantidade. O vetor de peso é então ajustado na direção oposta ao vetor gradiente. A função objetivo calculada sobre todos os exemplos de treinamento, pode ser vista como uma espécie de paisagem montanhosa no espaço de alta dimensão dos valores de peso. O vetor gradiente negativo indica a direção da descida mais íngreme nessa paisagem, aproximando-a ao mínimo, onde o erro de saída é baixo, em média.

Na prática, a maioria dos praticantes usa um procedimento denominado gradiente descendente estocástico (stochastic descent gradient - SDG). Isso consiste em mostrar o vetor de entrada para alguns exemplos, computando as saídas e os erros, calculando o gradiente médio para esses exemplos e ajustando os pesos de acordo. O processo é repetido para muitos pequenos conjuntos de exemplos do conjunto de treinamento até que a média da função custo determine as paradas. É chamado de estocástico porque cada pequeno conjunto de exemplos fornece uma estimativa ruidosa do gradiente médio em todos os exemplos. Esse procedimento simples geralmente encontra um bom conjunto de pesos surpreendentemente rápidos quando comparado a técnicas de otimização muito mais elaboradas. Após o treinamento, o desempenho do sistema é medido em um conjunto diferente de exemplos chamado conjunto de testes. Isso serve para testar a capacidade de generalização da máquina - sua capacidade de produzir respostas sensatas sobre novos insumos que nunca viu durante o treinamento.

Muitas das aplicações práticas atuais de aprendizado de máquina usam classificadores lineares em cima de recursos de engenharia manual. Um classificador linear de duas classes calcula uma soma ponderada dos componentes do vetor de recursos. Se a soma ponderada estiver acima do limite, a entrada será classificada como uma categoria

específica.

Desde a década de 1960, sabemos que os classificadores lineares só podem gravar seu espaço de entrada em regiões muito simples, a saber, meio-espacos separados por um hiperplano. Mas problemas como reconhecimento de imagem e fala requerem que a função de entrada-saída seja insensível a variações irrelevantes da entrada, como variações na posição, orientação ou iluminação de um objeto, ou variações no tom ou no acento da fala, enquanto é muito sensível a variações específicas, por exemplo, a diferença entre um lobo branco e uma raça de cão branco semelhante ao lobo chamado Samoieda. No nível dos pixels, as imagens de dois samoiedos em poses diferentes e em ambientes diferentes podem ser muito diferentes umas das outras, enquanto duas imagens de samoieda e um lobo na mesma posição e em fundos semelhantes podem ser semelhantes entre si. Um classificador linear ou qualquer outro classificador superficial que opera em pixels brutos não poderia distinguir os dois últimos ao colocar os dois primeiros na mesma categoria. É por isso que os classificadores superficiais exigem um bom extrator de características que resolva o dilema de seletividade-invariância - aquele que produz representações que são seletivas aos aspectos da imagem que são importantes para a discriminação, mas que são invariantes a aspectos irrelevantes como a pose do animal. Para tornar os classificadores mais poderosos, é possível usar funções não lineares genéricas, como os métodos do kernel, mas recursos genéricos, como os que surgem com o kernel gaussiano, não permitem generalizar bem longe dos exemplos de treinamento. A opção convencional é entregar extractores de bom design, o que requer uma quantidade considerável de habilidades de engenharia e conhecimento de domínio. Mas tudo isso pode ser evitado se bons recursos puderem ser aprendidos automaticamente usando um procedimento de aprendizado de propósito geral. Essa é a principal vantagem do DL.

Uma arquitetura de DL é uma pilha multicamada de módulos simples, todos ou a maioria deles, sujeitos a aprendizado, e muitos dos quais calculam mapeamentos de entrada-saída não-lineares. Cada módulo na pilha transforma sua entrada para aumentar a seletividade e a invariância da representação. Com camadas múltiplas não-lineares, digamos uma profundidade de 5 a 20, o sistema pode implementar funções extremamente complexas de suas entradas que são simultaneamente sensíveis a detalhes minuciosos - distinguindo Samoiedas de lobos brancos - e insensíveis a grandes variações irrelevantes como o fundo, pose, iluminação e objetos ao redor.

Deste modo, o DL pode ser usado para a reconstrução de imagens com o espaço K subamostrado, i.e., CS-MRI, tendo como proposta aqui a avaliação de eficácia da DL em diferentes situações de CS-MRI (HYUN et al., 2018) (YANG et al., 2017) (MEHRTASH et al., 2017).

1.7 Redes Generativas Adversarias

As Redes Generativas Adversarias (GANS) são uma modalidade de Deep Learning. Elas permitem uma representação profunda dos dados sem necessitar de uma grande quantidade de treinamento. Para isso, elas utilizam de uma rede geradora, para gerar imagens, e uma rede discriminadora, que busca identificar se aquela imagem é falsa ou real (CRESWELL et al., 2018).

Em seu texto Ian Goodfellow afirma que as Redes GANS estão sendo o modelo mais bem sucedido de redes generativas, sendo utilizadas numa gama bem ampla de problemas e encontrando as soluções na maior parte delas, ou pelo menos, tendo um resultado melhor que outras técnicas mais antigas (GOODFELLOW et al., 2020).

Em resumo, as GANS podem ser entendidas como uma rede formada por duas partes, uma delas é a Rede Geradora, onde tem como foco entender os principais conceitos e artifícios da imagem e gerar uma nova. Sua outra parte é a Rede Discriminadora, que a todo momento irá verificar se a imagem de entrada é falsa ou real.

A Rede Discriminadora irá sempre ser treinada usando as imagens existentes no dataset original, dessa forma, ela consegue captar os artifícios, detalhes e especificidades que fazem daquele conjunto de dados algo real e válido.

Quando uma imagem é gerada pela Rede Geradora, a Rede Discriminadora irá avaliar se aquela imagem é falsa ou real. Neste momento, essa arquitetura faz jus ao nome "Adversárias", visto que a Discriminadora irá classificar, inicialmente, como falsa a imagem gerada e a Rede Geradora irá tentar entender os motivos de ser falsa e como ela pode melhorar a imagem, até chegar no momento em que ela consegue enganar a Rede Discriminadora.

Neste momento, as imagens geradas serão tão fidedignas a original que nem uma rede treinada para classificar imagens como reais ou falsas consegue saber a verdade. Portanto, os resultados obtidos dessas redes tendem a ter um alto grau de confiabilidade e fazem da GAN a melhor arquitetura para geração de imagens e dados (GOODFELLOW et al., 2020).

1.7.1 Utilizações e Casos de Uso

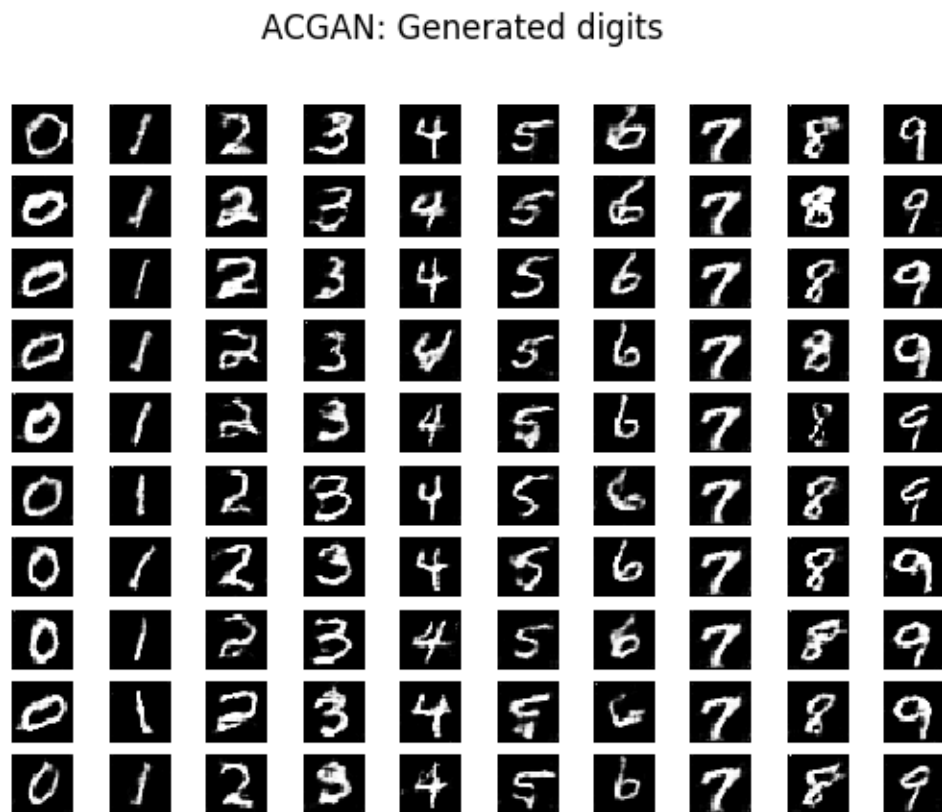
Temos diversas implementações e utilizações das GANs no mundo real, variando desde geração e identificação gráfica (por exemplo, gerar dígitos numéricos), como a "AC-GAN" (ODENA; OLAH; SHLENS, 2017), alterar domínios de imagens (por exemplo, transformar uma foto em uma pintura), como a "CycleGAN" (ZHU et al., 2020) ou até mesmo traduzir uma imagem para um outro contexto (por exemplo, imagem real é transformada numa

imagem de satélite), como a "Pix2Pix" (ISOLA et al., 2018).

1.7.1.1 AC-GAN

Utiliza de um classificador auxiliar para identificar e compreender os números e conseguir gerar imagens, a partir do zero, que se assemelhem com os dígitos numéricos.

Figura 2 – Números Gerados pela AC-GAN. Fonte: (ODENA; OLAH; SHLENS, 2017)



1.7.1.2 CycleGAN

Em resumo, recebe dois ou mais domínios distintos de imagens, como Cavalos e Zebras e altera uma imagem de um domínio para pertencer a outro. Neste caso, uma Zebra sendo transformada em um Cavalo ou vice-versa.

Figura 3 – Cavalos sendo transformados em Zebras pela CycleGAN. Fonte: (ODENA; OLAH; SHLENS, 2017)

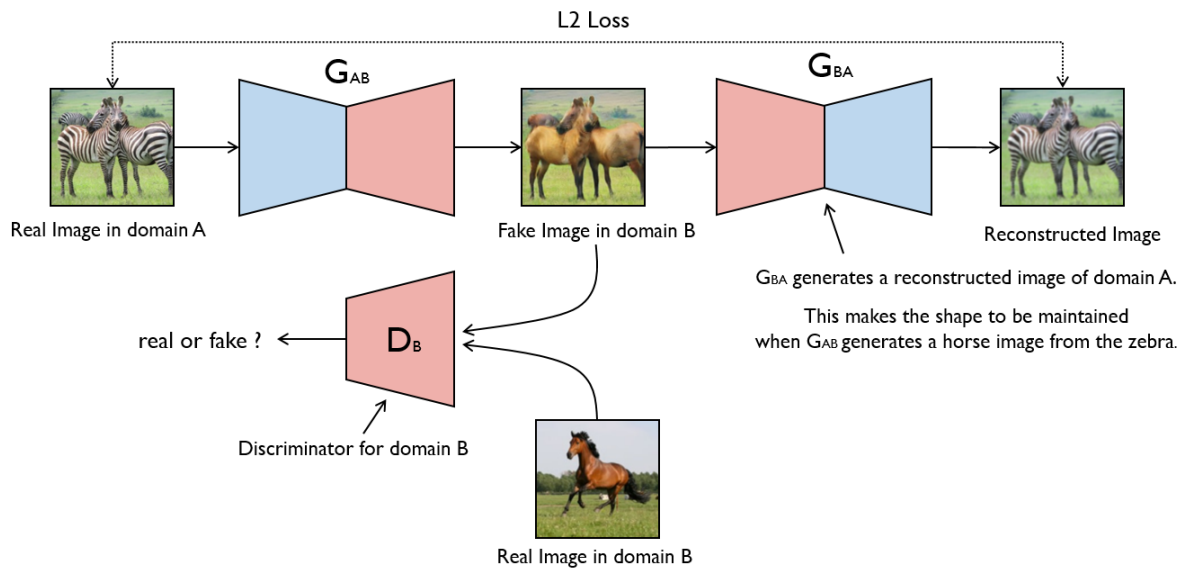
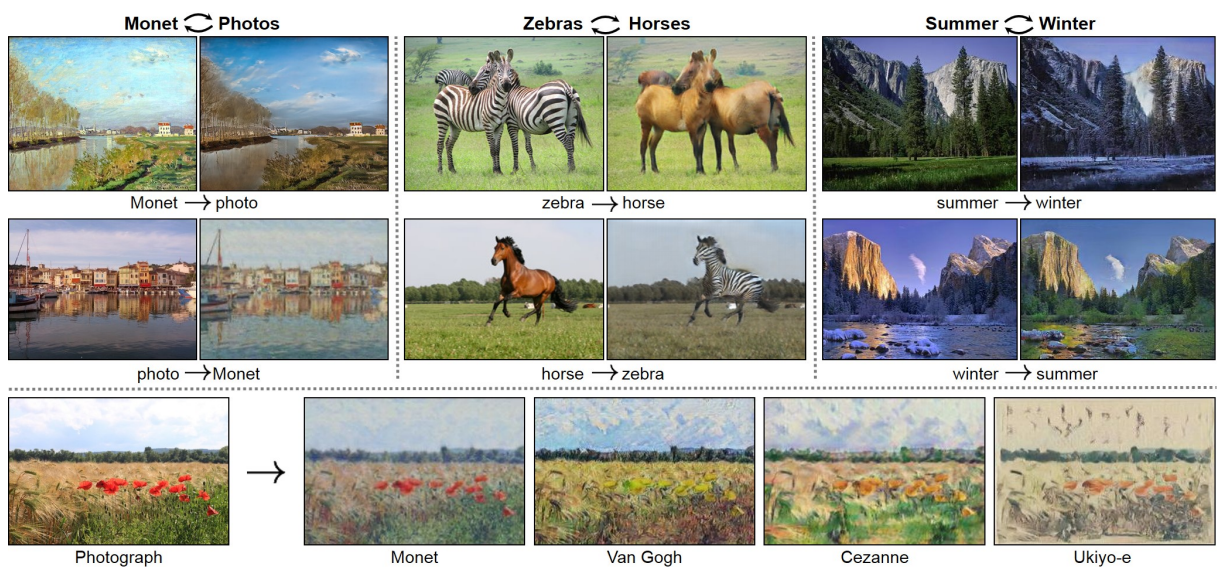


Figura 4 – Usos diversos da CycleGAN. Fonte: (ODENA; OLAH; SHLENS, 2017)



1.7.1.3 Pix2Pix

Consiste em fazer uma tradução direta entre imagens, afim de pegar os principais atributos e característica de um dataset base e traduzi-lo condicionalmente a um outro dataset (conjunto destino). A principal diferença para o CycleGAN é que sempre será do conjunto base ao conjunto destino.

Por exemplo, um desenho pode ser transformado numa imagem de prédio real, pegando condições como: Caso a cor do retângulo seja verde, será uma sacada do prédio, se for azul claro, será uma porta, se for azul escuro, uma janela.

