

UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MARCO ANTÔNIO BERNARDI GRIVOL

**DIAGNÓSTICO DA DOENÇA DE ALZHEIMER USANDO  
AUTOENCODERS APLICADOS A IMAGENS DE  
RESSONÂNCIA MAGNÉTICA**

SÃO CARLOS  
2022



MARCO ANTÔNIO BERNARDI GRIVOL

**DIAGNÓSTICO DA DOENÇA DE ALZHEIMER USANDO  
AUTOENCODERS APLICADOS A IMAGENS DE  
RESSONÂNCIA MAGNÉTICA**

Trabalho de conclusão de curso apresentado à Universidade Federal de São Carlos como requisito para obtenção de graduação em Bacharelado em Ciência da Computação

Orientador: Prof. Dr. Ricardo José Ferrari

SÃO CARLOS  
2022

# Resumo

A doença de Alzheimer (DA) é uma doença neurodegenerativa que causa danos associados à memória e pensamento, acarretando declínio gradual da capacidade de julgamento, raciocínio e aprendizagem. Uma das formas de auxílio ao diagnóstico da DA é a análise de imagens estruturais de ressonância magnética (RM) do paciente. Estudos recentes utilizam técnicas de visão computacional, processamento de imagens e aprendizado de máquina com o objetivo de auxiliar o diagnóstico da DA. Este trabalho tem como objetivo a extração de características de imagens de RM com autoencoders empilhados e autoencoders convolucionais empilhados, e a classificação das imagens de RM via XGBoost nas classes cognitivamente normal (CN) e DA. Este trabalho também compara as métricas *Mean Squared Error* (MSE) e *Structural Similarity Index* (SSIM) como funções de perda nos autoencoders.

# Sumário

Sumário . . . . .	iii	
<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>5</b>
1.1	Contextualização e Motivação . . . . .	5
1.2	Objetivo Geral . . . . .	5
1.3	Objetivos Específicos . . . . .	6
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>7</b>
2.1	Doença de Alzheimer . . . . .	7
2.2	Autoencoders . . . . .	7
2.2.1	Autoencoders Convolucionais . . . . .	8
2.3	SSIM . . . . .	9
2.4	XGBoost . . . . .	11
2.5	Métricas de Avaliação . . . . .	12
2.6	Revisão Bibliográfica . . . . .	13
<b>3</b>	<b>METODOLOGIA . . . . .</b>	<b>15</b>
3.1	Visão Geral . . . . .	15
3.2	Imagens . . . . .	15
3.2.1	Bases de Imagens . . . . .	15
3.2.2	Pré-Processamento . . . . .	16
3.2.3	<i>Data Augmentation</i> . . . . .	16
3.2.4	Estratificação por Idade . . . . .	17
3.3	Validação Cruzada Aninhada . . . . .	18
3.4	Arquitetura dos Autoencoders . . . . .	18
3.4.1	Autoencoder Empilhado com MSE . . . . .	19
3.4.2	Autoencoder Empilhado com <i>SSIM</i> . . . . .	20
3.4.3	Autoencoder Empilhado Convolucional . . . . .	21
3.5	Hiperparâmetros do Classificador . . . . .	21
<b>4</b>	<b>RESULTADOS E DISCUSSÃO . . . . .</b>	<b>23</b>
4.1	Treinamento dos Autoencoders . . . . .	23
4.1.1	AE com função de perda MSE . . . . .	23
4.1.2	AE com função de perda <i>SSIM</i> . . . . .	25
4.1.3	ACE . . . . .	26
4.2	Resultados da Classificação . . . . .	27
<b>5</b>	<b>CONCLUSÕES . . . . .</b>	<b>31</b>
	Referências . . . . .	33



# 1 Introdução

## 1.1 Contextualização e Motivação

A doença de Alzheimer (DA) é uma doença neurodegenerativa e a forma mais comum de demência, representando cerca de 60 - 80% de todos os casos (THIES; BLEILER, 2012). Demência é um termo utilizado para descrever diversos tipos de doenças e condições decorrentes da morte ou falha neuronal que afetam a memória, o comportamento e a habilidade de raciocinar claramente. Pacientes com DA apresentam um declínio gradual na capacidade de reter novas informações devido ao dano causado nas regiões do cérebro relacionadas em formar novas memórias. Com o progresso da doença, indivíduos com DA apresentam redução na capacidade cognitiva e dificuldades em realizar atividades cotidianas, como andar e engolir (THIES; BLEILER, 2012).

Com o aumento da expectativa de vida no mundo, doenças neurodegenerativas crônicas tornam-se mais prevalente e, no caso do Alzheimer, o diagnóstico precoce permite iniciar o tratamento o mais cedo possível e, assim, retardar os sintomas da doença (PRINCE; BRYCE; FERRI, 2011; PRINCE et al., 2015). Dessa forma, métodos computacionais precisos e eficazes para diagnóstico tornam-se ferramentas importantes para acelerar o diagnóstico da doença.

Com base no entendimento do processo de neurodegeneração da DA é possível utilizar biomarcadores de imagens de Ressonância Magnética (RM) para auxiliar o diagnóstico e excluir outras possíveis comorbidades de demências tratáveis, ou sugerir a presença de comorbidades que agravam os sintomas da demência, como a doença cerebrovascular (GAUTHIER et al., 2021; TAKIZAWA et al., 2015). Uma forma de diagnosticar a DA com imagens de RM é a observação de padrões específicos de atrofia advindos da perda de tecidos (KNOPMAN et al., 2021). Dessa forma, estudos recentes utilizando ferramentas computacionais buscam desenvolver métodos para classificação da DA em imagens de RM.

O resultado produzido pela técnica de RM é uma imagem 3D, usualmente visualizada por médicos como imagens 2D com a seleção de cortes nos planos axial, coronal e sagital. Somado a isso, a atrofia na região dos hipocampos mencionada anteriormente é utilizada como um indicativo da presença e progresso da DA via uso da escala *Medial Temporal lobe Atrophy* (MTA) (GAUTHIER et al., 2021).

Perante o exposto, este trabalho utiliza imagens bidimensionais de RM estrutural dos três planos da região hipocampal para a classificação das imagens nas classes cognitivamente normal (CN) e DA. Para isso, são utilizados autoencoders para extração de características das imagens e, posteriormente, a classificação desses atributos com o classificador XGBoost (*eXtreme Gradient Boosting*) (CHEN; GUESTRIN, 2016)]. São propostos três autoencoders (GOODFELLOW; BENGIO; COURVILLE, 2016; LECUN et al., 1998): um autoencoder empilhado com função de perda *Mean Squared Error* (MSE), um autoencoder empilhado com função de perda *Structural Similarity Index Measure* (SSIM) (WANG et al., 2004); um autoencoder convolucional empilhado. Devido à baixa quantidade de amostras da classe DA, os autoencoders são treinados exclusivamente com amostras da classe CN e, portanto, é esperado que os dados extraídos do *bottleneck* para uma imagem da classe DA apresentem diferenças significativas para diferenciação da classe CN por meio do classificador.

## 1.2 Objetivo Geral

O principal objetivo deste trabalho conclusão de curso (TCC) de graduação é a classificação de imagens 2D de RM utilizando a extração de características via autoencoders e classificação nas classes CN e DA com XGBoost.

### 1.3 Objetivos Específicos

Para alcançar o objetivo geral do projeto, alguns objetivos específicos foram definidos:

1. Estudar a teoria do aprendizado não supervisionado com autoencoders.
2. Estudar a teoria do aprendizado supervisionado com XGBoost.
3. Organização e preparação das imagens que serão utilizadas neste trabalho.
4. Treinamento dos autoencoders especializados na classe CN nos planos axial, coronal e sagital.
5. Transferência de aprendizado para os autoencoders especializados por faixa etária.
6. Classificação nas classes CN e DA dos atributos extraídos via autoencoders com XGBoost.
7. Avaliação quantitativa com métricas de acurácia,  $F_1$ -score, precisão e revocação.
8. Análise e discussão dos resultados.

## 2 Fundamentação Teórica

Este capítulo apresenta os principais fundamentos teóricos das técnicas utilizadas neste projeto.

### 2.1 Doença de Alzheimer

As doenças neurodegenerativas são patologias caracterizadas pela destruição irreversível de neurônios de certas regiões do cérebro, o que leva à perda progressiva e incapacitante de determinadas funções do sistema nervoso (FALCO et al., 2016). Os sintomas iniciais da DA são caracterizados pela perda de memória e dificuldades cognitivas (THIES; BLEILER, 2012). Os dados neuropatológicos mais relevantes em pacientes de DA são a presença de atrofia cortical difusa, degeneração neurovascular, perdas neuronais e sinápticas envolvendo vários sistemas de neurotransmissão (FALCO et al., 2016). A perda sináptica e neuronal são as causas mais prováveis associadas à perda cognitiva (KNOPMAN et al., 2021).

O comprometimento cognitivo aumenta exponencialmente em pessoas de idade avançada, aumentando significativamente depois dos 65 anos. A incidência de demência em indivíduos entre 65 – 70 anos é de aproximadamente 1 a cada 100 por ano e aumenta para 4 a cada 100 por ano entre indivíduos com 80 a 90 anos (KNOPMAN et al., 2021). Comprometimento Cognitivo Leve (CCL) refere-se aos primeiros sintomas de declínio cognitivo que afetam uma ou mais áreas sem comprometer significativamente atividades funcionais diárias (KNOPMAN et al., 2021; GAUTHIER et al., 2021). Um paciente deixa de possuir CCL e passa a possuir demência quando o comprometimento cognitivo afeta as habilidades de independência para realizar tarefas diárias (KNOPMAN et al., 2021).

A patologia da doença também apresenta o acúmulo de placas senis extracelulares compostas de agregados filamentosos da proteína  $\beta$ -amiloide e massas neurofibrilares intracelulares, formadas principalmente pela proteína tau (FALCO et al., 2016; KNOPMAN et al., 2021). As placas e massas presentes no cérebro de pacientes acometidos pela DA estão localizadas sobretudo nas amígdalas cerebelosas, nos hipocampus e no córtex entorrinal do lóbulo temporal (FALCO et al., 2016).

Devido ao processo de neurodegeneração da DA é possível utilizar biomarcadores de imagens de RM para auxiliar o diagnóstico e excluir outras possíveis comorbidades de demências tratáveis, ou sugerir a presença de outras comorbidades que agravam os sintomas da demência, como a doença cerebrovascular (TAKIZAWA et al., 2015; GAUTHIER et al., 2021). Uma das formas de diagnosticar a doença via imagens de RM é a observação de padrões específicos de atrofia advindos da perda de tecidos (KNOPMAN et al., 2021). Dessa forma, estudos recentes utilizando o diagnóstico de DA com imagens de RM buscam desenvolver métodos capazes de reconhecer as diferenças entre imagens de cérebros normais e cérebros acometidos pela neurodegeneração ocasionada pela doença.