Figura 5 – Tradução de imagens de satélite para imagem de superfície via Pix2Pix. Fonte: (ISOLA et al., 2018)

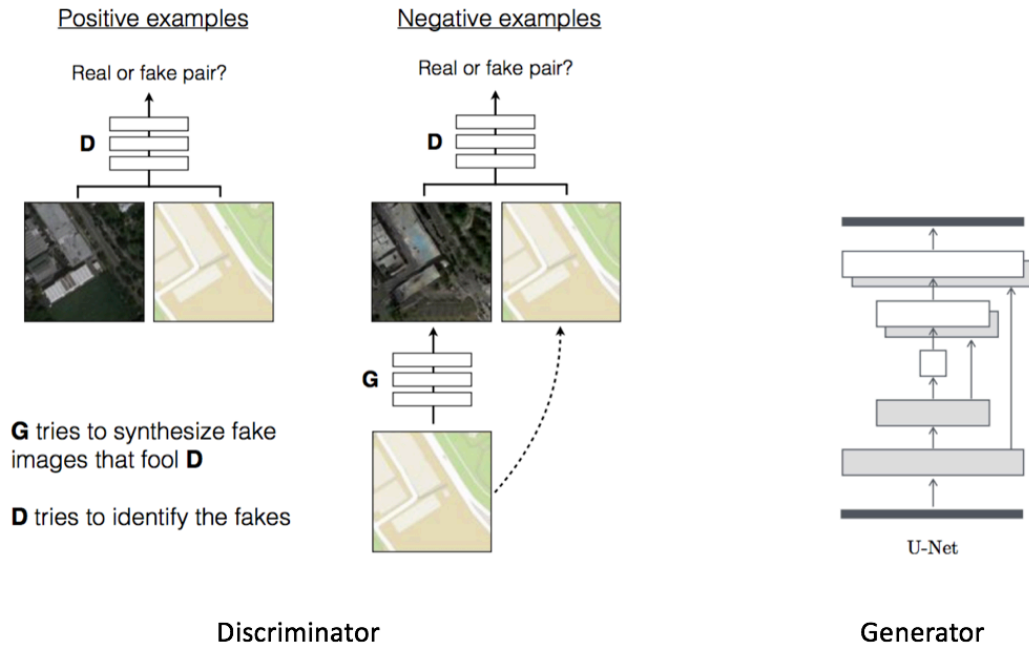
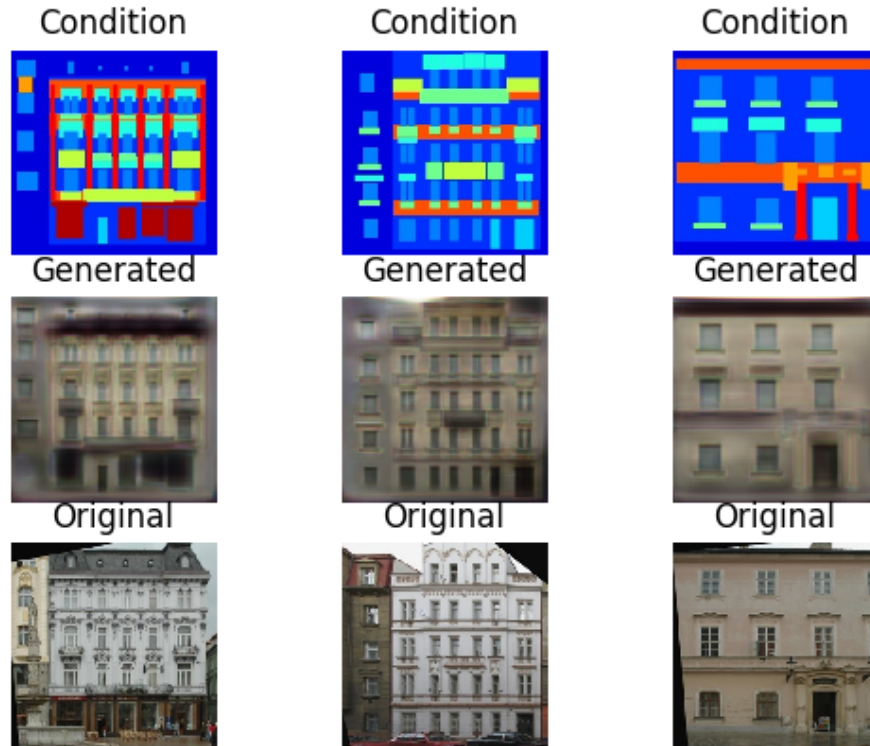


Figura 6 – Tradução de desenho para prédio real usando a Pix2Pix. Fonte: (ISOLA et al., 2018)



1.8 Índice de Similaridade Estrutural e Erro Quadrático Médio

Em síntese, através do uso de Deep Learning pode-se restaurar o espaço-k para que fique completamente amostrado com subsequente reconstrução da imagem do espaço frequência para imagem plana (espaço natural) . Para tanto, a validação consiste na análise da qualidade deste processo deve-se comparar a imagem reconstruída com a original através do Índice de Similaridade Estrutural (structural similarity index measure – SSIM) e Erro Quadrático Médio (mean squared error – MSE).

O cálculo do SSIM pode ser feito através da seguinte fórmula:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

onde:

- μ_x : a média de x ;
 μ_y : a média de y ;
 σ_x^2 : a variância de x ;
 σ_y^2 : a variância de y ;
 σ_{xy} : a covariância de x e y ;
 c_1 : $(k_1 L)^2$;
 c_2 : $(k_2 L)^2$;
 L : $2 \text{ bits por pixel} - 1$;
 k_1 : 0.01, por padrão;
 k_2 : 0.03, por padrão.

Já o cálculo do MSE pode ser feito através da seguinte fórmula

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

onde:

- Y : valores observáveis (reais) da variável sendo prevista;
 \hat{Y} : valores previstos pelo algoritmo;

O mais fácil e mais usado, segundo Wang et al. (2004), é o MSE, uma vez que esse é apenas a representação do erro quadrático acumulado entre duas imagens, no caso, a reconstruída e a original. Ou seja, a diferença quadrada entre cada píxel da imagem dividido pelo número de Pixels. Neste mesmo artigo, Wang et al. (2004) apresenta um modelo mais poderoso de comparação de imagens, o SSIM, onde neste algoritmo o que é levado em conta são as estruturas visíveis na imagem, ou seja, ao invés de apenas percorrer pixel a pixel e calcular a diferença, o que acontece é a visualização da imagem como um todo, com a interpretação da estrutura visível da imagem reconstruída sendo comparada com a imagem original real, sabendo assim, o quão fiel a reconstrução está, visivelmente.

1.9 Revisão Bibliográfica

Apesar dos estudos de Hyun et al. (2018) e Souza e Frayne (2019) fazerem uso de algoritmos e métodos de AI recentes para o sensoriamento comprimido, falta ainda na literatura estudos sistemáticos da eficácia do método em diferentes porcentagens e contextos.

Por conseguinte, a presente proposta de pesquisa se coloca nesta lacuna, avaliando a reconstrução por sensoriamento comprimido via uma arquitetura atualmente bem-sucedida de AI, i.e., a aprendizagem profunda (deep learning - DL), cuja eficiência é

medida e avaliada através do erro médio quadrático (mean squared error – MSE) e medidas do índice de similaridade estrutural (structural similarity index measure – SSIM) entre a imagem reconstruída e uma imagem de referência reconstruída, visando a redução no tempo de aquisição do imageamento da ressonância magnética e, portanto, melhor custo benefício ao paciente.

Sendo assim, utilizamos diferentes porcentagens de subamostragem, indo de 10% e chegando a 50%, avaliando a eficácia da reconstrução nesses casos.

Também buscamos a maneira mais simplificada de se trabalhar, visto que as técnicas existentes possuem um grande custo de processamento, devido à complexidade do algoritmo.

Metodologia

2.1 Imagens de ressonância magnética

As imagens de ressonância magnética foram recebidas na forma de dados brutos, que são tradicionalmente processados com a transformada de Fourier inversa, para ter-se a partir os dados no espaço-k, foram subamostrados de forma esparsa através da subamostragem Cartesiana, reduzindo o tamanho e quantidade de informações do arquivo de dados brutos.

2.2 Técnicas de inteligência artificial

Para a reconstrução das imagens foram utilizadas as redes deep learning descritas anteriormente. Na camada de entrada teremos os dados brutos já esparsos e nas camadas seguintes foram buscados padrões de agrupamento de dados, e posteriormente uma tentativa de reconstrução da imagem, que finalmente é avaliada através da comparação com a imagem atual, utilizando algoritmos como o de interpolação linear, para metrificar a porcentagem da acurácia adquirida pela imagem final em relação a imagem reconstruída pelo método convencional.

2.3 Avaliação e análise dos resultados

Os resultados de tal processo foram comparados com a imagem original, analisando a eficiência da imagem obtida através de mediadas quantitativas da qualidade das imagens reconstruídos, ou seja, o índice de similaridade estrutural (structural similarity index measure - SSIM) e erro quadrático médio (mean squared error - MSE).

As avaliações de eficácia da reconstrução CS-MRI com AI e DL foi feita comparando a imagem reconstruída com a imagem de referência, ou seja, a imagem reconstruída

pelo método proposto foi comparada com a imagem original reconstruída com o espaço-k completo, utilizando as métricas SSIM e MSE.

As reconstruções CS-MRI foram feitas amostrando o espaço-k em diferentes porções, no caso, de 10% a 50% do espaço-k, com um intervalo de 10%.

2.4 Ambiente para testes

No objetivo de garantir que os resultados estejam preparados para uma análise fiel de dados, sem intervenções de casualidades ou interferência de processos externos, o ambiente para testes foi preparado para projetar um isolamento total do código e garantir que todos os recursos do sistema estejam destinados apenas ao trabalho em questão, nada além disso.

Para isso, também foi utilizada apenas uma máquina para todos os testes, com garantia de que todo o sistema estava igual durante todos os testes. As especificações de tal computador são:

- CPU: AMD Ryzen 9 3900X
- GPU: NVIDIA GeForce RTX 2080 TI
- RAM: 64GB DDR4 3466MHz
- SSD: SSD Intel 660p, 1TB, M.2 NVMe

2.5 Tecnologias utilizadas

Para a elaboração da solução final o Python foi utilizado para a parte do código que envolve inteligência artificial e no processo de manipulação das imagens de ressonância magnética.

Dentre os frameworks e pacotes utilizados no Python destacam-se o TensorFlow, para a construção das redes neurais, o ISMRMRD-Python para trabalhar com as imagens de ressonância e o NumPy para trabalhar com dados numéricos.

Também foi elaborado um *scraper web*, um software coletador de dados proveniente de sites, para a automação da extração das imagens de ressonância magnética no dataset original, onde este foi feito também com Python, através do framework Scrapy.

2.6 Aquisição das imagens

De primeiro momento foram validados os provedores de imagens de ressonância magnética para identificar qual a melhor fonte para aquisição das imagens. Após análise, foi escolhido o dataset “NYU machine learning data” provido pela New York University, através do site MRIDATA.ORG.

Em seguida foi elaborado o algoritmo para trabalhar com o formato de imagem proveniente, recuperando os dados tanto no espaço-k quanto a imagem já reconstruída (com o espaço-k completo).

Para os testes iniciais foram adquiridas um total de 125 imagens, todas do mesmo dataset e de imagens de ressonâncias magnéticas de joelhos.

Já para o teste final foram colhidas 1000 imagens, sendo feita uma distribuição de 70% para treino, 20% para validação e 10% para testes.

2.7 Trabalho com o espaço-k

Afim de garantir um bom esparsamento dos dados, de forma a tornar a rede neural útil para casos reais, foi levantado um estudo sobre qual a melhor forma de se esparsar os dados do espaço-k e a conclusão chegada foi utilizar a Subamostragem Cartesiana para imagens de ressonância magnética, que se baseia em dividir o espaço-k em espaçamentos internos equidistantes, gerando no final um novo subespaço-k.

Todo o processo com o espaço-k foi feito através da uma imagem real provida do dataset escolhido (NYU machine learning data) com amostragem completa dos dados.

2.8 Abordagem inicial

Tendo a finalidade de apenas comprovar a ideia central do trabalho, esta etapa visou buscar na literatura modelos existentes de redes neurais que possam gerar um resultado rápido que, através do espaço-k esparsado, consiga reconstruir uma imagem de forma que se assimile a original (com espaço-k completo).

Para isso, foi escolhida a rede SRGAN (LEDIG et al., 2017), que mistura Redes Gerativas Adversariais, (GAN’s, Generative Adversarial Networks) com redes de Super Resolução (SR). Desta forma, ao mesmo tempo em que a rede tenta melhorar a qualidade de uma imagem, ela faz isso de tal forma que o resultado gerado seja similar a original, ou

seja, a rede aprende qual a forma correta de tratar as ‘falhas’ da imagem gerada através dos dados esparsos.

Por padrão, a rede é treinada para avaliar o erro através da função do erro quadrático médio, dando uma taxa de assertividade maior do que utilizando índice de similaridade estrutural.

2.8.0.1 Redes GAN’s

Como dito anteriormente, as redes GAN’s são redes que funcionam através de duas redes, sendo uma delas a Geradora, que gera as imagens e Discriminadora, que verifica se a imagem gerada por ser considerada válida ou não.

Tendo em vista que a rede Discriminadora aprende através das imagens reais, ela tem uma taxa de aprendizado e acerto maior do que a Geradora no início, o que faz necessário que a rede Geradora tenha um aprendizado muito bom para ser capaz de ‘enganar’ a rede Discriminadora e fazer a resposta ser de que a imagem é válida.

Sendo assim, os dados em falta da subamostragem são gerados de uma forma bem realista.

2.8.0.2 Redes de Super Resolução

As redes de super resolução funcionam, principalmente, através de álgebra aritmética e têm como objetivo usar como entrada uma imagem de baixa resolução e através do processamento da rede gerar uma imagem de alta resolução referente àquela entrada.

Para atingir tais objetivos, os mecanismos mais usados são de *upscaling* (aumento na escala da imagem), *interpolation* (interpolações na imagem) e *zooming* (realização de zooms na imagem). Neste caso, é empregada a metodologia de *upscaling*.

Outro ponto pela escolha desta metodologia é o fato de ser a que mais se aproxima do inverso da subamostragem, sendo assim, podemos dizer que estamos fazendo o oposto da subamostragem inicial, voltando para como era antes do processo.

Então, as imagens que entram nesta etapa da rede terão sua qualidade aprimorada.

2.8.0.3 Funcionamento padrão da SRGAN

Por padrão, as redes SRGAN são feitas para melhorar a qualidade de imagens em RGB no espaço real, isto é, imagens convencionais, como JPG e PNG, e não no espaço frequência, como o espaço-k.

Como entrada essa rede recebe uma imagem em baixa resolução, distorcida, com ruídos ou com *blur* (borrada) e gera como saída uma imagem em alta resolução sem nenhuma das deformidades encontradas na entrada.

Em primeiro momento a arquitetura da GAN provê uma geração de imagens o mais parecida da realidade possível, porém, ainda pode apresentar falhar e necessidade de aprimoramento e neste momento entra a rede de super resolução, fazendo um *upscale* na imagem e então passando para o discriminador.

2.8.1 Fluxo de uso da rede neural inicial

Em primeiro momento, os dados são extraídos do dataset obtido pelo MRIDATA, salvando o espaço-k original em arquivo *NPY* (array de multi-dimensão do numpy, um pacote Python para trabalhar com dados numéricos) e reconstruindo a imagem, salvando-a em um JPG.

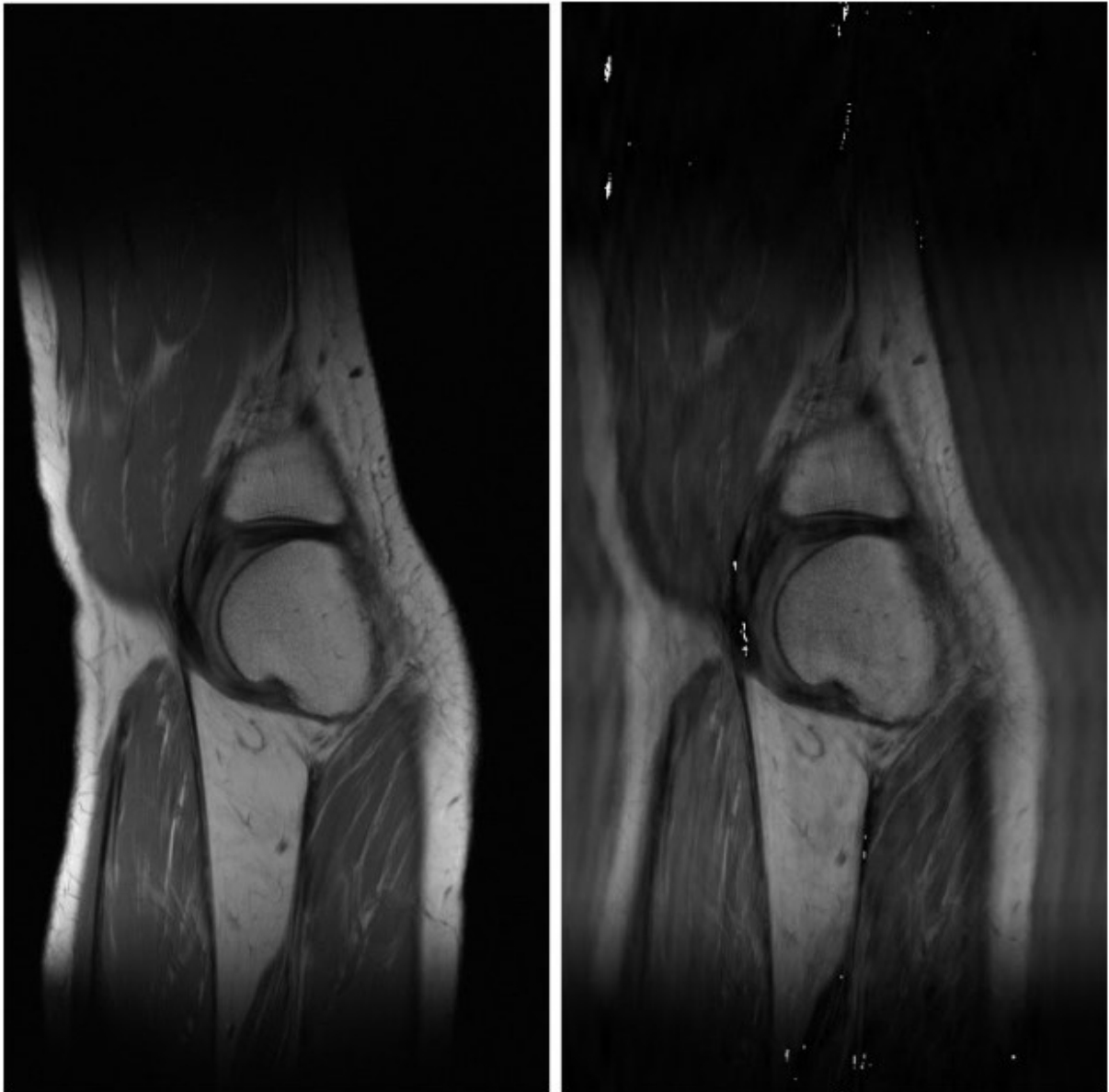
Em seguida os dados passam pelo processo de subamostragem, onde o espaço-k 100% se torna um espaço-k de 80%, logo após a finalização do procedimento, os dados também são salvos da mesma forma que os originais (completos).

Tendo preparado os dados, chega a hora de enviá-los para a rede neural. Ambas as versões (esparsas e completas) possuem o mesmo nome, porém, uma fica na pasta '*fullysampled*' e outra na pasta '*subsampled*' e desta forma a rede consegue identificar a relação entre as imagens.

O '*input*' (entrada) da rede é a imagem sub-amostrada e como saída temos uma imagem gerada que tenta se assimilar ao máximo com a imagem completa correspondente, para isso, a rede faz uso da sua parte GAN, comparando o quão parecida são as imagens e verificando se poderia ou não ser considerada como uma imagem válida (gerada, porém similar a real). Já para a melhoria da qualidade, consertando as falhas provenientes da subamostragem, a rede faz uso da sua parte de Super Resolução.

Ao final do treinamento temos que a rede recebe apenas a imagem subamostrada e automaticamente gera uma imagem a qual ela tenta ao máximo fazer parecer com a que seria a versão completa original. Veja a Figura 7 para mais detalhes, onde na esquerda temos a imagem melhorada pela rede e na direita temos a imagem esparsa.

Figura 7 – Imagem melhorada (esquerda) obtida a partir de uma imagem reconstruída (direita) tomando uma amostra esparsa do espaço K da imagem original. Fonte do autor.



2.8.2 Dificuldades

Nesta metodologia avaliativa o principal problema enfrentado foi a necessidade de adequação da rede para se trabalhar com dados no espaço-k, tendo que ser alterado desde o formato da imagem utilizada, de um RGB colorida convencional para uma imagem em tons de cinza à adição de operadores de números complexos, devido ao valor do pixel em espaço frequência estar no formato de número complexo, tendo uma parte imaginária.

Outro problema enfrentado foi o tempo de processamento e treinamento da rede, o qual apresentou um tempo de total de 18 horas e 24 minutos, mesmo com apenas 100

imagens, por consequência, há a necessidade de um maquinário muito potente, ou seja, o custo para adquiri-lo seria caro.

Por fim, a depuração de rede para melhorias de performance se mostrou bem complexa, devido ao uso de duas redes bem poderosas e complexas juntas, no caso, as redes de Super Resolução e as Redes GAN's, que são altamente *black-box* (redes caixa-preta, praticamente impossível de entender as decisões tomadas durante o aprendizado).

Com isso, a abordagem final do trabalho teve de ser uma nova, mais focada e específica para o trabalho proposto.

2.9 Abordagem final

Considerando-se os principais problemas e dificuldades existentes na metodologia inicial, a primeira etapa da abordagem final do trabalho foi o estudo de motivos que causariam tais dificuldades e levantamento de possibilidades de melhoria.

Para o tempo de processamento e necessidade de recursos de alto desempenho foi pensado como reduzir a complexidade da rede, mais especificamente, as ligações e camadas ocultas.

Ao mesmo tempo, foi pensado em como tornar mais fácil a análise individual e parte a parte da rede, para entendermos o que ela está fazendo e onde estão os problemas.

Por fim, foi necessário garantir que todo o processo fosse preparado para aceitar os dados brutos no espaço-k sem ter problema com o formato dos dados.

2.9.1 Preparação dos dados

Para a fase de preparação dos dados foi utilizado o mesmo algoritmo da abordagem inicial, visto que ele se comprovou ser o que mais se encaixava para o problema.

Em resumo, o *scraper* percorre o site do MRIDATA.ORG e realiza o download dos datasets e em seguida, usando o pacote *ISMRRD-Python* são extraídos os dados dos dataset, sendo eles, o espaço-k completo de cada imagem.

Com o espaço-k completo de cada imagem já salvo, é realizada a reconstrução através da Transformada Inversa de Fourier para termos a imagem base de comparação, isto é, a imagem original, reconstruída pelo métodos convencionais.

Por fim, através de técnicas de subamostragem usando o pacote *NumPy* o espaço-k é esparsado através da subamostragem cartesiana em 10%, 20%, 30%, 40% e 50%. Tal subamostragem é salva em formato de array do *NumPy*, no caso, no formato *.NPY*.

Feito isso, o processo de preparação dos dados é encerrado e inicia-se a fase de pré-processamento.

2.9.2 Pré-processamento dos dados

A Figura 8 mostra o processo de subamostragem, onde os dados são retirados da imagem original. Já a A Figura 9 demonstra como essa imagem com dados esparsos é transformada em uma outra imagem, chamada de *imagem sub-amostrada*, a qual, embora sub-amostrada, possui a mesma proporção.

Com isso, simulamos uma imagem sub-amostrada real, visto que se fosse adquirida em uma máquina real de RMN, o resultado não iria possuir espaços em branco ou vazios.

Para fins lúdicos e de exemplificação, como as imagens de ressonância magnéticas são todas em tons de cinza, porém complexas de se analisar em um pequeno tamanho, os exemplos utilizados nas figuras abaixo são imagens simplificadas em tons de cinza. Nestas imagens, cada pixel é representado por um quadrado de uma cor na escala de cinza, dessa forma, a cada quadrado podemos perceber uma troca de coluna ou linha da imagem.

Figura 8 – Representação da Imagem Original (completa) sendo sub-amostrada.

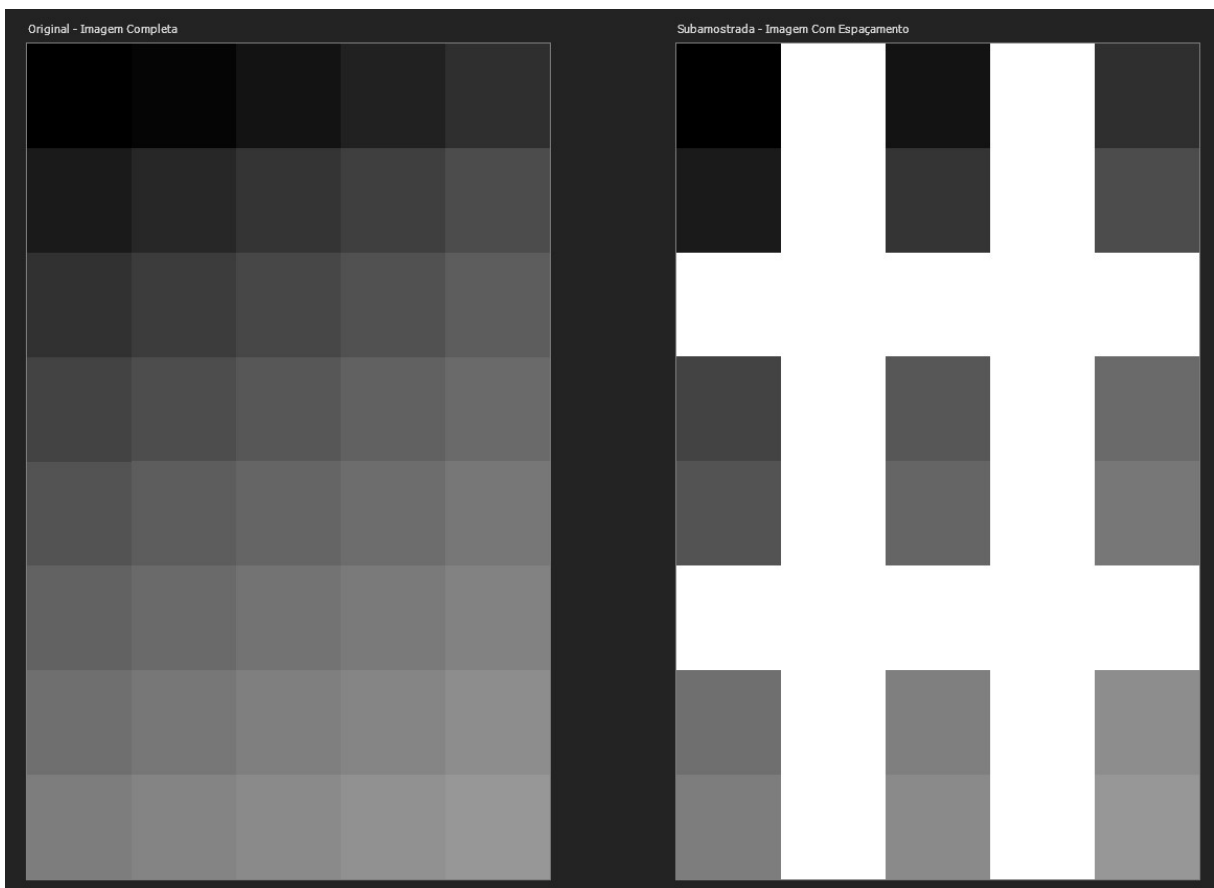
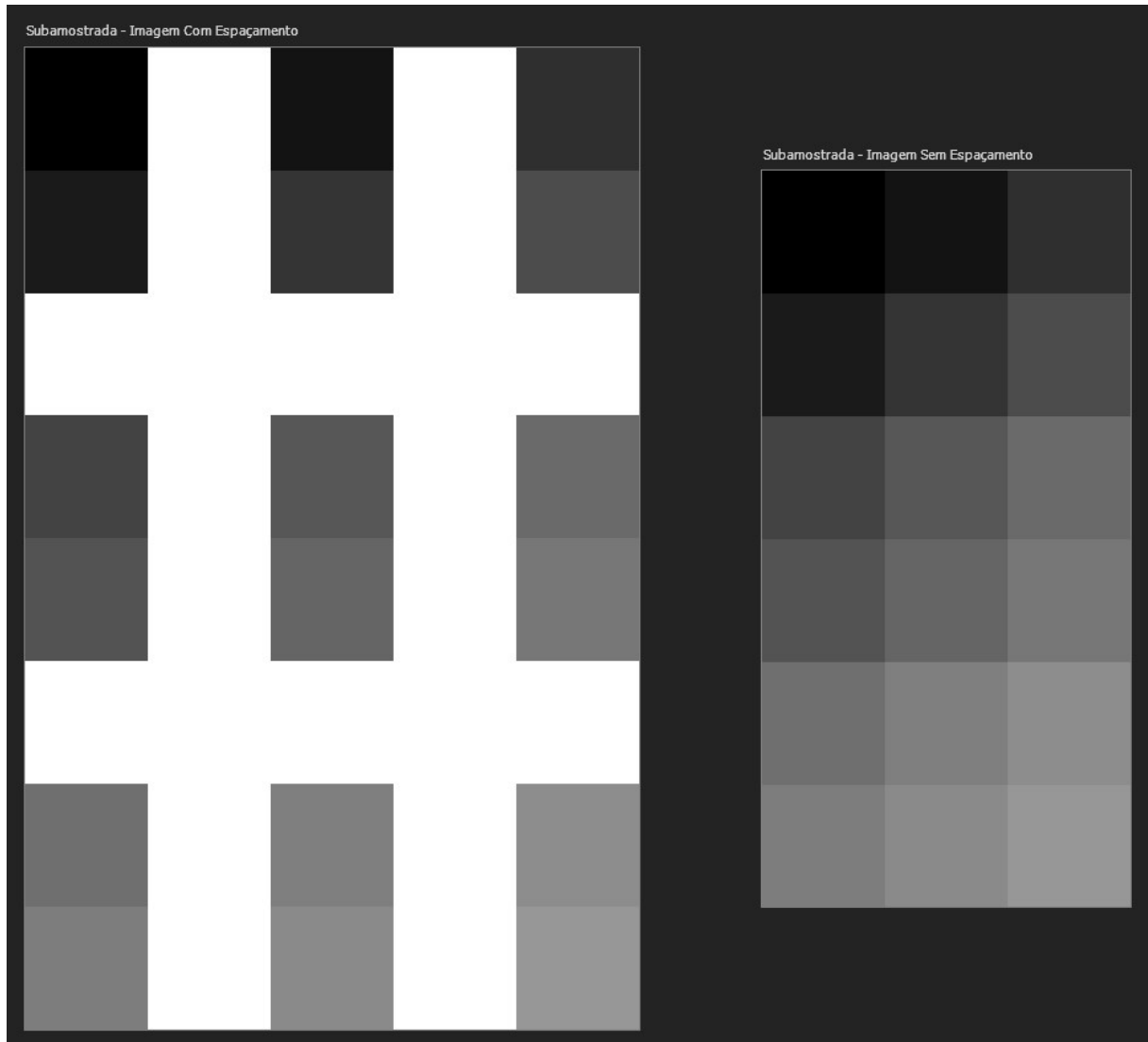