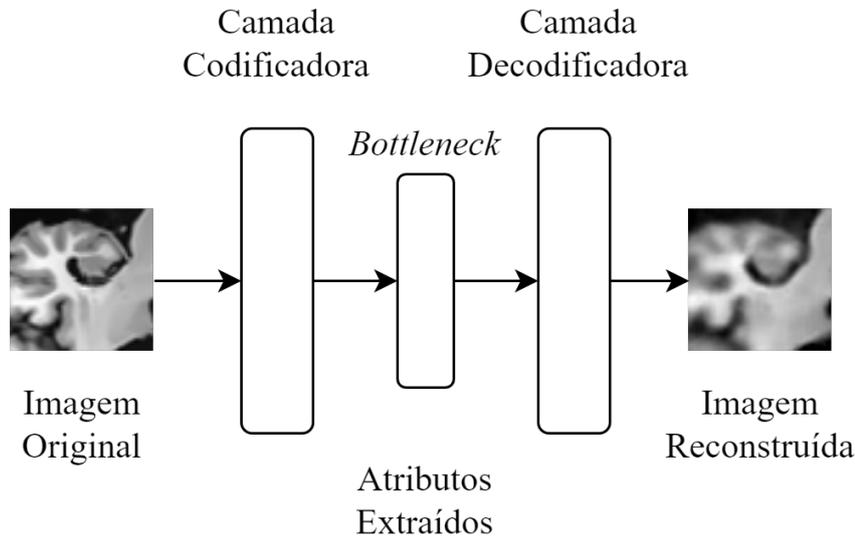
### 2.2 Autoencoders

Um autoencoder é uma rede treinada para reconstruir sua entrada na camada de saída (BANK; KOENIGSTEIN; GIRYES, 2021; GOODFELLOW; BENGIO; COURVILLE, 2016). O objetivo de um autoencoder é aprender as funções  $A : \mathbb{R}^n \rightarrow \mathbb{R}^p$  (codificação) e  $B : \mathbb{R}^p \rightarrow \mathbb{R}^n$  (decodificação) que satisfazem

$$\min E(A, B) = \arg \min_{A, B} \Delta(\mathbf{x}, B \circ A(\mathbf{x})), \quad (2.1)$$

em que  $\mathbf{x}$  é a imagem original,  $B \circ A(\mathbf{x})$  é a imagem reconstruída e  $\Delta$  é a função de perda da reconstrução entre a entrada na etapa de codificação e saída da decodificação (BALDI, 2012; BANK; KOENIGSTEIN; GIRYES, 2021).

Figura 1 – Exemplo simplificado de um autoencoder com uma única camada de codificação e decodificação.



Fonte: elaborada pelo autor.

Autoencoders são utilizados em tarefas de redução de dimensionalidade e extração de atributos via redução na quantidade de nós em cada camada na etapa de codificação, culminando em um gargalo (*bottleneck*, vide Figura 1), i.e.,  $p < n$ . Esses autoencoders são considerados uma generalização da técnica Análise de Componentes Principais ou Principal Component Analysis (PCA) (JOLLIFFE, 2002), os quais permitem o aprendizado de dados não lineares (GOODFELLOW; BENGIO; COURVILLE, 2016; BANK; KOENIGSTEIN; GIRYES, 2021). No caso em que  $p \geq n$ , obtém-se um autoencoder *overcomplete*.

O treinamento de um autoencoder ocorre em duas etapas, o pré-treinamento guloso por camadas (*greedy layer-wise*), seguido pela etapa de *fine-tuning*. Na etapa de pré-treinamento, cada camada é treinada individualmente de forma não-supervisionada, i.e., a primeira camada da codificação e decodificação são unidas e treinadas, em seguida, os pesos dessas camadas são fixados e a segunda camada de cada etapa é então adicionada (empilhada) para o treinamento utilizando como entrada a saída da camada anterior. O objetivo dessa etapa é obter uma boa inicialização dos pesos para todas as camadas. Ao final deste processo, é obtido um Autoencoder Empilhado (AE) cujos pesos finais são ajustados em uma etapa de *fine-tuning* com todas camadas juntas (GOODFELLOW; BENGIO; COURVILLE, 2016).

### 2.2.1 Autoencoders Convolucionais

Um dos problemas de redes neurais com camadas totalmente conectadas e, conseqüentemente, autoencoders com essas camadas, é a incapacidade de tratar translações, mudanças de escala e distorções locais nos dados de forma eficiente (LECUN et al., 1998). Somado a isso, imagens apresentam forte correlação em estruturas locais (LECUN et al., 1998). Dessa forma, redes neurais convolucionais extraem características locais dos dados via implementação da operação de convolução no lugar de uma multiplicação de matrizes (LECUN et al., 1998; GOODFELLOW; BENGIO; COURVILLE, 2016). Embora o termo convolução seja amplamente utilizado na literatura ao tratar esses tipos de redes, este é um nome impróprio uma vez que essas redes implementam a operação de correlação cruzada, todavia por convenção histórica, este trabalho utiliza o termo convolução para nomear essas redes e operações.

Seja  $w$  um filtro, ou *kernel*, de dimensão  $m \times n$ , com  $a = m - 1$  e  $b = n - 1$ . A operação de correlação cruzada de  $w$  com uma imagem  $f(x,y)$ , onde  $x$  e  $y$  são coordenadas de pixels em  $f$ , denotada por  $(w \star f)(x,y)$ , é definida pela seguinte expressão (GONZALEZ; WOODS, 2018):

$$(w \star f)(x, y) = \sum_{s=0}^a \sum_{t=0}^b w(s, t) f\left(x + s - \frac{a}{2}, y + t - \frac{b}{2}\right). \quad (2.2)$$

Camadas tradicionais totalmente conectadas utilizam multiplicações de matrizes para descrever as interações entre os nós da rede. Por outro lado, redes convolucionais possuem interações esparsas ao utilizar um *kernel* de dimensões menores que a entrada, e.g., uma imagem pode conter milhares ou milhões de pixels, mas ao utilizar um *kernel* com apenas algumas poucas dezenas de pixels, é possível extrair características significativas, como cantos ou bordas, utilizando menos parâmetros e, consequentemente, reduzindo os requisitos de memória do modelo e a quantidade de operações computacionais necessárias (GOODFELLOW; BENGIO; COURVILLE, 2016).

Somado a isso, a operação de convolução apresenta o princípio da equivariância à translações, i.e., uma função  $f(x)$  é equivariante à uma função  $g$  se  $f(g(x)) = g(f(x))$ . Ao trabalhar com imagens, se  $g$  é uma função que translada a entrada então ao aplicar a operação de convolução  $f$  em  $g(x)$ , obtém-se uma invariância à translação (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa propriedade é especialmente importante ao trabalhar com imagens de RM visto que, mesmo após o correto registro de imagens, diferenças inerentes entre o cérebro de cada paciente e ao mecanismo de aquisição das imagens, duas imagens de RM podem não estar totalmente alinhadas.

Após a etapa de convolução ( $w \star f$ ), uma segunda etapa de *pooling* reduz as dimensões da entrada por meio de uma função não linear que substitui uma região da entrada por um “resumo estatístico” dos valores próximos à esta região (GOODFELLOW; BENGIO; COURVILLE, 2016). Operações de *pooling* são úteis pois ajudam a reduzir a sensibilidade da rede à translações e distorções locais, i.e., após a detecção de uma característica importante através do *kernel* na convolução, a posição exata desse atributo torna-se menos importante. Apenas sua posição aproximada relativa à outros atributos é relevante (LECUN et al., 1998). Este trabalho utiliza a operação de *max pooling* que extrai o valor máximo em uma região local retangular da imagem.

Assim como os AE, Autoencoders Convolucionais (AC) tem como objetivo a extração de características em um *bottleneck* via etapas de codificação e decodificação. Todavia, AC implementam as operações de convolução e *pooling* na etapa de codificação e, operações de de-convolução (*deconvolution*) e *upsampling* na etapa de decodificação. O treinamento de um AC também pode ser realizado por camadas empilhadas, nesse caso denomina-se Autoencoder Convolutional Empilhado (ACE).

## 2.3 SSIM

A função erro médio quadrado, do inglês *Mean Squared Error* (MSE), é uma das métricas (de dissimilaridade) mais usadas para comparar imagens (WANG et al., 2004). MSE pode ser utilizada como função de perda entre a imagem original  $\mathbf{x}$  e a imagem reconstruída  $\mathbf{y} = B \circ A(\mathbf{x})$  como:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2. \quad (2.3)$$

A métrica MSE é de fácil cálculo e matematicamente conveniente para problemas de otimização. Todavia, pode não ser uma métrica adequada para medir a fidelidade e qualidade entre imagens, i.e., duas imagens distorcidas com valores iguais de MSE podem apresentar grandes diferenças no tipo de distorção (WANG et al., 2004). O exemplo ilustrativo da Figura 2 apresenta o problema de MSE ao comparar as imagens claramente distintas mas que apresentam valores de MSE próximos.

Figura 2 – Comparação entre MSE e SSIM. (a) Imagem original. (b) Ruído “sal e pimenta”. (c) Suavização com filtro gaussiano. (d) Transformação de intensidade gamma.



Fonte: elaborada pelo autor.

Perante o exposto, WANG et al. (2004) apresentam o framework *Structural Similarity Index* (SSIM) para medir a similaridade entre imagens. Esse framework é inspirado no funcionamento do sistema de visão humana e busca avaliar a degradação estrutural da informação em imagens, avaliando se a informação original pode ser reconstruída via transformação linear na luminância das imagens. Os autores definem informação estrutural de uma imagem como os atributos que representam a estrutura de objetos na cena, independente da média de luminância e contraste.

Para comparar uma imagem referência  $\mathbf{x}$  com uma possível imagem degradada  $\mathbf{y}$ , as medidas de luminância, contraste e estrutura são comparadas independentemente. A primeira etapa é a comparação da luminância entre as imagens, a média de intensidade é definida como

$$\mu_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N x_i. \quad (2.4)$$

A função de comparação para luminância é dada por

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1}{\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1}, \quad (2.5)$$

em que  $C_1$  é uma constante para evitar instabilidade quando  $\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2$  aproxima-se de zero e pode ser definida como  $C_1 = (K_1 L)^2$ , com  $L$  representando a variação máxima de intensidade (255 para imagens com 8 bits de profundidade), e  $K_1 \ll 1$  como uma pequena constante. Em seguida, a média de intensidade  $\mu_{\mathbf{x}}$  é subtraída da imagem, resultando em  $\mathbf{x} - \mu_{\mathbf{x}}$ , tal que  $\sum_{i=1}^N x_i = 0$ . A seguinte equação estima o contraste do sinal:

$$\sigma_{\mathbf{x}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_{\mathbf{x}})^2}. \quad (2.6)$$

A função de comparação  $c(\mathbf{x}, \mathbf{y})$  para o contraste entre as imagens é dada por:

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + C_2}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2}, \quad (2.7)$$

sendo  $C_2 = (K_2L)^2$  e  $K_2 \ll 1$ . Finalmente, a imagem é normalizada com  $(\mathbf{x} - \mu_{\mathbf{x}})/\sigma_{\mathbf{x}}$  e  $(\mathbf{y} - \mu_{\mathbf{y}})/\sigma_{\mathbf{y}}$  para a comparação estrutural, definida como:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{\mathbf{xy}} + C_3}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + C_3}. \quad (2.8)$$

As equações Equação 2.5, Equação 2.7 e Equação 2.8 são combinadas para formar o índice:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = l(\mathbf{x}, \mathbf{y})^\alpha \cdot c(\mathbf{x}, \mathbf{y})^\beta \cdot s(\mathbf{x}, \mathbf{y})^\gamma, \quad (2.9)$$

em que  $\alpha > 0$ ,  $\beta > 0$  e  $\gamma > 0$  são parâmetros para ajustar a importância relativa dos componentes de comparação. Para simplificar, os autores definem  $\alpha = \beta = \gamma = 1$  e  $C_3 = C_2/2$ . Logo, é possível definir SSIM como:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\mathbf{xy}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2)}. \quad (2.10)$$

A medida SSIM mede a similaridade entre duas imagens, retornando um valor no intervalo  $[0,1]$ , sendo que o valor 1.0 é obtido no caso das imagens serem idênticas. Para ser utilizado como função de perda, a Equação 2.10 é modificada para indicar dissimilaridade e, portanto, obtém-se o complemento

$$\Delta\text{SSIM}(\mathbf{x}, \mathbf{y}) = 1 - \text{SSIM}(\mathbf{x}, \mathbf{y}). \quad (2.11)$$

## 2.4 XGBoost

XGBoost (CHEN; GUESTRIN, 2016) é um algoritmo de aprendizado supervisionado que utiliza o método *gradient boosted trees* (FRIEDMAN, 2001) para capturar dependências entre dados complexos e permitir a escalabilidade em conjuntos de dados grandes. A implementação do XGBoost está disponível em diversas linguagens de programação e possui código aberto. Seu impacto é amplamente reconhecido em inúmeras competições de aprendizado de máquina e mineração de dados (CHEN; GUESTRIN, 2016).

*Gradient tree boosting* (FRIEDMAN, 2001) é uma técnica utilizada em problemas de regressão e classificação para produzir um modelo preditivo via um *ensemble* de árvores preditivas fracas (*weak learners*) (FRIEDMAN, 2002) cuja decisão conjunta gera uma acurácia arbitrariamente alta no conjunto de treinamento (DUDA; HART; STORK, 2000). Um *weak learner* é definido como um classificador cuja acurácia é, no mínimo, levemente superior à aleatoriedade (DUDA; HART; STORK, 2000). Portanto, o objetivo de XGBoost é criar um *ensemble* de árvores preditivas, a classificação final para uma determinada amostra é dada pela soma das predições de cada árvore (CHEN; GUESTRIN, 2016).

Para um determinado conjunto de dados com  $n$  amostras e  $m$  atributos  $D = \{(\mathbf{x}_i, y_i)\}$ , onde  $|D| = n$ ,  $\mathbf{x}_i \in \mathbb{R}^m$ ,  $y_i \in \mathbb{R}$ , um *ensemble* de árvores utiliza um somatório com  $K$  funções para prever a saída

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^K f_k(\mathbf{x}_i), \quad f_k \in F, \quad (2.12)$$

em que  $F = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$  com  $q: \mathbb{R}^m \rightarrow T$ ,  $w \in \mathbb{R}^T$  é o espaço das árvores de regressão. A estrutura de cada árvore que mapeia um exemplo ao índice correspondente em uma folha é definido por  $q$ , e  $T$  é a

quantidade de folhas na árvore e cada  $f_k$  representa uma árvore independente  $q$  com folhas de pesos  $w$ . O conjunto de funções de um modelo é definido por

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2.13)$$

sendo  $l$  uma função de perda diferenciável que mede a diferença entre a predição  $\hat{y}_i$  e o rótulo verdadeiro  $y_i$ .  $\Omega$ , definido como  $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ , é um termo de regularização do modelo (CHEN; GUESTRIN, 2016).

## 2.5 Métricas de Avaliação

Tabela 1 – Matriz de confusão para classificação binária.

		Classe predita	
		+	-
Classe verdadeira	+	VP	FN
	-	FP	VN

Fonte: (FACELI et al., 2021).

Este trabalho trata o problema da classificação binária cujo objetivo é identificar se um paciente possui, classe positiva (+), ou não, classe negativa (-), a DA. Dessa forma, os possíveis resultados de uma determinada classificação são apresentados por meio de uma matriz de confusão na Tabela 1, em que (FACELI et al., 2021):

- Verdadeiros Positivos (VP) corresponde ao número de exemplos da classe positiva classificados corretamente.
- Verdadeiros Negativos (VN) corresponde ao número de exemplos da classe negativa classificados corretamente.
- Falsos Positivos (FP) corresponde ao número de exemplos da classe negativa classificados como pertencentes à classe positiva.
- Falsos Negativos (FN) corresponde ao número de exemplos da classe positiva classificados como pertencentes à classe negativa.

A análise de desempenho do preditor XGBoost é realizada quantitativamente através das seguintes métricas obtidas a partir da matriz de confusão definida na Tabela 1, com  $N = VP + FP + FN + VN$ :

Tabela 2 – Métricas para avaliar o desempenho do classificador.

Métrica	Equação	Intervalo
Acurácia Total	$acc(\hat{f}) = \frac{VP+VN}{N}$	[0, 1]
Precisão	$prec(\hat{f}) = \frac{VP}{VP+FP}$	[0, 1]
Sensibilidade ou Revocação	$revc(\hat{f}) = \frac{VP}{VP+FN}$	[0, 1]
$F_1$	$F_1(\hat{f}) = \frac{2 \times prec(\hat{f}) \times revc(\hat{f})}{prec(\hat{f}) + revc(\hat{f})}$	[0, 1]

Fonte: (FACELI et al., 2021).

As métricas da Tabela 2 serão utilizadas para a avaliação dos resultados para comparar quantitativamente os diferentes autoencoders.

## 2.6 Revisão Bibliográfica

Utilizando como critério as diferentes formas de utilizar imagens de RM, três abordagens distintas são identificadas.

A primeira abordagem consiste em realizar a pré-extração de atributos antes da utilização com autoencoders. Em SUK, LEE e SHEN (2015), os autores utilizam 93 regiões de interesse calculando o volume da substância cinzenta para imagens de RM e média de intensidade de imagens de PET-*scan* como atributos de entrada para um autoencoder empilhado e posterior classificação nas classes CN×DA, CN×CCL e CCL×DA com um *Multi-kernel* Support Vector Machine (SVM). Com uma abordagem similar, FERRI et al. (2021) utilizaram 42 métricas extraídas de exames de eletroencefalograma (EEG) obtidos com o paciente em repouso e 16 biomarcadores, como volume intracranial total e volume de substância cinzenta em dois tipos de autoencoders: (i) autoencoder empilhado com uma camada *softmax* para classificação; (ii) dois autoencoders especializados nas classes CN e DA com classificação via comparação do erro de reconstrução das imagens.

A segunda abordagem utiliza imagens bidimensionais de RM obtidas a partir do recorte de um plano (axial, coronal ou sagital) e seleção de uma fatia. Utilizando esse processo, YAGIS, SECO e CITI (2021) realizaram a extração de características com autoencoders convolucionais empilhados e classificação via RNA para as classes CN e DA.

A terceira abordagem utiliza imagens tridimensionais de RM. O benefício dessa abordagem é a possibilidade de capturar relações e estruturas locais que não são encontradas em imagens 2D. A partir dessa premissa, PAYAN e MONTANA (2015) utilizaram *overcomplete* autoencoders esparsos treinados em imagens 3D de RM com *patches* aleatórios de  $5 \times 5 \times 5$  voxels e 150 nós na camada escondida. Os pesos desta camada foram transformados em filtros 3D para as operações de convolução. Com isso, esses filtros são adicionados à uma RNA com três camadas treinada anteriormente com gradiente descendente para a classificação nas classes CN, CCL e DA. Os autores também apresentam uma versão modificada do modelo para classificação de imagens 2D. HEDAYATI, KHEDMATI e GORJIKOLAIE (2021) também utilizaram imagens 3D de RM e extração de características com autoencoders convolucionais. Para isso, os autores treinaram onze diferentes autoencoders, cada um representando uma imagem *template* diferente que considera pequenas alterações inerentes aos equipamentos e posicionamento dos pacientes durante o processo de aquisição das imagens. Os atributos extraídos foram utilizados como entrada em uma RNA convolucional para classificação entre as classes, CN×DA, CN×CCL e CCL×DA.

Em suma, os resultados obtidos pelos artigos citados é apresentado na Tabela 3.

Tabela 3 – Comparação entre os principais resultados obtidos pelas abordagens.

Referência	Dados de RM	Modelo	Acurácia (%) CN × DA
SUK, LEE e SHEN (2015)	93 atributos de regiões de interesse	AE e <i>Multi-kernel</i> SVM	$98.8 \pm 0.4$
FERRI et al. (2021)	16 atributos de regiões de interesse	(i) AE com <i>softmax</i> (ii) AE especializado com erro de reconstrução	(i) $84.3 \pm 6.9$ (ii) $82.4 \pm 7.3$
YAGIS, SECO e CITI (2021)	Imagens 2D	ACE e Rede Neural Artificial	$74.66 \pm 0.01$
PAYAN e MONTANA (2015)	Imagens 2D e 3D	AE e Rede Neural Convolucional	(2D e 3D) $95.39$
HEDAYATI, KHEDMATI e GORJIKOLAIE (2021)	Imagens 3D	Onze ACE especializados por <i>template</i> e Rede Neural Convolucional	95

Fonte: elaborada pelo autor.