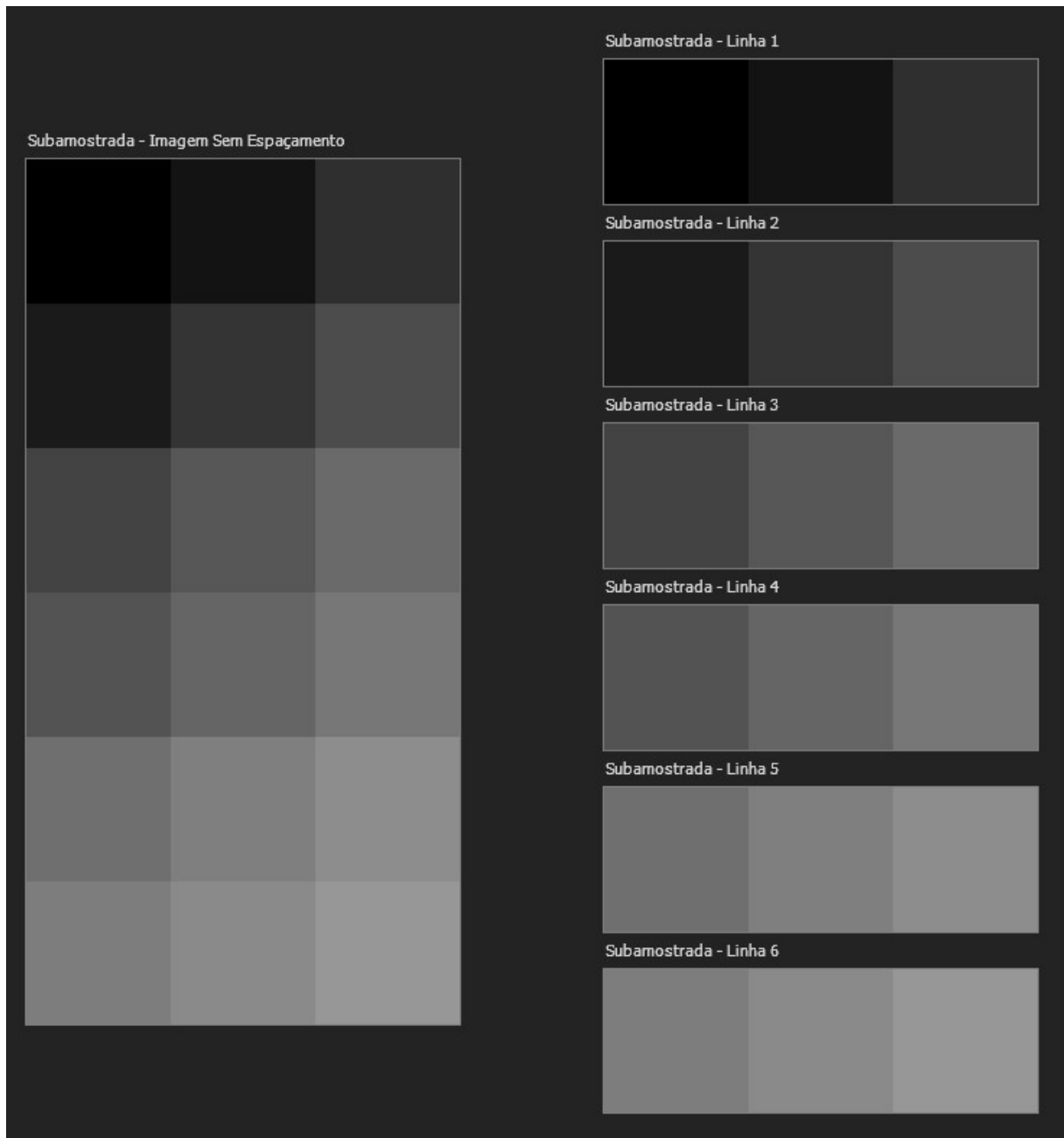
Figura 9 – Representação da subamostragem da imagem original (completa) se tornando a imagem sub-amostrada.



Tendo a imagem sub-amostrada pronta, a imagem segue para a etapa de separação das linhas, isto é, cada linha da imagem é considerado um vetor independente, conforme mostrado na Figura 10, porém, todas as linhas são salvas como pertencente a tal imagem sub-amostrada e possui seus índices originais mantidos.

Sendo assim, o algoritmo sabe de qual imagem sub-amostrada cada vetor pertence e qual sua posição.

Figura 10 – Imagem sub-amostrada sendo separada em linhas.



Com isso, os dados estão preparados para entrarem na rede neural reconstruí-los.

2.10 Arquitetura da Rede

O modelo da rede consiste em treiná-la para reconstruir linha a linha, sendo assim, trabalhando com dados de uma só dimensão, o que reduz tanto tempo quanto a complexidade da rede.

Desse jeito, a arquitetura consiste em duas camadas de Convolução de uma di-

mensão (Conv1D), uma vez que essas pegam semelhanças e características nos vizinhos próximos, o que queremos para saber qual o valor correto para completar os dados em falta através dos dados existentes, seguidas de duas camadas Dense, visto que essas buscam a característica no todo, ou seja, considerando todo o vetor que chega nesta camada. Ao término de ambas, temos uma camada de Dropout, visto que essas removem as características encontradas em excesso. Por fim, são repetidas as camadas de Conv1D e Dense.

Tendo em mente que o MSE, assim como abordado anteriormente, se dá pelo cálculo acumulado da diferença entre cada pixel da imagem, usaremos o MSE como método de cálculo de erro, onde o que será avaliado é o erro entre cada posição do vetor, entre o vetor reconstruído e o vetor original.

Dessa forma, a entrada da rede pós treinada sempre será um vetor único, representando uma linha da imagem sub-amostrada e sua saída será também um vetor único, representando a reconstrução da linha usada como entrada.

2.11 Fluxo de Reconstrução

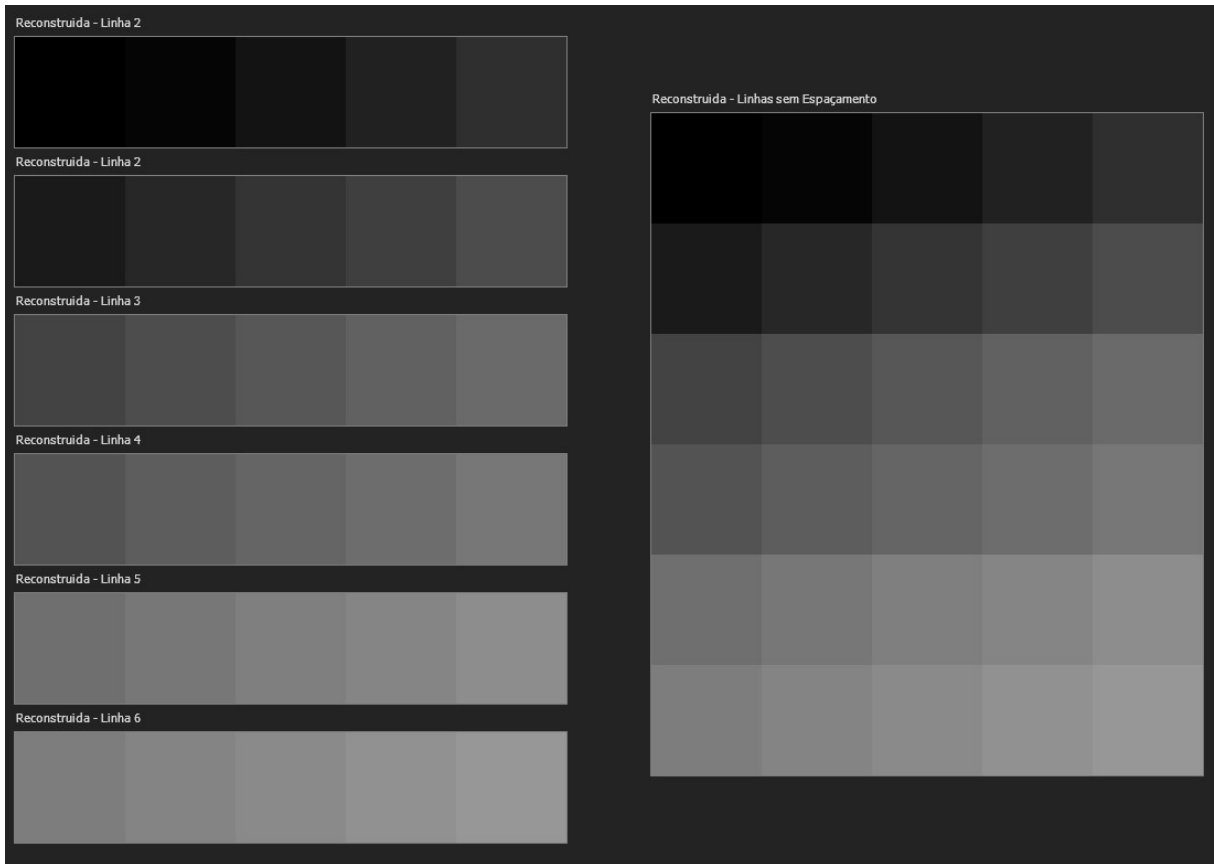
Uma vez que todos os dados estão preparados, isto é, a imagem sub-amostrada está com todas suas linhas separadas, inicia-se o fluxo de reconstrução dela, iniciando por enviar para a rede neural esses dados, onde o algoritmo seleciona linha a linha (vetor por vetor) e usá-lo como entrada, gerando uma saída para cada uma, conforme mostrado na Figura 11.

Figura 11 – Linhas da imagem sub-amostrada sendo reconstruídas.



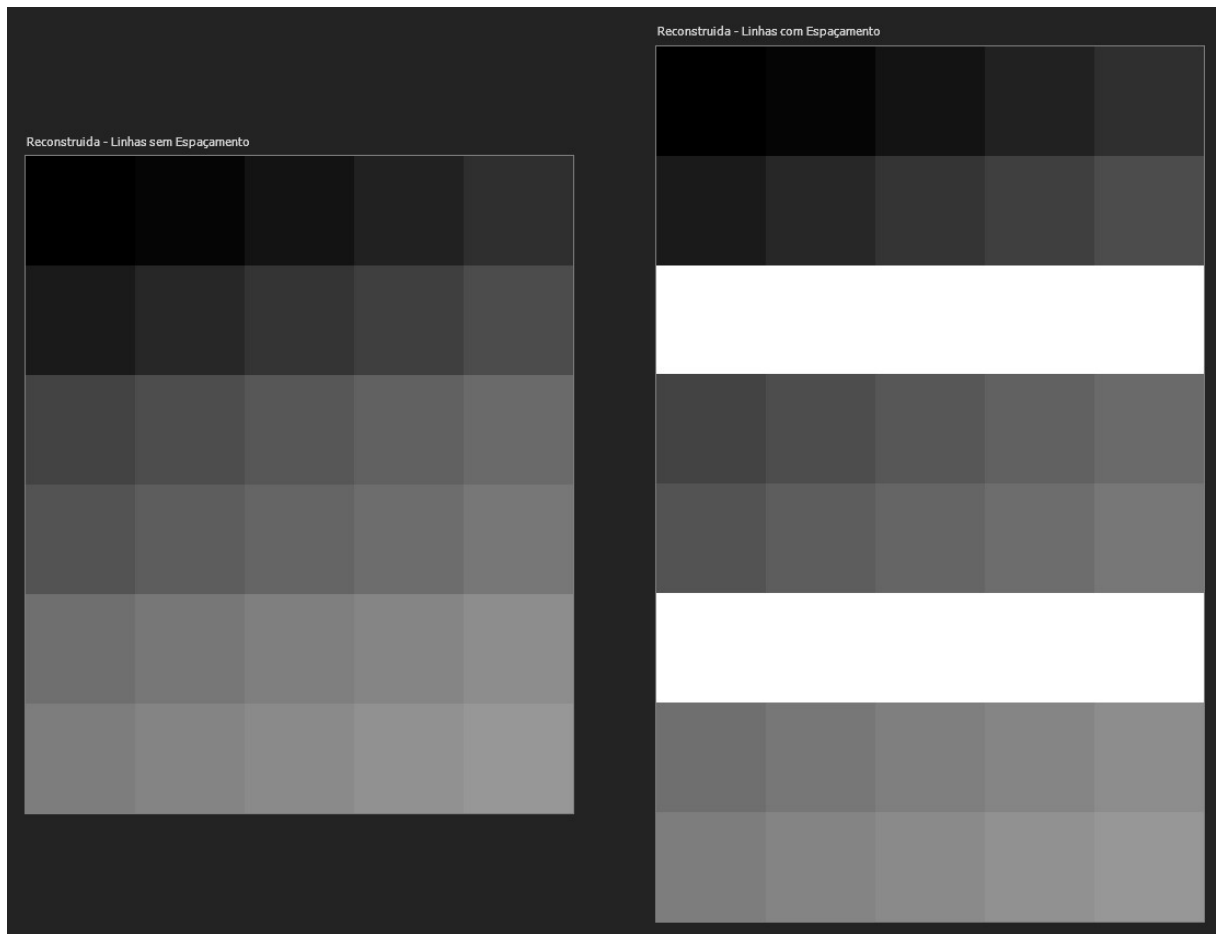
Uma vez que todas as linhas sejam reconstruídas, é gerada uma nova imagem com elas, sendo a primeira imagem reconstruída, onde as linhas estão reconstruídas, porém, as colunas ainda não, conforme a Figura 12.

Figura 12 – Linhas reconstruídas formando a imagem reconstruída inicial (apenas linhas reconstruídas).



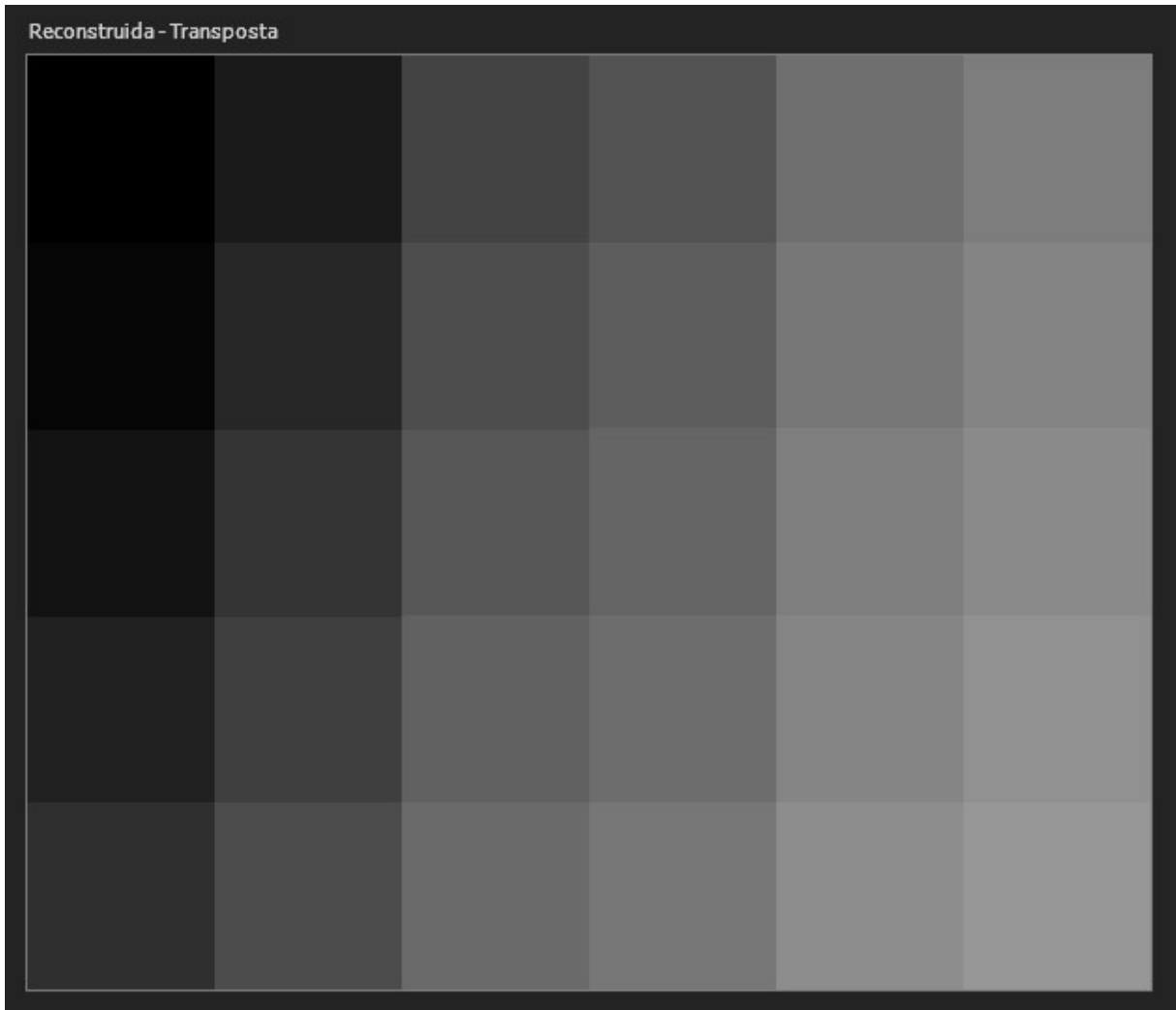
Se nos atentarmos, veremos que a imagem possui o mesmo número de colunas da imagem original, pois o número de dados (pixel) em cada linha já está igual ao original, pois foi totalmente reconstruído. Porém, as linhas ainda estão em menor número. Para ficar mais claro, caso tentarmos criar uma nova imagem de tamanho igual a original e encaixarmos as linhas que temos atualmente, teremos uma imagem, neste caso, similar a Figura 13.

Figura 13 – Imagem com apenas as linhas reconstruídas sendo inseridas em uma imagem de tamanho igual a imagem original, evidenciando a necessidade de reconstruir mais dados.



Para resolver isso, trabalhando a imagem como uma matriz de duas dimensões, ou seja, eixo X (largura) e eixo Y (altura), vamos calcular a transposta dessa imagem, gerando uma imagem, neste exemplo, similar a da Figura 14.

Figura 14 – Imagem transposta da imagem reconstruída inicial (apenas linhas reconstruídas).



Este passo é feito desta forma para garantir uma automação e padronização melhor do código, onde o *script* responsável por separar e refazer a imagem trabalha com os índices de cada linha (vetor) que existe na imagem, independente do seu tamanho ou proporção.

Vale ressaltar que esse processo é bem rápido, sendo de complexidade $O(n)$, então, perto da complexidade e tempo total da abordagem, ele não impacta de modo negativo.

Portanto, foi escolhido manter esse processo ao invés de quebrar a padronização de estrutura do código e criar um *script* intermediário que faça o mesmo processo, só que usando diretamente colunas.

Agora, iremos novamente separar cada linha da imagem, como mostrado na Figura 15 e em seguida enviar cada linha desta nova imagem para a rede neural, pois como realizamos a transposta, as linhas viraram colunas e as colunas viraram linhas, sendo assim, estaremos trabalhando com as colunas da imagem subamostrada originalmente.

Quando estas forem reconstruídas teremos, para este exemplo o que é representado pela Figura 16

Figura 15 – Imagem transposta tendo suas linhas separadas.

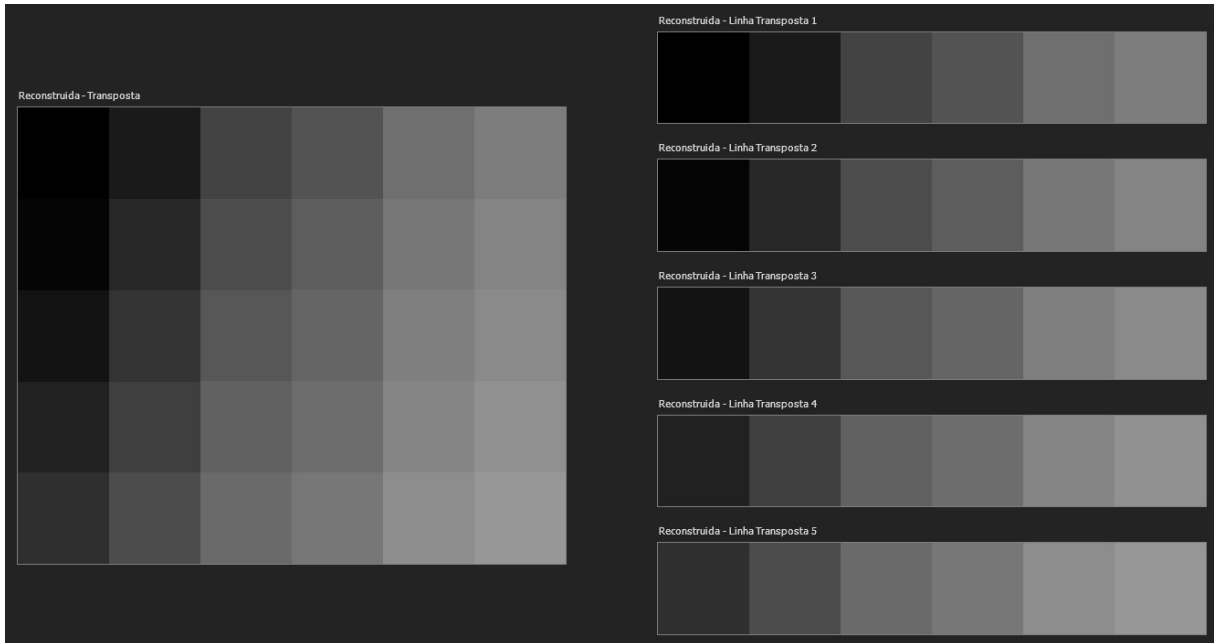
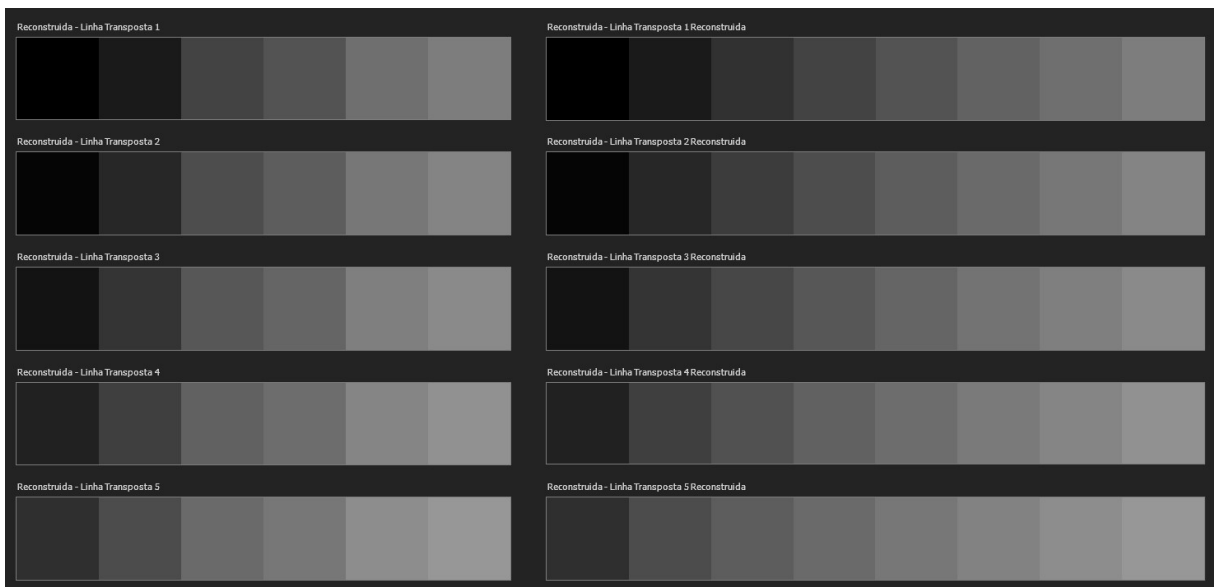
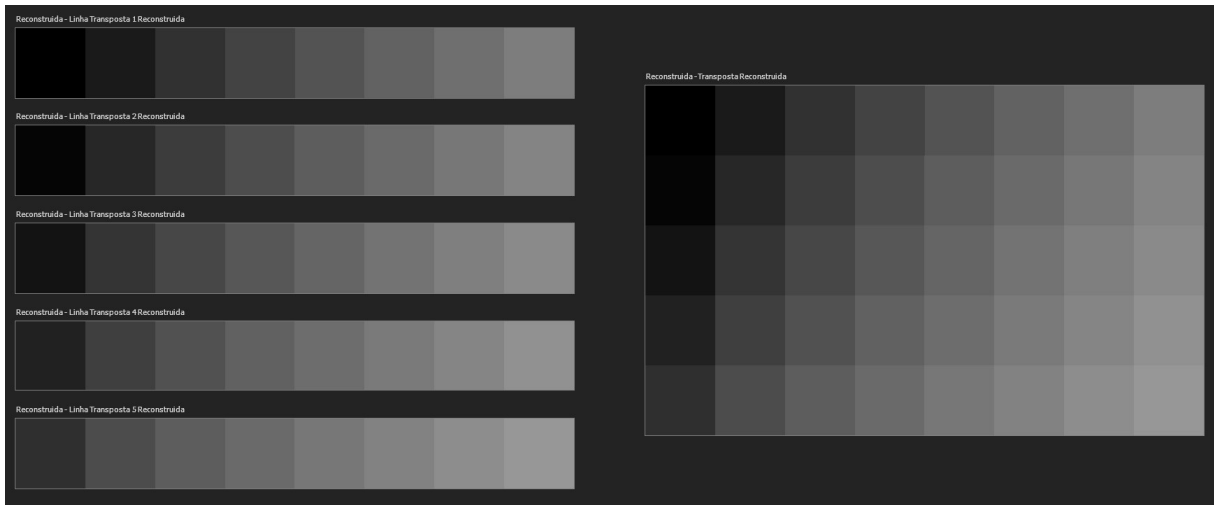


Figura 16 – Linhas da imagem transposta sendo reconstruídas.



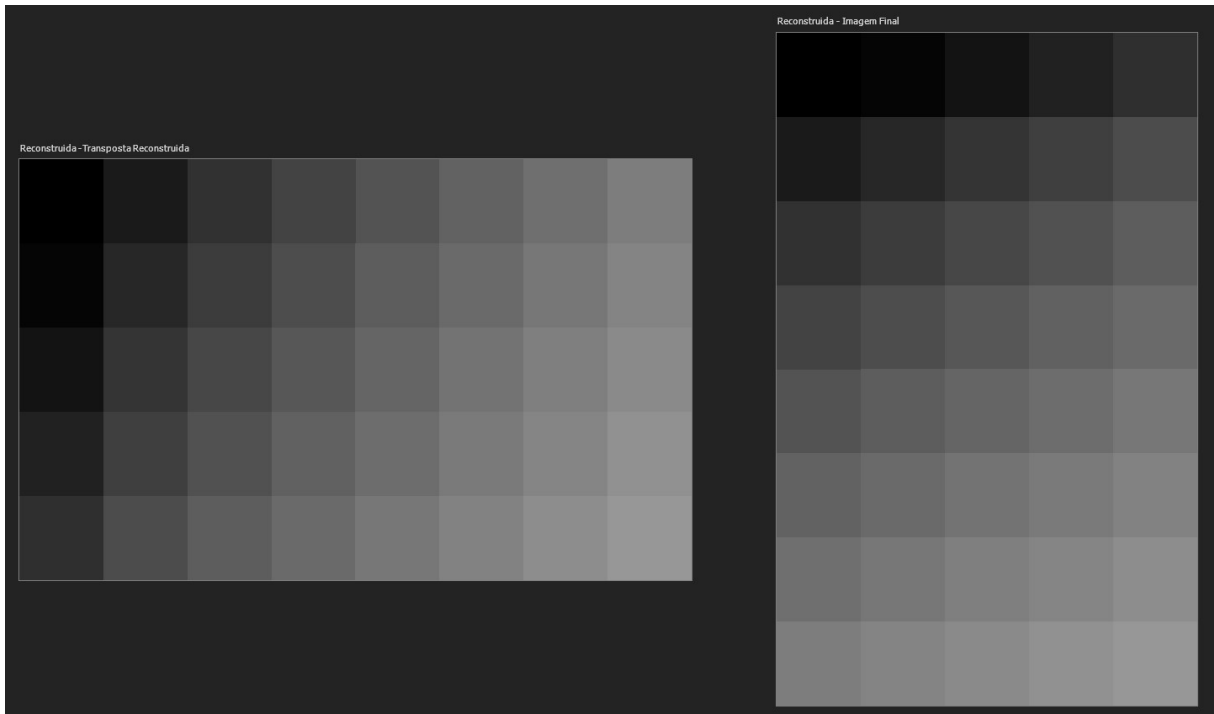
Novamente, após esse processo, iremos juntar todas as linhas e gerar uma nova imagem, conforme mostrado na Figura 17

Figura 17 – Linhas reconstruídas da imagem transposta sendo transformada em imagem.