## 3 Metodologia

Este capítulo apresenta a metodologia adotada neste trabalho de conclusão de curso de graduação.

### 3.1 Visão Geral

O desenvolvimento deste trabalho é resumido nas etapas seguintes:

1. Após a etapa de pré-processamento e *data augmentation*, obtém-se como entrada 25,518 imagens 2D de RM da região hipocampal com 23,976 pertencentes à classe CN e 1,542 à DA.
2. Seleção de 154 imagens de cada classe, aproximadamente 10% da classe DA, para validação dos hiperparâmetros de XGBoost.
3. Divisão dos dados da classe CN em  $k$  partições para treinamento e teste com validação cruzada.
4. Treinamento dos autoencoders especializados na classe CN com  $k - 1$  partições.
5. A partição restante é concatenada com as imagens da classe DA para realizar uma nova validação cruzada estratificada.
6. Treinamento do classificador XGBoost em ambas as classes.
7. Classificação com XGBoost nas classes CN e DA com a partição de teste.
8. Avaliação dos resultados com as métricas acurácia, *f1-score*, precisão e revocação.

### 3.2 Imagens

As imagens utilizadas neste projeto já foram pré-processadas e passaram por uma etapa de *data augmentation* em um trabalho de Iniciação Científica do bolsista Paulo Henrique Dal Bello (FAPESP, processo: 19/08553-5) orientado pelo Prof. Dr. Ricardo José Ferrari. Embora as etapas mencionadas não foram realizadas neste projeto, o processo será descrito nesta seção para fins de documentação.

#### 3.2.1 Bases de Imagens

As imagens tridimensionais de RM simuladas e reais de cérebros humanos foram obtidas das seguintes bases de imagens.

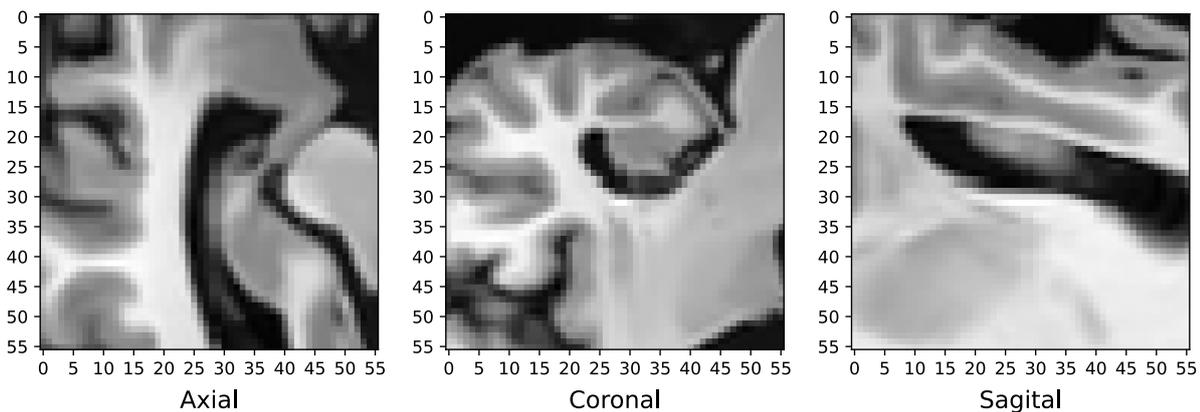
1. **IXI:** Imagens clínicas (total de 581 imagens de RM; T1-w, T2-w, e PD) do bando de imagens de domínio público do “*Biomedical Image Analysis Group, Imperial College London*”, intitulada “*IXI database*”;
2. **ADNI e EADC-ADNI:** Imagens clínicas de pacientes com e sem (grupo CN) a doença de Alzheimer obtidas do projeto “*The Alzheimer’s Disease Neuroimaging Initiative*” (ADNI). A base de imagens ADNI possui um total de 727 imagens T1-w de RM (313 pacientes ECS, 176 com CCL e 238 com DA), sendo que 135 delas também compõem a base EADC-ADNI, que possui casos de CCL e CLL tardio.
3. **NAC:** Atlas do cérebro desenvolvido pelo “*Neuroimage Analysis Center*”, contendo imagens de RM ponderadas em T1-w e T2-w como padrões e malhas poligonais de 149 estruturas cerebrais.

### 3.2.2 Pré-Processamento

Imagens de RM são comumente afetadas por diferentes tipos de artefatos que devem ser minimizados antes da utilização das imagens técnicas quantitativas (MANJÓN, 2017). Com isso, a primeira etapa de pré-processamento é a redução de ruído aleatório inerente do processo de aquisição da imagem (MANJÓN, 2017) pelo uso do algoritmo *Non-LocalMeans* (NLM) (BUADES; COLL; MOREL, 2011). Em seguida, é realizada a correção do *bias field*, uma variação de intensidade não uniforme causada por variações no campo magnético, com o algoritmo N4ITK (TUSTISON et al., 2010). A terceira etapa consiste em padronizar os histogramas com uma técnica específica para imagens de RM (NYUL; UDUPA; XUAN, 2000).

Finalmente, é necessário selecionar a região hipocampal nas imagens. Para isso, as imagens são alinhadas a um mesmo sistema de coordenadas base em uma imagem *template*, esse processo é feito pelo software NiftyReg, que implementa o algoritmo Aladin (OURSELIN et al., 2001). Em seguida, as regiões hipocampais são extraídas usando como guia as máscaras dos hipocampos que acompanham a imagem de referência da base NAC, resultando em uma imagem com  $56 \times 56$  voxels. A última etapa consiste em obter o recorte das fatias centrais da região hipocampal (plano axial, coronal e sagital) com o auxílio da biblioteca NiBabel. Dessa forma, cada imagem tridimensional de RM é transformada em três imagens bidimensionais, uma para cada plano, com  $56 \times 56$  pixels, vide Figura 3.

Figura 3 – Três imagens bidimensionais extraídas de uma imagem tridimensional de RM.



Fonte: elaborada pelo autor.

### 3.2.3 Data Augmentation

A maneira mais simples de generalizar um modelo de aprendizado de máquina é treiná-lo em mais dados. No entanto, quando aumentar a quantidade de amostras não é possível, torna-se necessário a criação de dados falsos a partir de variações nos dados originais, este processo é denominado *data augmentation* (GOODFELLOW; BENGIO; COURVILLE, 2016). Assim, após a etapa de pré-processamento, esse processo foi aplicado às 1998 amostras por corte (axial, coronal e sagital) para obtenção de 23.982 amostras por corte. Foram aplicadas as seguintes transformações nos dados originais para aumento de 1998 :

1. Espelhamento: com base na característica cerebral de simetria entre hemisférios, foi realizado o espelhamento das imagens dos hipocampos esquerdos para dobrar o número de imagens.
2. *Bias field*: para geração de novas imagens, cada imagem da base teve uma versão corrompida por um *bias field* retirado e corrigido das imagens na etapa de pré-processamento. O novo *bias field* foi escolhido a partir de uma função pseudorrandômica.

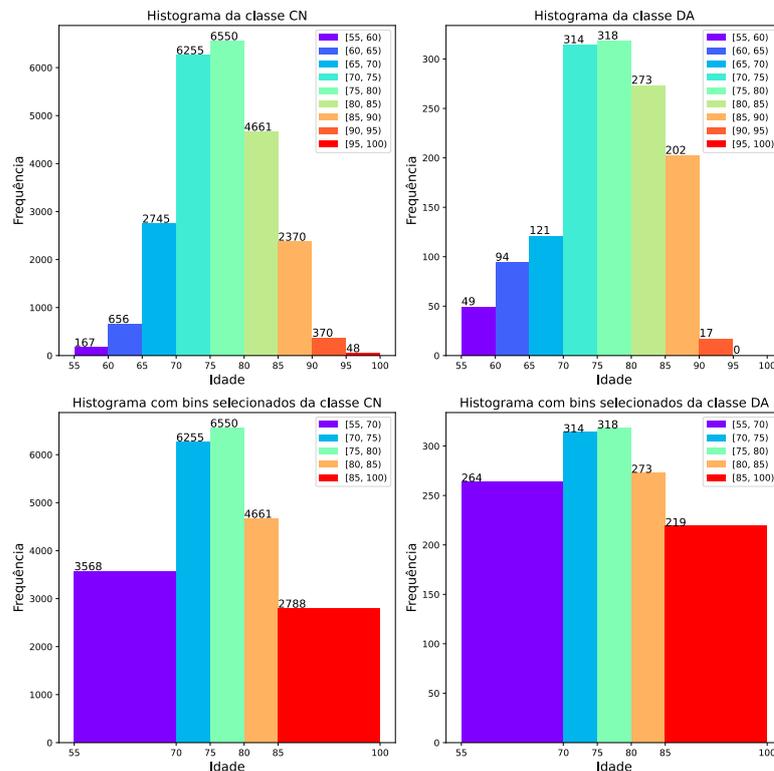
3. Ruído térmico: uma distribuição Riciana (obtida através da aproximação de Gaussianas) foi adicionada às imagens.
4. Rotação: 5° no sentido horário ou anti-horário aleatoriamente.
5. Translação: 10% em ambos os eixos nos sentidos direita-baixo, direita-cima, esquerda-baixo e esquerda-cima.
6. Escala: aumento ou diminuição, escolhidos aleatoriamente, em 10% da escala original.

### 3.2.4 Estratificação por Idade

Devido ao aumento exponencial na incidência de demência (seção 2.1) torna-se necessário maior atenção aos pacientes com idade avançada. Ademais, o envelhecimento é um dos fatores mais impactantes no processo natural de neurodegeneração (HOU et al., 2019). Dessa forma, a estratificação por idade é necessária porque existe uma modificação estrutural natural que ocorre com o envelhecimento e se isso não for considerado, os resultados podem ser enviesados. Ao realizar o imageamento por RM, a idade do paciente é um fator que deve ser levado em consideração ao realizar a análise dos biomarcadores por imagem. Assim, o conjunto original de dados também será avaliado com uma versão estratificada por faixa etária. A Figura 4 apresenta a distribuição inicial dos dados.

O fator limitante para realizar a estratificação é a quantidade de amostras da classe DA e, portanto, os *bins* selecionados para separar o conjunto de dados busca balancear essa classe. Os seguintes *bins* foram definidos para estratificar os dados: [55, 70), [70, 75), [75, 80), [80, 85) e [85, 100).

Figura 4 – Histograma inicial e modificado para as classes CN e DA.