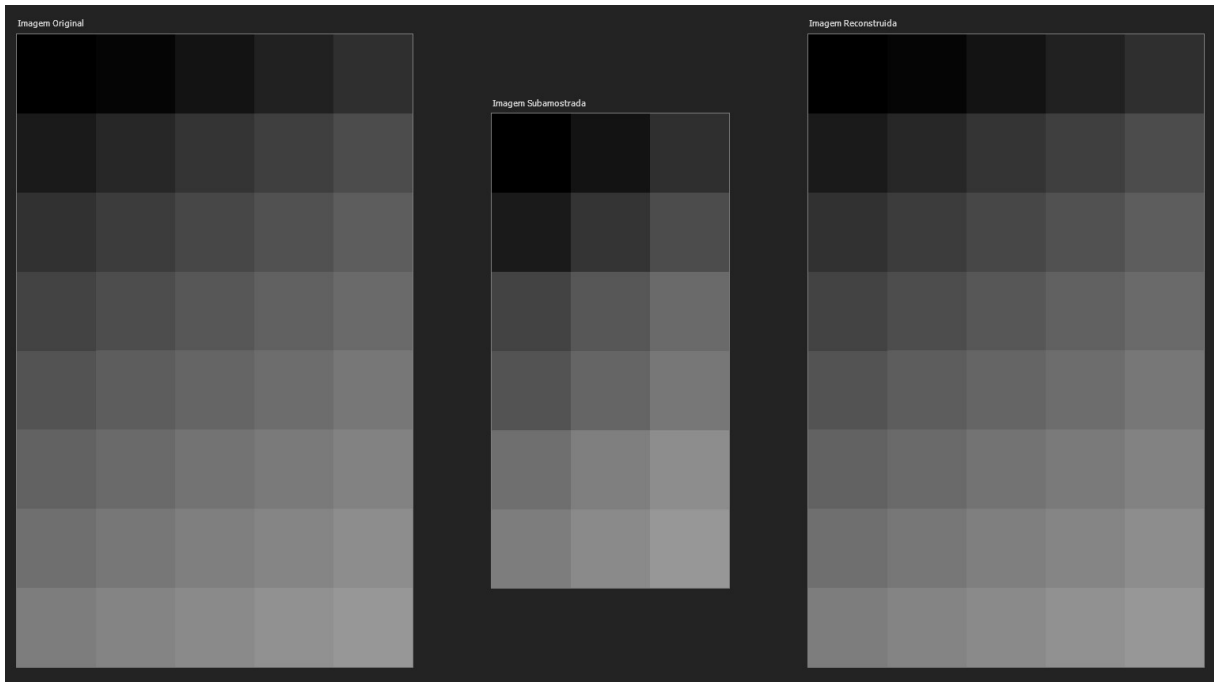
Por fim, iremos novamente trabalhar nessa imagem como uma matriz e realizar a sua transposta, ou seja, a transposta inversa, gerando assim a imagem reconstruída final, que deverá ser a mais fiel possível à imagem original, conforme mostrado na Figura 18

Figura 18 – Imagem transposta reconstruída gerando a imagem reconstruída final através da transposta inversa .



Logo, ao longo do processo podemos comparar a imagem original (espaço-k completo), a imagem sub-amostrada (entrada da rede neural) e imagem final (reconstruída) através da representação da Figura 19

Figura 19 – Imagem original (completa) à esquerda, imagem sub-amostrada (entrada da rede neural) ao centro e imagem final (reconstruída) à direita.



2.12 Análise de qualidade da imagem

Sabendo que o SSIM consegue captar de forma mais precisa se a estrutura visível na imagem é mais fidedigna à original do que o MSE, essa métrica foi considerada a métrica escolhida para a análise final da geração da rede, ou seja, após a imagem ter sido totalmente reconstruída, nós usaremos o algoritmo do SSIM para metrificar o quão boa foi a reconstrução que tivemos.

Será traçada a meta de uma média de 95% de similaridade estrutural em cada intervalo de amostragem.

Resultados

Em primeiro momento foi feita uma abordagem inicial apenas para validar a ideia do trabalho e testar as metodologias estudadas.

Logo de início, ao estudar as possíveis fontes para o dataset, foi sugerido o MRI-DATA.ORG e o mesmo foi validado, perante a larga coleção de dados disponíveis para download, onde facilmente foram adquiridas 1000 imagens para a execução do trabalho.

Em seguida foi avaliada a possibilidade de subamostrar o espaço-k de uma imagem completa adquirida deste dataset e simular o uso de uma imagem sub-amostrada real, através da subamostragem Cartesiana. Tal processo se viável e eficaz, sendo realizado em 10%, 20%, 30%, 40% e 50%.

Com isso feito, testamos a primeira abordagem, utilizando a rede SRGAN modificada para suportar os dados da imagem de ressonância magnética. Embora este processo tenha sido apenas uma abordagem inicial, ele já mostrou bons resultados, conseguindo treinar com eficácia a máquina para reconstruir uma imagem sub-amostrada em 20%, ou seja, com apenas 80% dos dados reais disponíveis.

Como mostrado na Figura 7, a imagem de saída, a partir de uma entrada sub-amostrada, se mostrou válida, com um bom resultado e reconstrução fiel à imagem original.

Como a análise da abordagem inicial foi originalmente o erro quadrático médio, o valor também se mostrou muito bom, com uma média de 0.002 e 0.003, conforme evidenciado na Figura 20.

Figura 20 – Erro quadrático médio da rede neural, onde Epoch representa a época atual, step a etapa desta época, time o tempo para executar tal etapa e mse o erro quadrático médio.

```
Epoch: [99/100] step: [0/12] time: 0.424s, mse: 0.002
Epoch: [99/100] step: [1/12] time: 0.435s, mse: 0.003
Epoch: [99/100] step: [2/12] time: 0.444s, mse: 0.002
Epoch: [99/100] step: [3/12] time: 0.449s, mse: 0.003
Epoch: [99/100] step: [4/12] time: 0.431s, mse: 0.003
Epoch: [99/100] step: [5/12] time: 0.429s, mse: 0.003
Epoch: [99/100] step: [6/12] time: 0.432s, mse: 0.002
Epoch: [99/100] step: [7/12] time: 0.424s, mse: 0.003
Epoch: [99/100] step: [8/12] time: 0.421s, mse: 0.003
Epoch: [99/100] step: [9/12] time: 0.435s, mse: 0.002
Epoch: [99/100] step: [10/12] time: 0.443s, mse: 0.003
```

Já utilizando a métrica do SSIM, tivemos um resultado de 0.9401, ou seja, as estruturas são 94.01% similares.

Porém, algumas dificuldades enfrentadas tornaram necessário alterar o código, como por exemplo, o tempo de treino de apenas 100 imagens ter sido de 18 horas e 24 minutos e a complexidade de entendimento e manutenção se mostrou bem complicado, visto que tanto GANs quanto redes de Super Resolução são abordagens muito *blackbox* (caixa-preta), ou seja, de difícil entendimento do que realmente está acontecendo internamente.

Além disso, devido às alterações feitas para cada subamostragem e às modificações para aceitar as imagens de ressonância magnética em seu estado bruto (no espaço-k) também serem complexas, essa abordagem mostrou inviável.

Portanto, iniciamos a segunda abordagem, que se tornou a abordagem oficial do trabalho, com uma metodologia mais simples, focando em garantir um melhor entendimento de tudo o que está acontecendo, diminuir o tempo de processamento e requisitos maquinários, otimizar o resultado final, tornar viável uma arquitetura organizada de código e conseguir alcançar uma melhor generalização da rede, com isso, possibilitando atingir um bom resultado nas mais diversas ocasiões de uso das imagens de ressonância magnética.

Os resultados dessa segunda abordagem mostraram-se bem eficazes e cumpriram com o esperado inicialmente.

A generalização ocorreu bem, visto que apenas precisamos de 2 treinamentos, um para 10%, 20% e 30% de subamostragem e outro para 40% e 50%.

A arquitetura final mostrou ser bem simples, não necessitando de fazer uso de nenhuma arquitetura já conhecida e com poucas camadas de processamento, além de que, a lógica desta abordagem foi bem simplificada e com isso, o aumentou significativamente o entendimento do que cada camada esta fazendo.

A organização do código também foi aprimorada, conseguindo modularizá-lo por completo, sendo assim, cada etapa é independente da outra, como por exemplo, a etapa de preparação de dados e pré-processamento são independentes entre si e podem ser retiradas quando formos trabalhar com dados reais já coletados de forma subamostrada.

O tempo caiu para 16 horas e 08 minutos de treino com 1000 imagens, para o caso da rede de 10%, 20% e 30% o que em proporção aproximadamente 11.40 vezes mais rápido do que a primeira abordagem, e 19 horas e 44 minutos para um treino com 1000 imagens para a rede de 40% e 50%, o que em proporção é aproximadamente 9.32 vezes mais rápido do que a primeira abordagem.

Por fim, os resultados finais também se mostraram melhores do que na primeira abordagem ao qualificá-los utilizando a métrica do SSIM, conforme mostrado na Tabela 1

Tabela 1 – Resultados da similaridade estrutural média entre as imagens reconstruídas em diferentes porcentagens em relação a imagem original

Porcentagem Sub amostrada	Média SSIM	Desvio Padrão SSIM	Variância SSIM
10%	0.98853	0,00634	0.00004
20%	0.98211	0,00832	0.00007
30%	0.96461	0,01258	0.00016
40%	0.97423	0,01124	0.00013
50%	0.95476	0,01536	0.00024

Para o cálculo da média, desvio padrão e variância, foram utilizadas 100 imagens do conjunto de validação, ou seja, 100 imagens novas que em nenhum momento estiveram presentes no treinamento. Coletando o valor destas 100 reconstruções, foram então calculadas as métricas acima.

Podemos concluir então, que para os dados subamostrados em até 20%, ou seja, com 80% dos dados existentes, a similaridade estrutural foi acima de 98%, ficando bem parecida com a imagem original. Para o caso de 30% de subamostragem, onde existem apenas 70% dos dados, a rede teve uma perda, possivelmente por ter sido gerada junto as imagens de 10% e 20%, porém, ainda acima dos 96% de similaridade.

Já para a segunda rede, o processo foi mais trabalhoso e complicado, visto que a quantidade de números reais existentes na imagem era bem menor, porém, mesmo assim a rede conseguiu uma média acima da expectativa (95%), onde para imagens com apenas 60% dos dados (subamostragem em 40%) tivemos 97% de similaridade e para as de 50% de dados existentes (subamostragem em 50%), tivemos também mais de 95%.

O que foi compreendido com isso, é que a solução proposta neste trabalho se tornou eficaz e validada, conseguindo gerar uma imagem com boa fidelidade até para casos onde a metade dos dados não precisam ser adquiridos.

Todavia, conforme cresce o número de dados subamostrados, vai se tornando cada vez mais difícil a reconstrução, visto que os vizinhos próximos a serem analisados pela rede (principalmente nas camadas de Convolução) se tornam mais escassos e portanto, o número de dados que a rede tem para se basear em como reconstruir o que falta é menor e mais incompleto.

Os valores de variância e desvio padrão mostraram-se baixos, provando que o algoritmo realmente foi eficaz e que o número encontrado como média se manterá próximo para imagens diferentes.

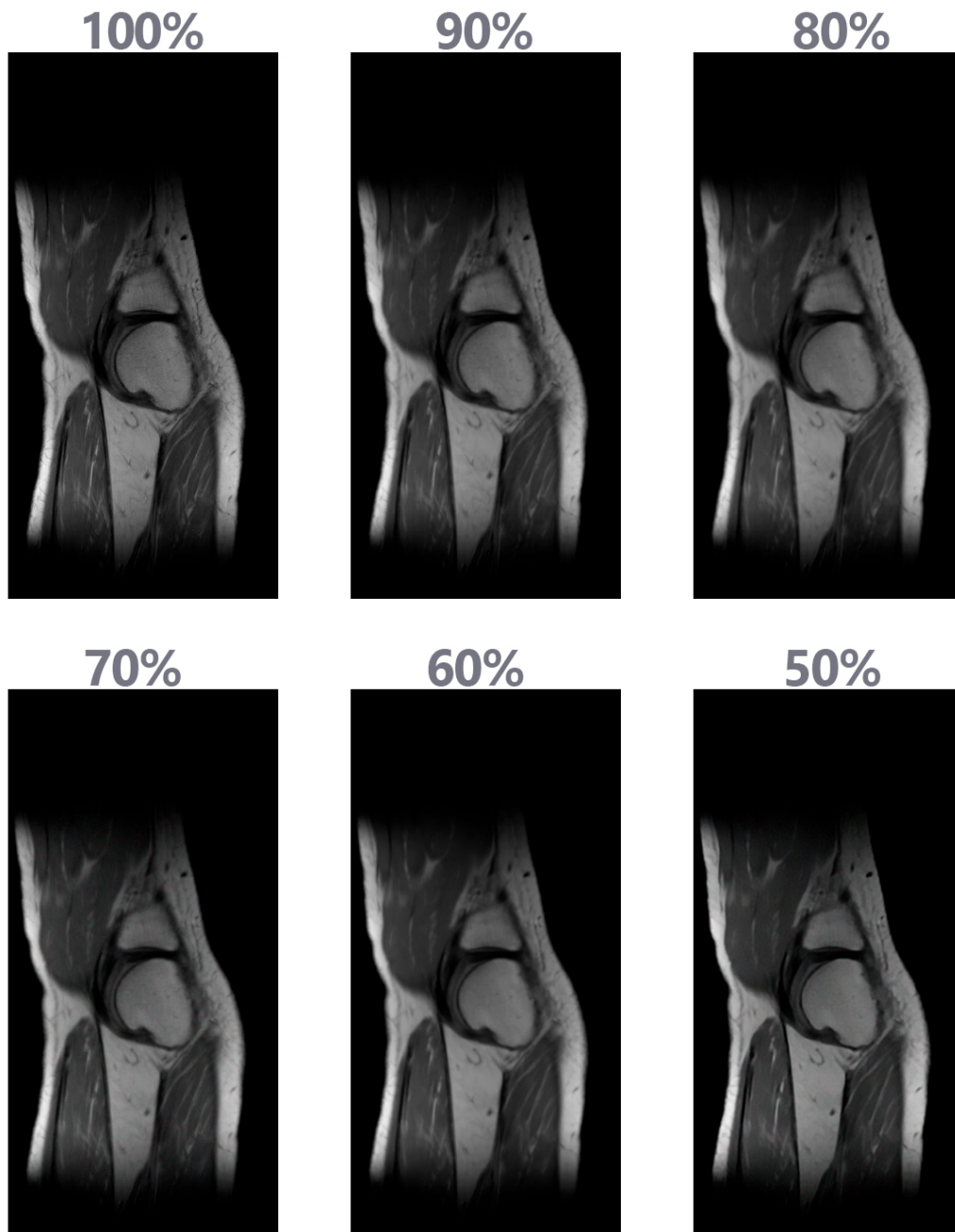
Também foi avaliado o Erro Quadrático Médio, porém, como tal métrica é usada em todo o treinamento o algoritmo teve como foco o tempo todo minimizar ao máximo esse valor e, dessa forma, o valor tendeu a chegar neste mínimo possível em todas as imagens, sendo uma média muito baixa e próxima uma das outras. Na Tabela 1 podemos notar o resultado desta análise.