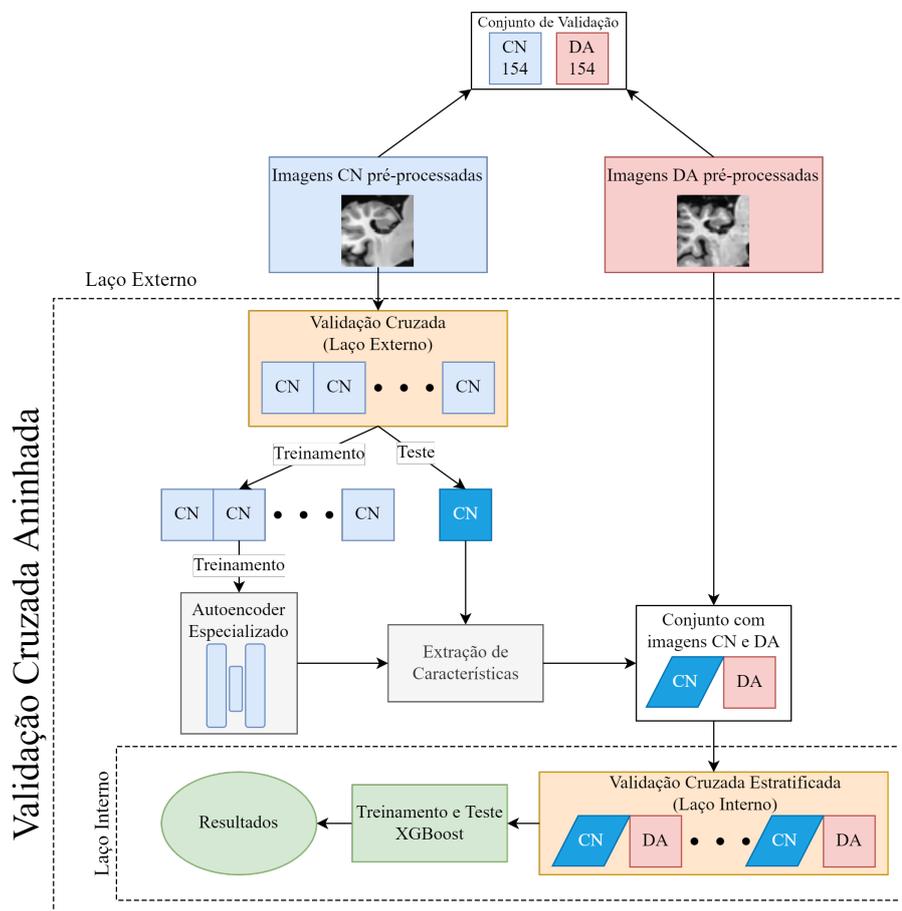
Fonte: elaborada pelo autor.

### 3.3 Validação Cruzada Aninhada

Para obter estimativas de desempenho preditivo mais confiáveis, os dados foram divididos em três conjuntos distintos: validação, treinamento e teste. O conjunto de validação é utilizado para estimar os diversos parâmetros do classificador XGBoost, sendo que as amostras desse conjunto foram selecionados com amostragem aleatória simples sem reposição, e com todas as amostras removidas do conjunto total. Os conjuntos de treinamento e teste, utilizados pelos autoencoders e XGBoost, foram definidos através da amostragem via validação cruzada aninhada.

A primeira etapa da validação cruzada aninhada, vide Figura 5, consiste em separar todas as imagens do conjunto CN em  $k$  partições, para treinamento dos autoencoders com  $k-1$  partições (validação cruzada no laço externo). A partição restante é concatenada com as amostras da classe DA para treinamento e teste do classificador XGBoost com validação cruzada estratificada (laço interno) em  $r$  partições. Dessa forma, foram obtidos  $k \times r$  resultados com a validação cruzada aninhada. O resultado final é dado pela média aritmética e desvio padrão de todas as execuções.

Figura 5 – Resumo ilustrativo das etapas de treinamento e teste dos autoencoders e do XGBoost.



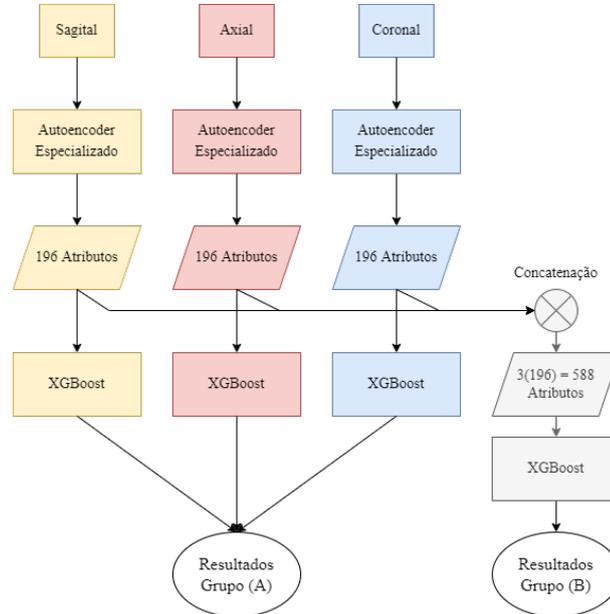
Fonte: elaborada pelo autor.

### 3.4 Arquitetura dos Autoencoders

Este trabalho utiliza três tipos de autoencoders para a extração de atributos. Todos os modelos apresentam em comum o treinamento *layer-wise* e redução da imagem original de  $56 \times 56$  pixels para 196 atributos na etapa de codificação e posterior decodificação e remodelagem (do inglês, *reshape*) para mesma dimensão de entrada. Logo, embora os modelos sejam diferentes, todos produzem um vetor de

196 atributos no *bottleneck* que será utilizado como entrada para o classificador. As arquiteturas dos autoencoders utilizadas neste trabalho são apresentadas nesta seção.

Figura 6 – Aquisição de resultados para autoencoders especializados por plano.



Fonte: elaborada pelo autor.

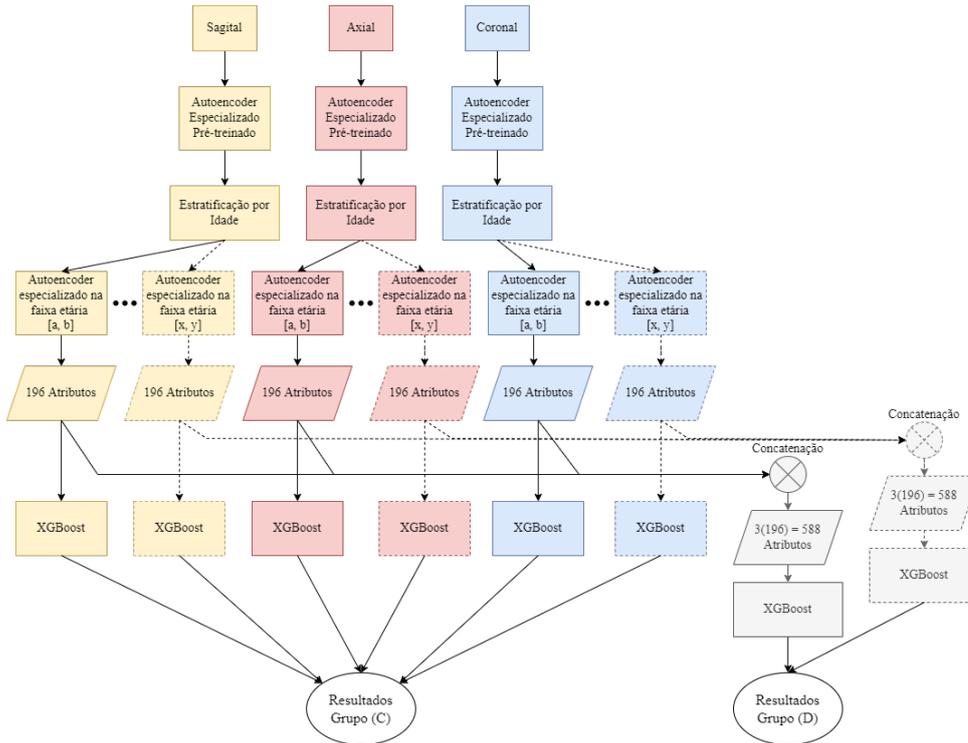
A fim de comparar as arquiteturas propostas, para cada modelo, quatro grupos distintos foram utilizados para aquisição dos dados. Os dois primeiros grupos, exemplificados na Figura 6, realizam o treinamento de três autoencoders especializados por planos (axial, coronal e sagital). No método (A), três classificadores são treinados e avaliados, um para cada autoencoder especializado. No método (B), os atributos extraídos são concatenados num vetor de 588 atributos para posterior classificação e avaliação com XGBoost.

De maneira similar, os grupos (C) e (D) apresentados na Figura 7 realizam a classificação dos atributos extraídos para cada plano (C) e, também, concatenação (D), todavia os dados são estratificados por idade e, para cada plano, os autoencoders são também especializados em uma determinada faixa etária. A fim de evitar *overfitting*, devido ao número reduzido de amostras após estratificação, a estratégia de aprendizado por transferência é utilizada. Para isso, os pesos dos autoencoders especializados por planos (Figura 6) são utilizados para iniciar os pesos dos novos autoencoders especializados por faixa etária. Dessa forma, o aprendizado desses autoencoders tem como objetivo realizar um ajuste preciso dos pesos das redes (*fine-tuning*).

### 3.4.1 Autoencoder Empilhado com MSE

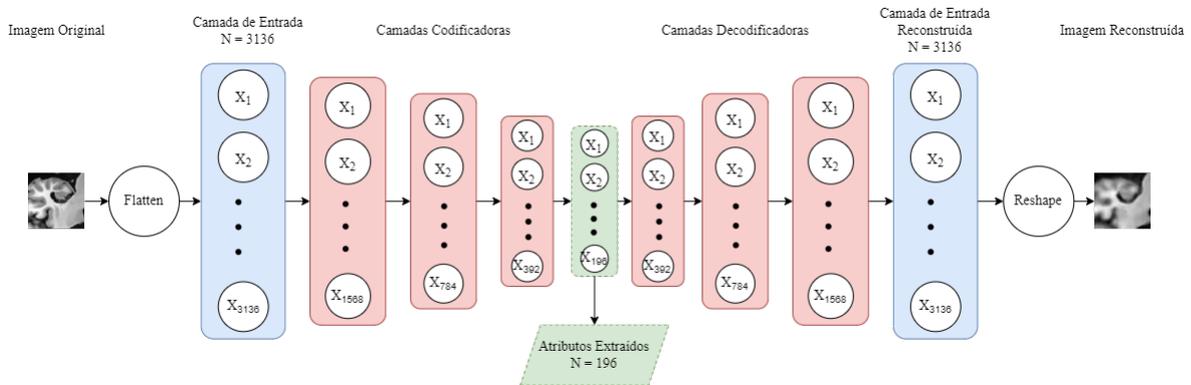
Este é o modelo mais simples e apresenta quatro camadas totalmente conectadas para codificação. A imagem de entrada de  $56 \times 56$  pixels é achatada (do inglês, *flattened*) em um vetor com 3136 atributos, seguida pelas seguintes reduções na quantidade de nós em cada camada:  $1568 \rightarrow 784 \rightarrow 392 \rightarrow 196$ . A etapa de decodificação consiste no processo reverso à codificação, i.e., a última camada possui 3136 nós e uma remodelação para  $56 \times 56$  pixels, vide Figura 8.

Figura 7 – Aquisição de resultados para autoencoders especializados por plano e faixa etária.



Fonte: elaborada pelo autor.

Figura 8 – Arquitetura do Autoencoder Empilhado com função de perda MSE.

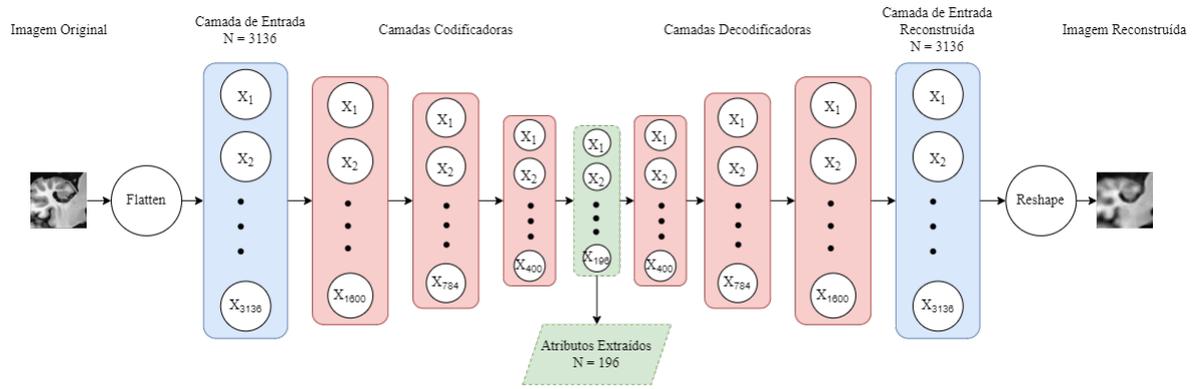


Fonte: elaborada pelo autor.

### 3.4.2 Autoencoder Empilhado com SSIM

Este modelo possui uma arquitetura similar ao anterior mas implementa a função de perda com SSIM. Para isso, é necessário modificar o modelo apresentado na subseção 3.4.1 para que a quantidade de nós em cada camada possa formar uma imagem quadrada. Portanto, no modelo da subseção 3.4.1, a quantidade de nós nas camadas de codificação é alterada de  $1568 \rightarrow 784 \rightarrow 392 \rightarrow 196$  para  $1600 \rightarrow 784 \rightarrow 400 \rightarrow 196$ . A camada de decodificação realizará o processo reverso com a mesma quantidade de nós por camada da codificação, vide Figura 9.

Figura 9 – Arquitetura do Autoencoder Empilhado com função de perda SSIM.



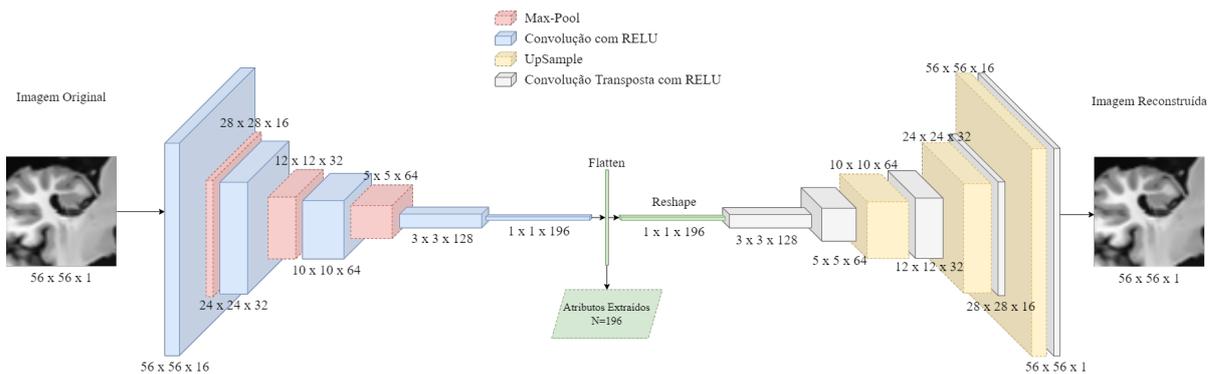
Fonte: elaborada pelo autor.

### 3.4.3 Autoencoder Empilhado Convolucional

É esperado que um ACE capture relações locais entre pixels e construa filtros para o reconhecimento de padrões nas imagens, tarefa que não seria possível para um modelo com camadas totalmente conectadas. Assim, um ACE com camadas de convolução e *pooling* é implementado tal que a quantidade de atributos na camada *bottleneck* é igual aos modelos da seção subseção 3.4.1 e subseção 3.4.2.

A etapa de codificação deste modelo possui três camadas de convolução e *max-pooling*, seguidas por duas camadas de convolução, vide Figura 10. O objetivo desta etapa é reduzir o tamanho da imagem de entrada via *pooling* e aumentar a quantidade de filtros a cada convolução. Com isso, a imagem de entrada de  $56 \times 56$  pixels é transformada para um único vetor de 196 atributos. Este autoencoder utiliza a função de perda MSE, visto que não seria possível aplicar a métrica SSIM devido à baixa dimensão das imagens após as operações de *pooling*.

Figura 10 – Arquitetura do Autoencoder Convolucional Empilhado.



Fonte: elaborada pelo autor.

## 3.5 Hiperparâmetros do Classificador

Testes preliminares indicaram alta tendência de *overfitting* para o classificador XGBoost que apresentava elevada acurácia no conjunto de treinamento e baixa acurácia no conjunto de validação. Com isso, três estratégias foram empregadas: *early stopping rounds*; redução na complexidade do modelo via ajuste dos parâmetros *max\_depth*, *min\_child\_weight* e *gamma*; adição de randomização para aumentar a resistência do modelo a ruídos com ajuste dos parâmetros *subsample* e *colsample\_bytree*. A estratégia *early stopping rounds* consiste em parar o treinamento do modelo caso a acurácia no conjunto de validação deixe de diminuir após uma determinada quantidade de iterações. Esse valor mínimo para encerrar

o treinamento foi determinado a partir de uma análise visual das curvas de acurácia do conjunto de treinamento e validação. Os parâmetros de complexidade são definidos como:

- **max\_depth**: profundidade máxima das árvores (valor padrão: 6). Quanto maior este valor, mais complexo será o modelo final.
- **min\_child\_weight**: valor mínimo necessário no somatório de pesos das folhas de uma árvore (valor padrão: 1). Quanto maior este valor, mais conservador será o modelo final.
- **gamma**: termo de regularização que define a perda mínima necessária para realizar uma nova divisão em uma folha (valor padrão: 0). Quanto maior este valor, mais conservador será o modelo final.

Os parâmetros de randomização são definidos como:

- **subsample**: razão (*ratio*) de elementos amostrados por iteração de treinamento (padrão: 1).
- **colsample\_bytree**: razão de colunas (atributos) amostrados por iteração (padrão: 1).

Com base em testes realizados no conjunto de validação, os parâmetros **max\_depth** e **subsample** foram fixados em 3 e 0.5, respectivamente. Os valores dos demais parâmetros foram determinados via *Grid Search*. Essa operação consiste em determinar para cada parâmetro desconhecido  $p$  um conjunto  $F_p$  cujos elementos são valores candidatos à escolha, e.g., o conjunto do parâmetro **gamma** foi determinado como  $F_{\text{gamma}} = \{2.5, 5, 10\}$ . Em seguida, todos conjuntos são unidos via produto cartesiano formando um novo conjunto  $F'$ , i.e.,  $F' = F_{p_1} \times F_{p_2} \times \dots \times F_{p_i}$ , cujas tuplas ordenadas são definidas como valores dos parâmetros candidatos  $p$ . Somado a isso, a operação de amostragem via validação cruzada é implementada para tornar os resultados menos dependentes da partição inicial dos dados. Dessa forma, a operação *Grid Search* executa o treinamento e teste do XGBoost  $\text{card}(F') \times K$  vezes, onde  $\text{card}(F')$  é a cardinalidade do conjunto  $F'$  e  $K$  é a quantidade de partições da validação cruzada. Finalmente, a tupla ordenada de  $F'$  com melhores resultados para uma determinada métrica escolhida previamente é utilizada para definir os valores finais de cada parâmetro.

Os grupos (A) e (C) utilizam um vetor de 196 atributos como entrada para o classificador. Nesse caso, os seguintes valores apresentados na Tabela 4 foram selecionados como candidatos para *Grid Search*.

Tabela 4 – Valores candidatos para *Grid Search* para os grupos (A) e (C).

Parâmetros	Valores Candidatos
<b>min_child_weight</b>	$F_{\text{mcw}} = \{2.5, 5\}$
<b>gamma</b>	$F_{\text{gamma}} = \{0, 1, 5\}$
<b>colsample_bytree</b>	$F_{\text{bytree}} = \{0.3, 0.5\}$

Fonte: elaborada pelo autor.

Já os grupos (B) e (D) realizam a concatenação dos vetores para obter 588 atributos como entrada para o classificador. Logo, devido ao aumento de complexidade dos dados, tornou-se necessário reduzir a complexidade do modelo. Para isso, foram utilizados os valores da Tabela 5.

Tabela 5 – Valores candidatos para *Grid Search* para os grupos (B) e (D).

Parâmetros	Valores Candidatos
<b>min_child_weight</b>	$F_{\text{mcw}} = \{2.5, 5, 10\}$
<b>gamma</b>	$F_{\text{gamma}} = \{2.5, 5, 10\}$
<b>colsample_bytree</b>	$F_{\text{bytree}} = \{0.3, 0.5\}$

Fonte: elaborada pelo autor.

## 4 Resultados e Discussão

Este capítulo apresenta os resultados obtidos durante o desenvolvimento do projeto.