Tabela 2 – Resultados do erro quadrático médio para as diversas porcentagens de subamostragem

Porcentagem Subamostrada	Erro Quadrático Médio
10%	0.000022
20%	0.000024
30%	0.000027
40%	0.000031
50%	0.000034

Para um melhor entendimento do resultado, a mesma imagem utilizada para exemplificar entrada e saída da abordagem inicial (disponível na Figura 7) foi utilizada para representar o resultado dessa nova abordagem, como disponível na Figura 21, onde na primeira linha temos, da esquerda pra direita, respectivamente, a imagem original (100% dos dados), a imagem sub-amostrada em 10% (90% dos dados existentes) e a imagem sub-amostrada em 20% (com 80% dos dados existentes). Já na segunda linha a imagem sub-amostrada em 30% (70% dos dados existentes), a imagem sub-amostrada em 40% (60% dos dados existentes) e por fim a imagem sub-amostrada em 50% (50% dos dados existentes).

Figura 21 – Imagem com os resultados obtidos na segunda abordagem para todas as subamostragens realizadas.



Também podemos observar com um zoom de 300% a diferença entre a imagem original (100% dos dados) e a reconstruída com uma sub-amostragem em 10% (90% dos dados existentes), que foi o melhor dos casos, na Figura 22 e a diferença entre a imagem original (100% dos dados) com a reconstruída com uma sub-amostragem em 50% (50% dos dados existentes) na Figura 23.

Figura 22 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 10% (90% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.

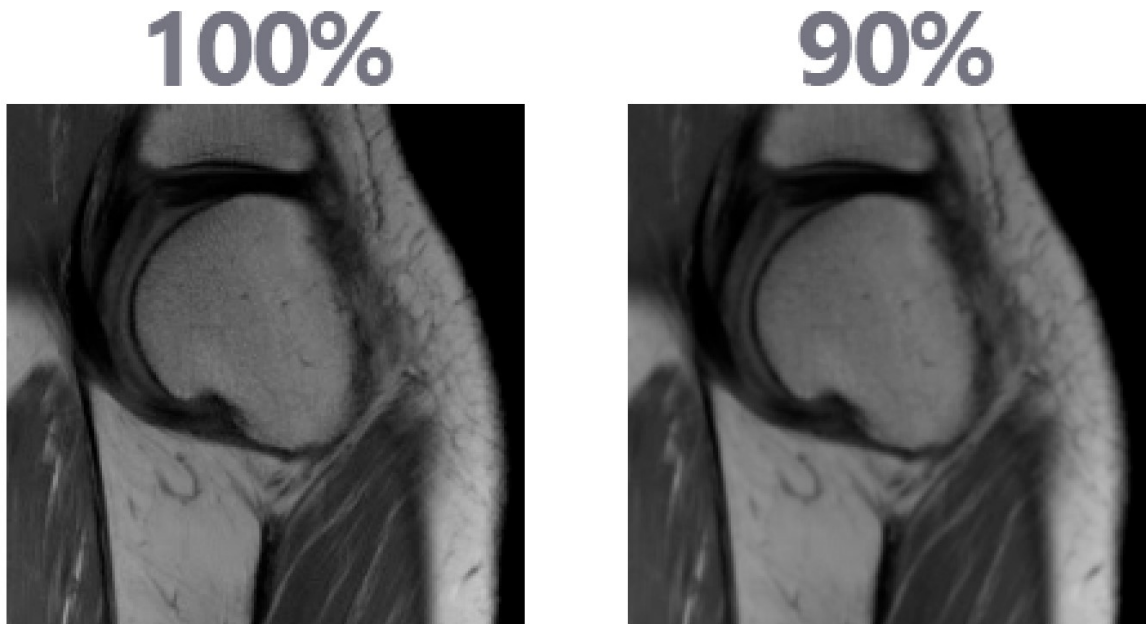
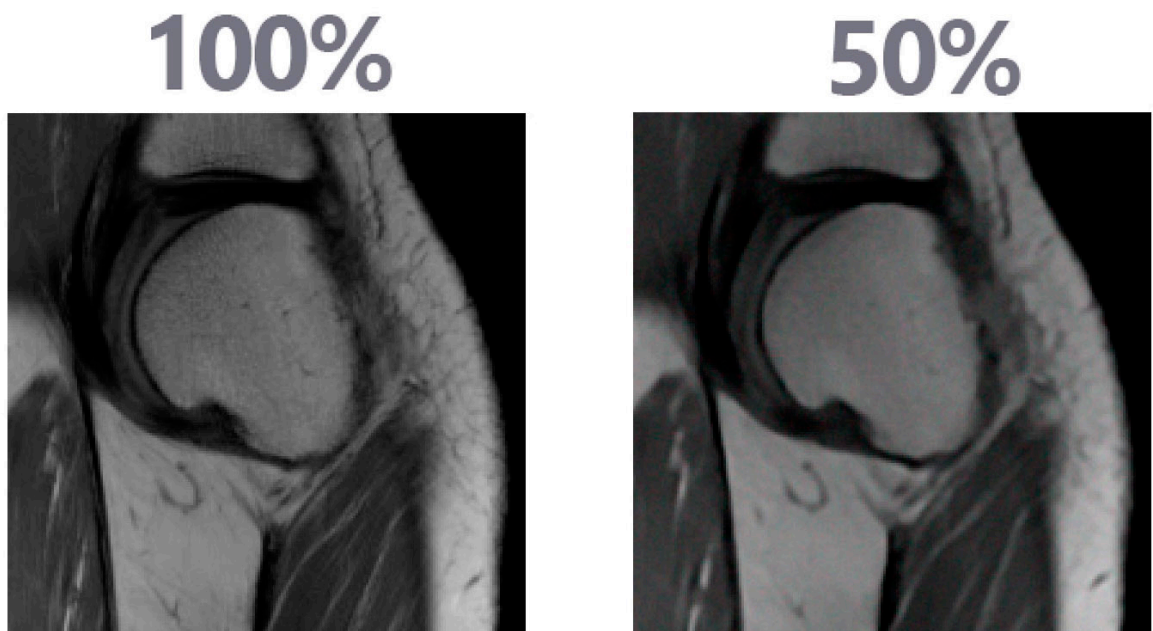


Figura 23 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 50% (50% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.



3.0.1 Validação com dataset secundário

Para garantir o funcionamento do algoritmo em datasets provenientes de máquinas desconhecidas foi feito o teste utilizando outro dataset do site MRIDATA.ORG.

Durante todo o desenvolvimento deste trabalho foi tido como base o dataset "NYU machine learning data" parte devido a já ser próprio para machine learning e parte pelo equipamento e procedimento de aquisição usados serem fiéis aos usados no dia a dia. Todavia, o fato de ser um dataset recomendado para machine learning não quer dizer que é um dataset que estará preparado para os casos reais dos hospitais brasileiros e internacionais.

Portanto, foi escolhido o dataset "Stanford Fullysampled 3D FSE Knees" também do site MRIDATA.ORG como forma de validação do resultado obtido, visto que neste não há nenhuma preparação própria para algoritmos de machine learning.

Para esta validação foi empregue as etapas de aquisição do dataset através do scraper e o pré-processamento dos dados, sem efetuar nenhum outro treinamento do modelo. Em seguida foi rodado apenas a validação dos resultados, para elaboração das métricas de avaliação.

Na Tabela 3 podemos ver os valores referentes as métricas do SSIM.

Tabela 3 – Resultados da similaridade estrutural média entre as imagens reconstruídas em diferentes porcentagens em relação a imagem original

Porcentagem Sub amostrada	Média SSIM	Desvio Padrão SSIM	Variância SSIM
10%	0,94899	0,00824	0,00007
20%	0,93300	0,01165	0,00010
30%	0,88744	0,02139	0,00029
40%	0,82810	0,02573	0,00032
50%	0,78290	0,03165	0,00050

Já na Tabela 4 podemos ver os valores referentes as métricas do MSE

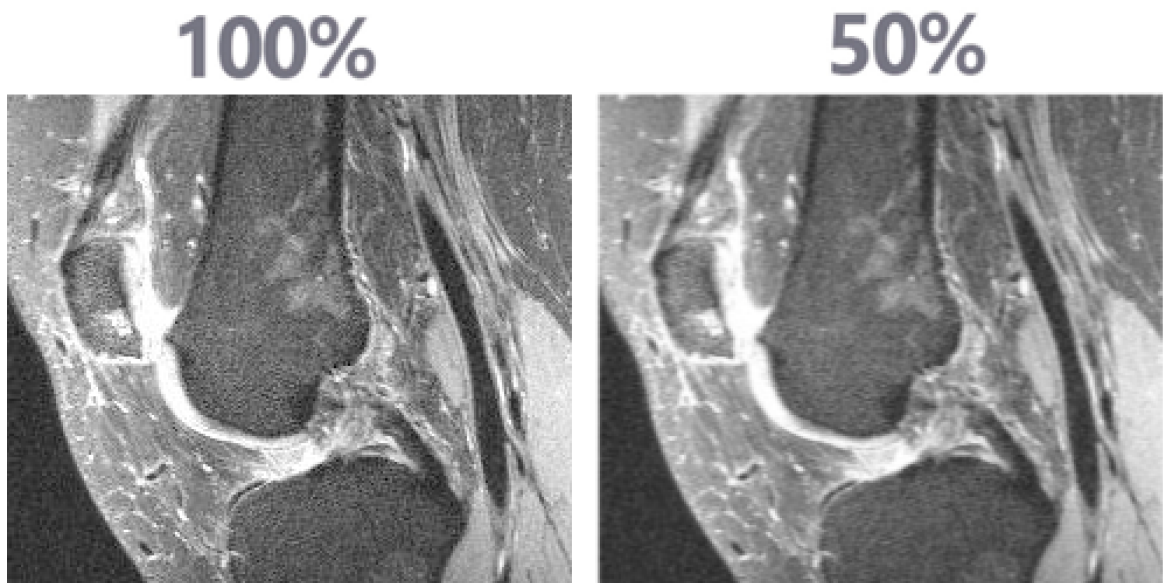
Tabela 4 – Resultados do erro quadrático médio para as diversas porcentagens de subamostragem

Porcentagem Subamostrada	Erro Quadrático Médio
10%	0.000037
20%	0.000041
30%	0.000055
40%	0.000062
50%	0.000081

Embora os resultados tenham parecidos proveitosos no início, com 10% e 20% de subamostragem, nota-se claramente que houve um resultado inferior para os demais valores de subamostragem se comparado com o dataset usado no treinamento original.

Na Figura 24 podemos ver, com um zoom de 300%, o resultado da reconstrução de 50%. Nela é possível notar que a imagem resultante da reconstrução perdeu alguns dos detalhes e todas suas bordas, até as internas, foram suavizadas, dando uma leve impressão de que a imagem está embaçada. Além disso, regiões da imagem que representam uma forma única estão mais finas na reconstrução, como a forma elíptica e escura do lado direito da imagem apresentada na figura.

Figura 24 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 50% (50% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.

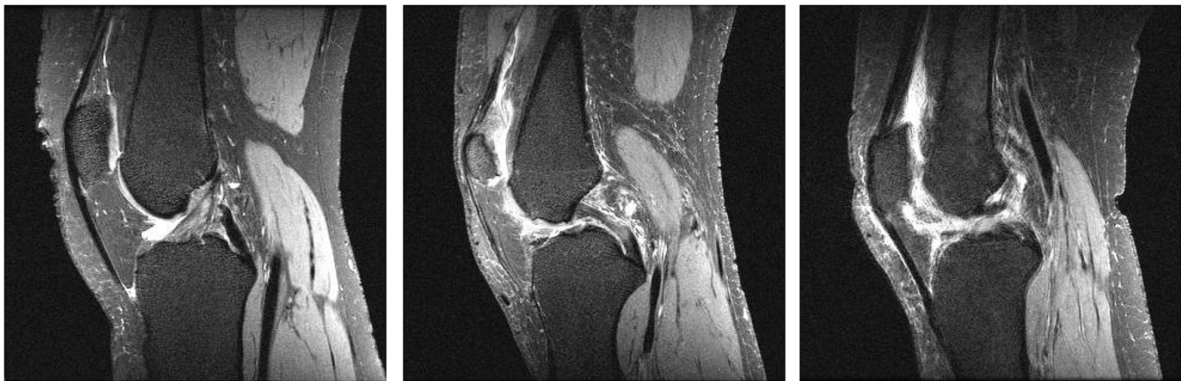


Após analisar com mais cuidado o motivo do ocorrido, foi evidenciado que a imagem gerada em cada uma das máquinas possuíam diferenças significantes até visualmente, conforme a Figura 25 (dataset original, do *NYU machine learning data*) e a Figura 26 (dataset secundário, do *Stanford Fullysampled 3D FSE Knees* evidenciam).

Figura 25 – Três imagens adquiridas do dataset "NYU machine learning data".



Figura 26 – Três imagens adquiridas do dataset "Stanford Fullysampled 3D FSE Knees".



Nota-se que as imagens do dataset de Stanford são mais pixelizadas do que os de NYU e seu intervalo de cores aparenta possuir tonalidades mais saturadas, com um branco e preto mais forte. Além disso, as da máquina provenientes da NYU possuem uma suavização, evitando os serrilhados presentes nas imagens de Stanford.

Pesquisando mais a fundo, nos detalhes de cada dataset no site do MRIDATA.ORG nota-se que há algumas diferenças também nas tecnologias e metodologias de captura das imagens de cada dataset, conforme pode ser visto na Tabela 5 logo abaixo.

Tabela 5 – Dados da captura das imagens de cada um dos datasets utilizados

Propriedade	Stanford	NYU
Largura de Banda do Receptor	50	0.793
Número de Canais	8	15
Tamanho da Matriz	320x320x1	768x770x1
Campo de Visão	160x160x153.6 (mm)	280x280.7x4.5 (mm)

Devido ao funcionamento do algoritmo de separar linha a linha e a rede neural completar os espaços que faltam, isto é, os que foram retirados na subamostragem, essas

propriedades são importantes para a forma com que a máquina aprendeu a reconstruir, pois se as bordas não forem suavizadas, como nas imagens do NYU a diferença de valor (cor) de um pixel para o outro é maior além disso, para imagens mais pixelizadas ou com ruídos há um padrão diferente de imagens mais limpas e com maior qualidade. Por fim, com um campo de visão diferente as imagens possuem diferentes focos e qualidade nos detalhes menores.

Somando tudo isso, ficou evidente que mesmo para imagens da mesma região, no caso, do joelho, se forem adquiridas de diferentes máquinas é necessário a inclusão delas no conjunto de treinamento, para que a rede se torne específica para aquela máquina.

Sabendo disso, foi criado um novo modelo, com dados apenas do dataset de Stanford para uma nova avaliação do algoritmo para diferentes casos.