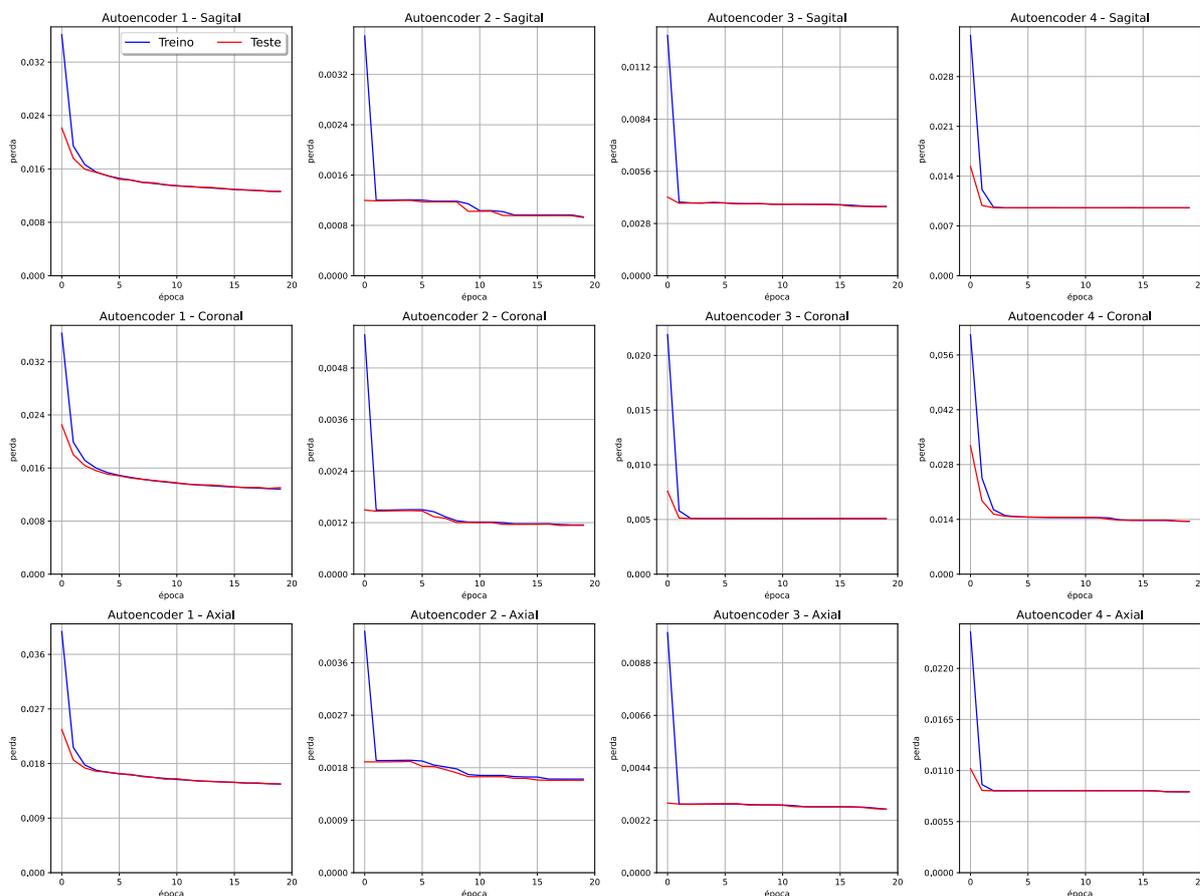
### 4.1 Treinamento dos Autoencoders

O objetivo desta seção é avaliar as curvas da função de perda para os conjuntos de treinamento e teste. Analisando as curvas nas Figuras 4.3, 4.6 e 4.9 é possível observar que em todas as métricas, os autoencoders especializados por faixa etária apresentaram os melhores resultados, e o ACE obteve os menores de perda.

#### 4.1.1 AE com função de perda MSE

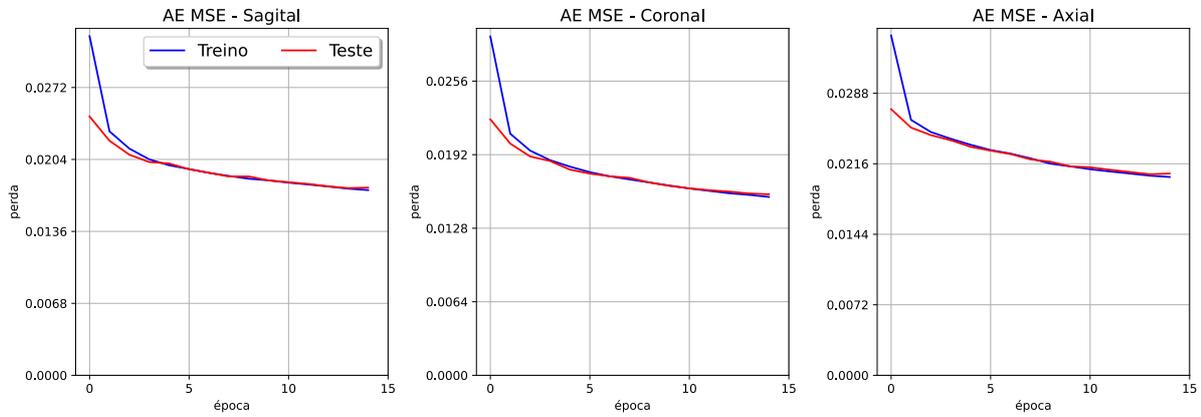
A Figura 11 apresenta as curvas de perda para os autoencoders treinados com a função de perda MSE antes de serem empilhados. O processo de *fine-tuning* das camadas empilhadas é apresentado na Figura 12. O aprendizado por transferência para treinamento dos autoencoders especializados por faixa etária pode ser visto na Figura 13.

Figura 11 – Autoencoders treinados por camada com função de perda MSE.



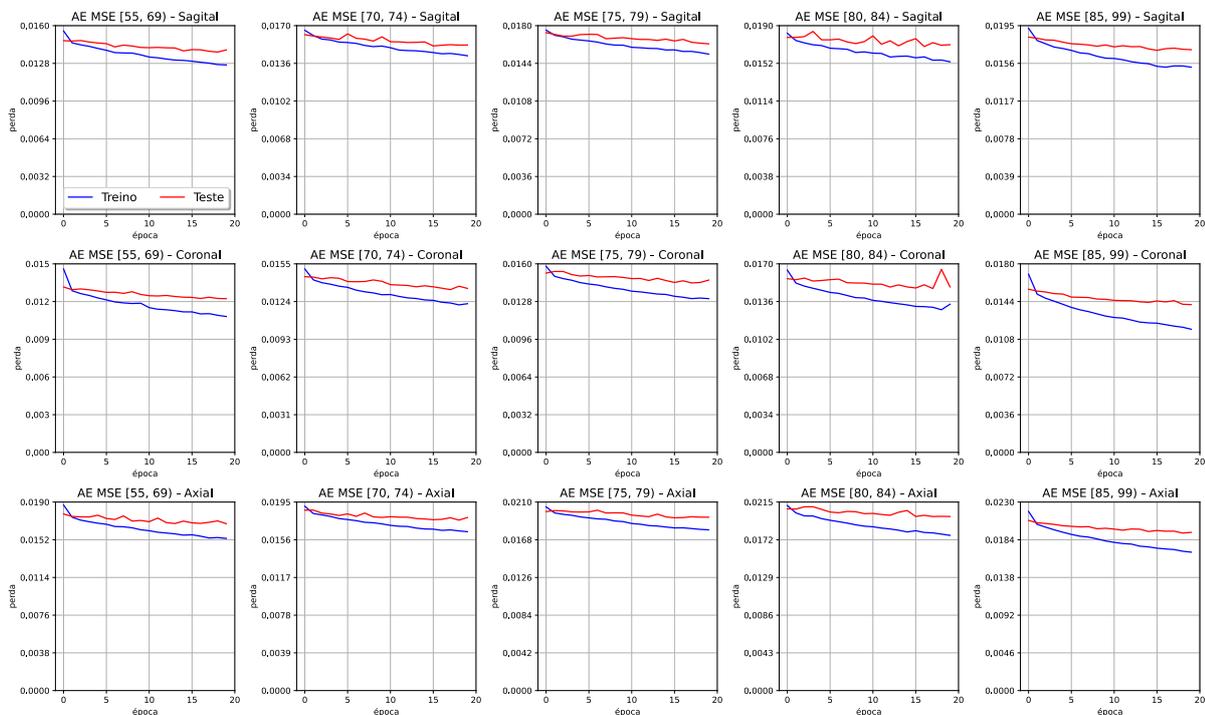
Fonte: elaborada pelo autor.

Figura 12 – *Fine-tuning* dos Autoencoders Empilhados com função de perda MSE.



Fonte: elaborada pelo autor.

Figura 13 – *Fine-tuning* dos Autoencoders Empilhados com estratificação por idade e função de perda MSE.