Foi realizada a execução de todo o fluxo da metodologia final do projeto, desde a aquisição da imagem pelo scraper e pré-processamento dos dados até o treinamento do modelo e execução do mesmo para as avaliações, com as mesmas porcentagens de imagens no conjunto de treinamento, validação e testes, isto é, foram colhidas 1000 imagens, sendo feita uma distribuição de 70% para treino, 20% para validação e 10% para testes.

Na Tabela 6 podemos ver os valores referentes as métricas do SSIM.

Tabela 6 – Resultados da similaridade estrutural média entre as imagens reconstruídas em diferentes porcentagens em relação a imagem original

Porcentagem Sub amostrada	Média SSIM	Desvio Padrão SSIM	Variância SSIM
10%	0,97271	0,00697	0,00004
20%	0,96541	0,00957	0,00009
30%	0,94532	0,01384	0,00018
40%	0,94305	0,01349	0,00017
50%	0,93089	0,01843	0,00029

Já na Tabela 7 podemos ver os valores referentes as métricas do MSE

Tabela 7 – Resultados do erro quadrático médio para as diversas porcentagens de subamostragem

Porcentagem Subamostrada	Erro Quadrático Médio
10%	0.000023
20%	0.000026
30%	0.000030
40%	0.000033
50%	0.000038

Desta vez, onde houve o treinamento do modelo para possuir dados das imagens específicas da máquina em questão, vemos valores que se aproximam mais aos do dataset de NYU. Devido ao fato dos ruídos, pixelização e falta de suavização nas bordas, além de

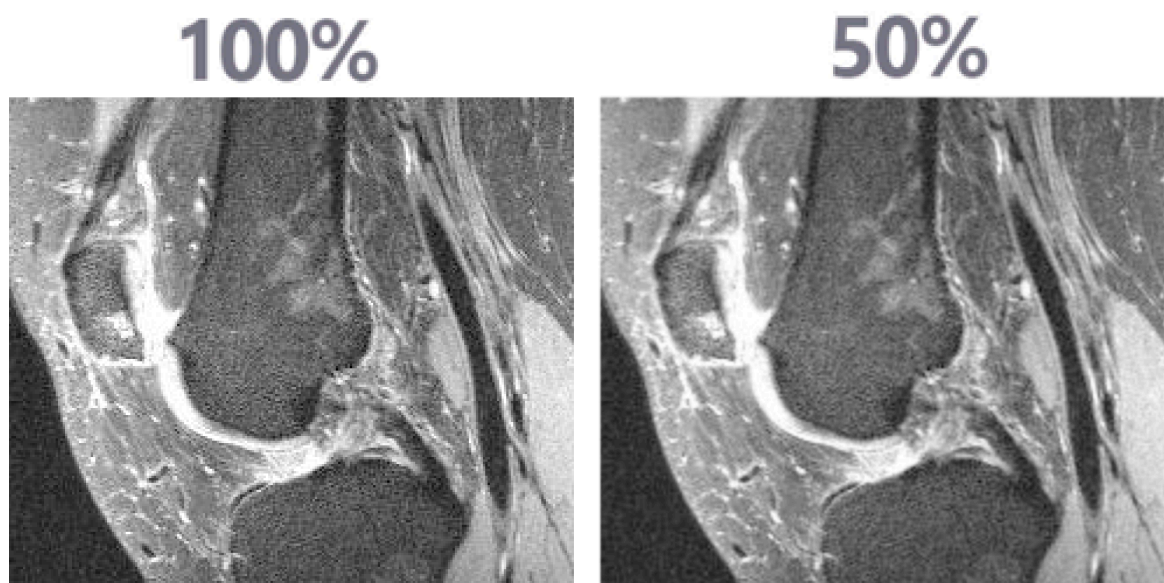
não ser preparado para Machine Learning, este dataset apresentou uma qualidade inferior, porém, já era algo esperado.

Tabela 8 – Resultado da Média do SSIM para cada execução do algoritmo, com o dataset oficial do NYU machine learning data e Stanford Fullysampled 3D FSE Knees, tanto treinado quanto não treinado.

Porcentagem Sub amostrada	NYU	Stanford Treinado	Stanford Não Treinado
10%	0,98853	0,97271	0,94899
20%	0,98211	0,96541	0,93300
30%	0,96461	0,94532	0,88744
40%	0,97423	0,94305	0,82810
50%	0,95476	0,93089	0,78290

A mesma imagem detalhada na Figura 24 foi processada novamente neste modelo treinado para o Stanford. Podemos ver, com um zoom de 300%, o resultado desta reconstrução de 50% na Figura 27.

Figura 27 – Imagem original (100% dos dados existentes) à esquerda e imagem reconstruída com uma subamostragem em 50% (50% dos dados existentes) à direita. Para melhor observação foi aplicado um zoom de 300%.



Agora podemos perceber que embora ainda haja um leve embaçamento na imagem (suavização das bordas) ele é bem menor e quase imperceptível a olho nu e sem o zoom. Além disso, o problema com certas regiões ficarem mais finas na reconstrução acabou, visto que isso era devido ao *salto* de cor, isto é, diferença de valores entre pixel vizinhos ser maior neste dataset do que no NYU, onde era a única fonte de treinamento do modelo.

Por consequência, temos como resultado que o modelo conseguiu aprender de forma eficaz as especificidades do novo dataset e garantir uma reconstrução mais fiel a imagem

original.

3.0.2 Discussão dos Resultados

Ao analisar os resultados obtidos neste trabalho em relação aos demais da literatura, nota-se uma viabilidade em termos de velocidade de processamento e recursos necessários. Isso se dá devido à forma com que a arquitetura da rede foi projetada e qual o objetivo de ter sido feita dessa forma.

Embora os avanços computacionais vem se mostrando eficazes nos últimos tempos, com um poder computacional muito elevado, não são todos que conseguem adquirir os melhores equipamentos do mercado, devido ao alto preço das peças que constituem o maquinário. Ademais, para determinados processos até supercomputadores podem demorar dias para processar, dependendo do nível da complexidade do algoritmo.

Desde o início do trabalho isso já estava sendo pensado, junto com a soma do ideal que o algoritmo deve a todo momento ser entendido por mais programadores.

Para conseguir alcançar ambos os objetivos o processo teve de ser o mais simples possível, de forma que ao mesmo tempo que não há grande complexidade computacional para ser processado, o código é fácil de ser lido e entendido por desenvolvedores externos.

Pensando em redução de complexidade foi utilizada a ideia de dividir e conquistar, junto com o pensamento de que é mais fácil entender algo de pouco em pouco do que tudo de uma vez.

Isso posto, o diferencial deste trabalho em relação aos demais é dividir a imagem (que é sempre tratada como dado único) em vários vetores, onde cada um representa apenas uma linha da imagem.

Com isso, usando como exemplo uma imagem 10x10 (10 pixels de largura por 10 pixels de altura) o computador não precisa entender como trabalhar com uma matriz de 100 valores cada uma, apenas precisa entender como trabalhar com um vetor de 10 valores e replicá-lo 10 vezes.

Assim como para máquina é mais fácil processar 10 dados por vez e dividir esse seu processamento em vários subprocessos (*Threads*) é mais fácil para os algoritmos de Aprendizado de Máquina aprenderem com dados mais simples e pequenos como vetores numéricos do que valores maiores e mais complexos, como matrizes que representam imagens,

Além disso, para a leitura do código é mais fácil entender como a reconstrução de um vetor é feito, visto que ele é apenas uma lista sequencial de números padronizados, do que entender como é reconstruída uma imagem e o que cada parte da rede neural irá fazer

com ela, como por exemplo nas partes onde são executadas as convoluções. Por fim, o número de linhas de código, funções aninhadas e abstração tende a diminuir, visto que os dados usados são mais fáceis e próximos dos dados primitivos da linguagem, usando assim, uma maior porcentagem de recursos nativos da linguagem, sem necessidade de recorrer a pacotes, módulos e *frameworks* externos.

Outro benefício adquirido é o aumento do dataset, visto que ao dividir o conteúdo de uma imagem em vetores nós aumentamos muito o número de inputs enviados para a rede, isto é, ao invés de treinar apenas uma vez para uma imagem 10x10, iremos treinar 10 vezes para um vetor de tamanho 10. Com isso, ao mesmo tempo que é algo simples para aprender, a máquina terá mais exemplos para entender como o processo correto deve ser feito.

Portanto, ao mesmo tempo que focamos em melhorar a legibilidade e entendimento da aplicação, a simplificação do código permite que a complexidade de execução diminua drasticamente e facilite o processamento por parte do computador.

Em resumo, o principal diferencial entre este trabalho e os demais existentes na literatura atual é a forma de trabalhar com a imagem, usando uma iteração sobre suas linhas, ou seja, usamos os vetores que constituem a imagem e não a matriz como um todo. Dessa forma, fica mais fácil e rápido de ser processado e aprendido. Por consequência, o entendimento e aprendizado do algoritmo são elevados e produzem melhores resultados.

Por fim, ficou evidente que o algoritmo funcionou de forma eficaz para diferentes máquinas de ressonância magnética, precisando apenas fazer um novo treinamento com imagens daquela máquina.

Conclusão

Analisando os resultados obtidos conclui-se que a proposta estabelecida neste trabalho foi alcançada com eficácia, gerando bons resultados em todas as subamostragens testadas.

Com a imagem subamostrada em 50% temos apenas que coletar a metade dos dados que a imagem completa (100%) precisou coletar, sendo assim, metade do tempo de captura poderá ser diminuído, gerando tanto um custo operacional menor para o hospital quanto também uma melhor experiência para o paciente, visto que o tempo dentro da máquina será significativamente reduzido.

A abordagem final ficou bem simples em relação as já existentes na literatura, devido a trabalhar com os vetores da imagem ao invés da imagem como um todo, e mostrou-se útil para uma generalização de dados, visto que para qualquer grupo de imagens que venha a ser usado, basta executar um retreino na rede, fazendo-a se adaptar a quantidade sub-mostrada e medidas da imagem original (formato, tamanho e proporção).

Espera-se que os resultados obtidos também sejam úteis para estudos futuros, para uma melhoria da metodologia, testes em imagens de outros conjuntos de dados e em outras regiões do corpo. E eventualmente, empregada para uso real nos hospitais e máquinas de ressonância magnética.

Referências Bibliográficas

- BARR, A.; FEIGENBAUM, E. A. The handbook of artificial intelligence. william kaufmann. *Inc., Los Altos, CA*, p. 163–171, 1981.
- BLOCH, F.; HANSEN, W.; PACKARD, M. Physical review. *Phys Rev*, v. 70, p. 460–473, 1946.
- CHOY, G. et al. Current applications and future impact of machine learning in radiology. *Radiology*, Radiological Society of North America, v. 288, n. 2, p. 318–328, 2018.
- CONSTANTINIDES, C. *Assessment of Cellular and Organ Function and Dysfunction using Direct and Derived MRI Methodologies*. [S.l.]: BoD–Books on Demand, 2016.
- CRESWELL, A. et al. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, IEEE, v. 35, n. 1, p. 53–65, 2018.
- FOO, T. K. et al. Lightweight, compact, and high-performance 3 t mr system for imaging the brain and extremities. *Magnetic resonance in medicine*, Wiley Online Library, v. 80, n. 5, p. 2232–2245, 2018.
- FOSTER, M. A. et al. *Magnetic resonance in medicine and biology*. [S.l.]: Pergamon Press, 1985.
- GIGER, M. L. Machine learning in medical imaging. *Journal of the American College of Radiology*, Elsevier, v. 15, n. 3, p. 512–520, 2018.
- GOODFELLOW, I. et al. Generative adversarial networks. *Communications of the ACM*, ACM New York, NY, USA, v. 63, n. 11, p. 139–144, 2020.
- HYUN, C. M. et al. Deep learning for undersampled mri reconstruction. *Physics in Medicine & Biology*, IOP Publishing, v. 63, n. 13, p. 135007, 2018.
- ISOLA, P. et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018.
- KOCAK, M. *Imagem por ressonância magnética (RM)*. 2019. <<https://www.msmanuals.com/pt/casa/assuntos-especiais/exames-de-diagnóstico-por-imagem-comuns/imagem-por-ressonância-magnética-rm>>. Acessado em: dia 26 de Junho de 2020.
- KONAR, A. *Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain*. [S.l.]: CRC press, 2018.

- LAUTERBUR, P. C. Image formation by induced local interactions: examples employing nuclear magnetic resonance. *nature*, Nature Publishing Group, v. 242, n. 5394, p. 190–191, 1973.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.
- LEDIG, C. et al. Photo-realistic single image super-resolution using a generative adversarial network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4681–4690.
- 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.
- MEHRTASH, A. et al. Deepinfer: Open-source deep learning deployment toolkit for image-guided therapy. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling*. [S.l.], 2017. v. 10135, p. 101351K.
- MELLIT, A.; KALOGIROU, S. A. Artificial intelligence techniques for photovoltaic applications: A review. *Progress in energy and combustion science*, Elsevier, v. 34, n. 5, p. 574–632, 2008.
- ODENA, A.; OLAH, C.; SHLENS, J. *Conditional Image Synthesis With Auxiliary Classifier GANs*. 2017.
- QUAN, T. M.; NGUYEN-DUC, T.; JEONG, W.-K. Compressed sensing mri reconstruction using a generative adversarial network with a cyclic loss. *IEEE transactions on medical imaging*, IEEE, v. 37, n. 6, p. 1488–1497, 2018.
- RECHT, M. P. et al. Optimization of mri turnaround times through the use of dockable tables and innovative architectural design strategies. *American Journal of Roentgenology*, Am Roentgen Ray Soc, v. 212, n. 4, p. 855–858, 2019.
- SANDILYA, M.; NIRMALA, S. Compressed sensing trends in magnetic resonance imaging. *Engineering science and technology, an international journal*, Elsevier, v. 20, n. 4, p. 1342–1352, 2017.
- SANTINI, F.; BIERI, O.; DELIGIANNI, X. Openforce mr: A low-cost open-source mr-compatible force sensor. *Concepts in Magnetic Resonance Part B: Magnetic Resonance Engineering*, Wiley Online Library, v. 48, n. 4, p. e21404, 2018.
- SENDERS, J. T. et al. Natural and artificial intelligence in neurosurgery: a systematic review. *Neurosurgery*, Oxford University Press, v. 83, n. 2, p. 181–192, 2018.
- SHRIVIDYA, G.; BHARATHI, S. Application of compressed sensing on magnetic resonance imaging: A brief survey. In: IEEE. *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. [S.l.], 2016. p. 2037–2041.
- SOUZA, R.; FRAYNE, R. A hybrid frequency-domain/image-domain deep network for magnetic resonance image reconstruction. In: IEEE. *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2019. p. 257–264.

- WANG, Z. et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, IEEE, v. 13, n. 4, p. 600–612, 2004.
- YANG, G. et al. Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. *IEEE transactions on medical imaging*, IEEE, v. 37, n. 6, p. 1310–1321, 2017.
- YASAKA, K. et al. Deep learning with convolutional neural network in radiology. *Japanese journal of radiology*, Springer, v. 36, n. 4, p. 257–272, 2018.
- ZHU, B. et al. Image reconstruction by domain-transform manifold learning. *Nature*, Nature Publishing Group, v. 555, n. 7697, p. 487–492, 2018.
- ZHU, J.-Y. et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2020.