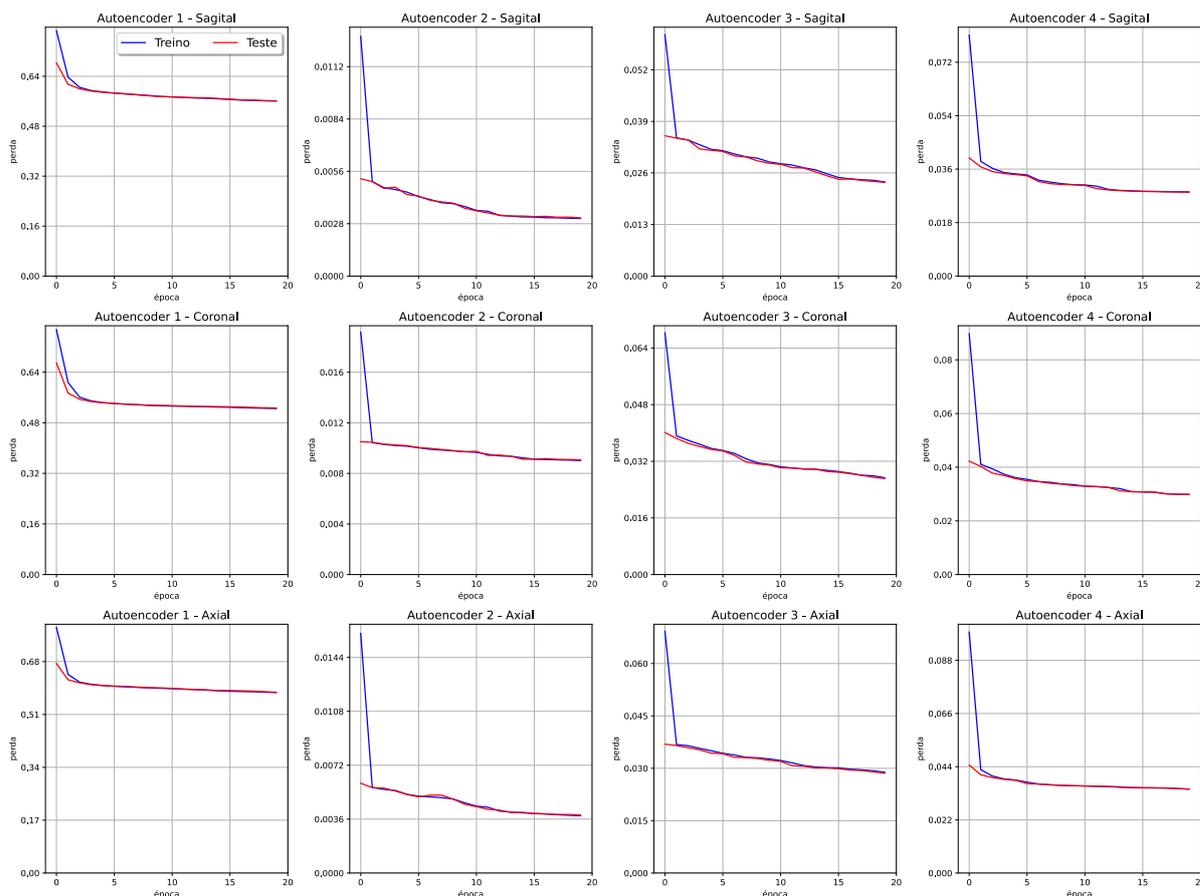


Fonte: elaborada pelo autor.

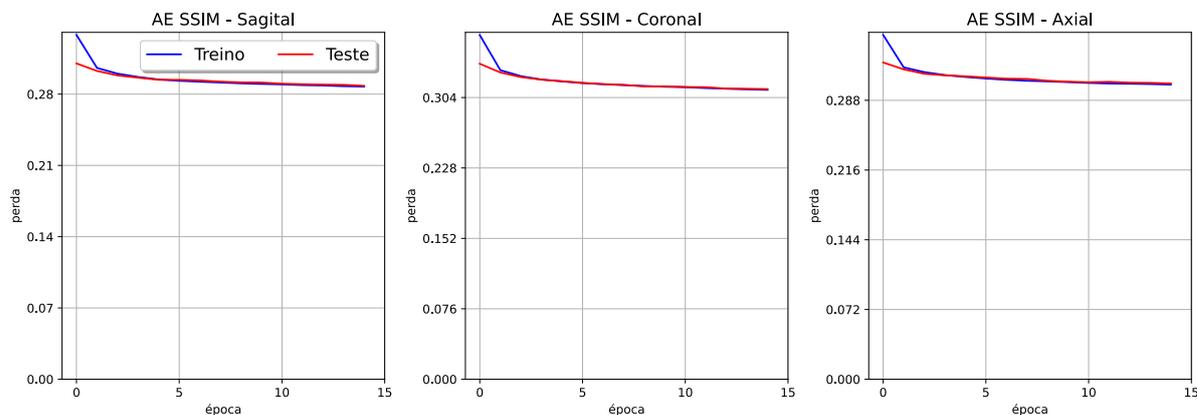
## 4.1.2 AE com função de perda SSIM

Os resultados da Figura 14 são, em geral, próximos aos valores obtidos da Figura 11. Dos doze gráficos apresentados, o AE com MSE obteve menor perda em 8 casos. O processo de *fine-tuning* (Figura 15) também apresentou vantagem para a função de perda com MSE, obtendo resultados superiores, embora próximos, para todos os três gráficos. O aprendizado por transferência (Figura 16) apresentou leve vantagem para a função MSE.

Figura 14 – Autoencoders treinados por camada com função de perda SSIM.

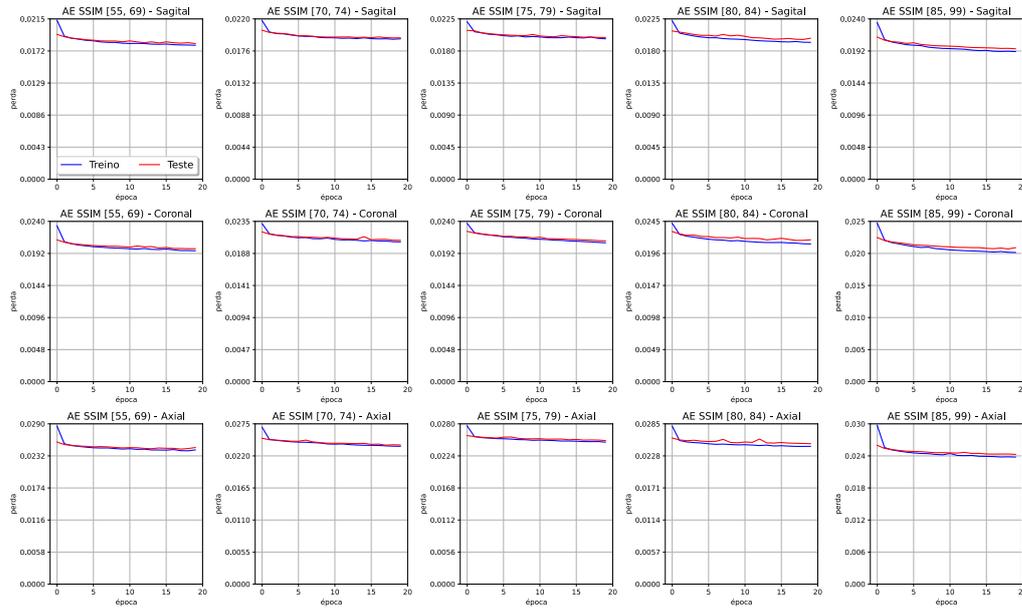


Fonte: elaborada pelo autor.

Figura 15 – *Fine-tuning* dos Autoencoders Empilhados com função de perda SSIM.

Fonte: elaborada pelo autor.

Figura 16 – *Fine-tuning* dos Autoencoders Empilhados com estratificação por idade e função de perda SSIM.

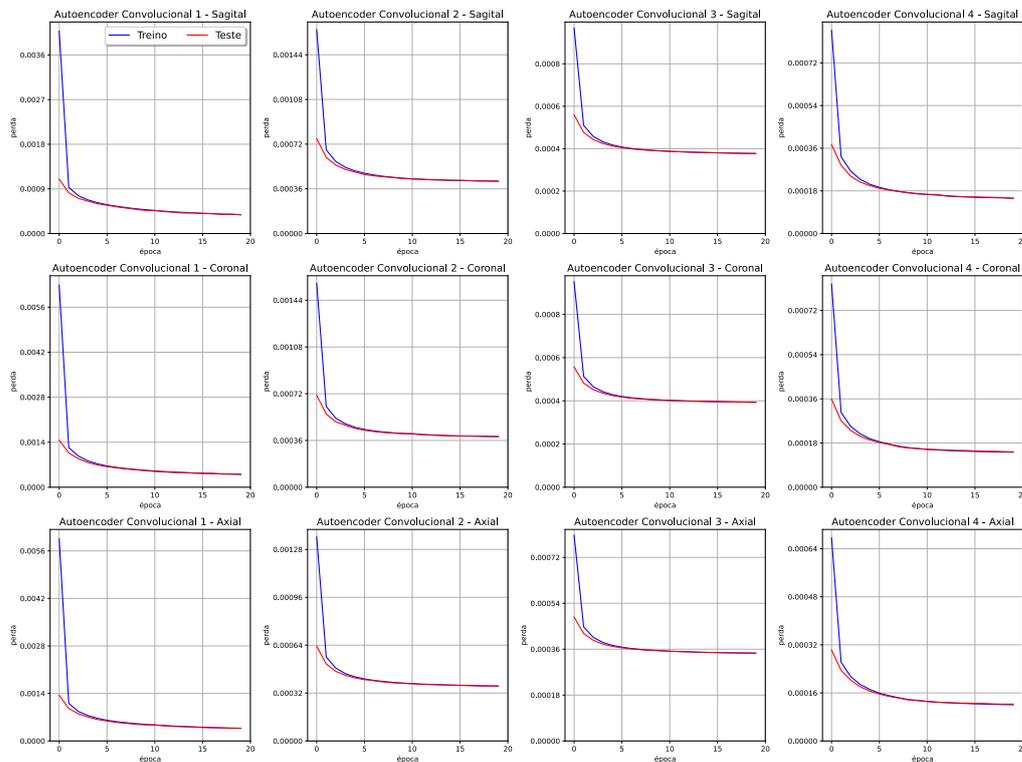


Fonte: elaborada pelo autor.

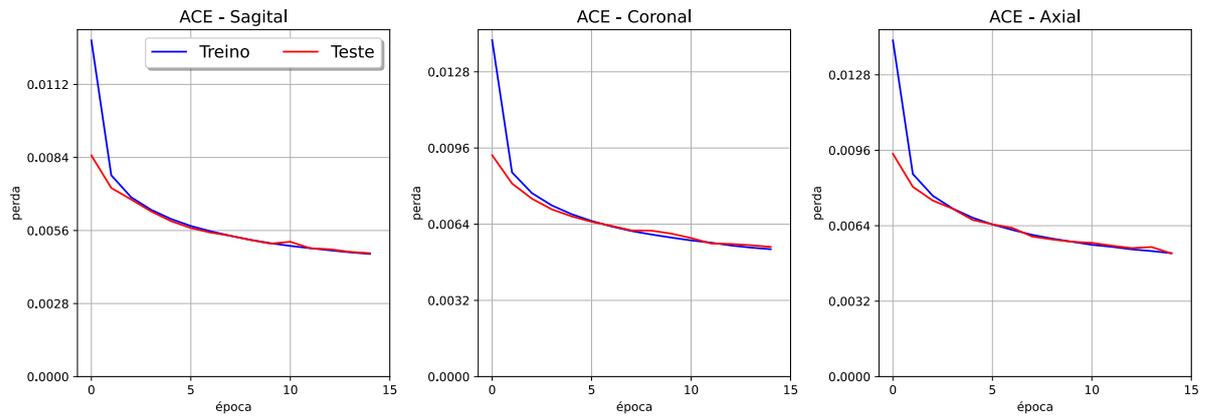
#### 4.1.3 ACE

Em todos os quesitos (Figura 17, Figura 18 e Figura 19), os autoencoders convolucionais empilhados obtiveram os melhores resultados.

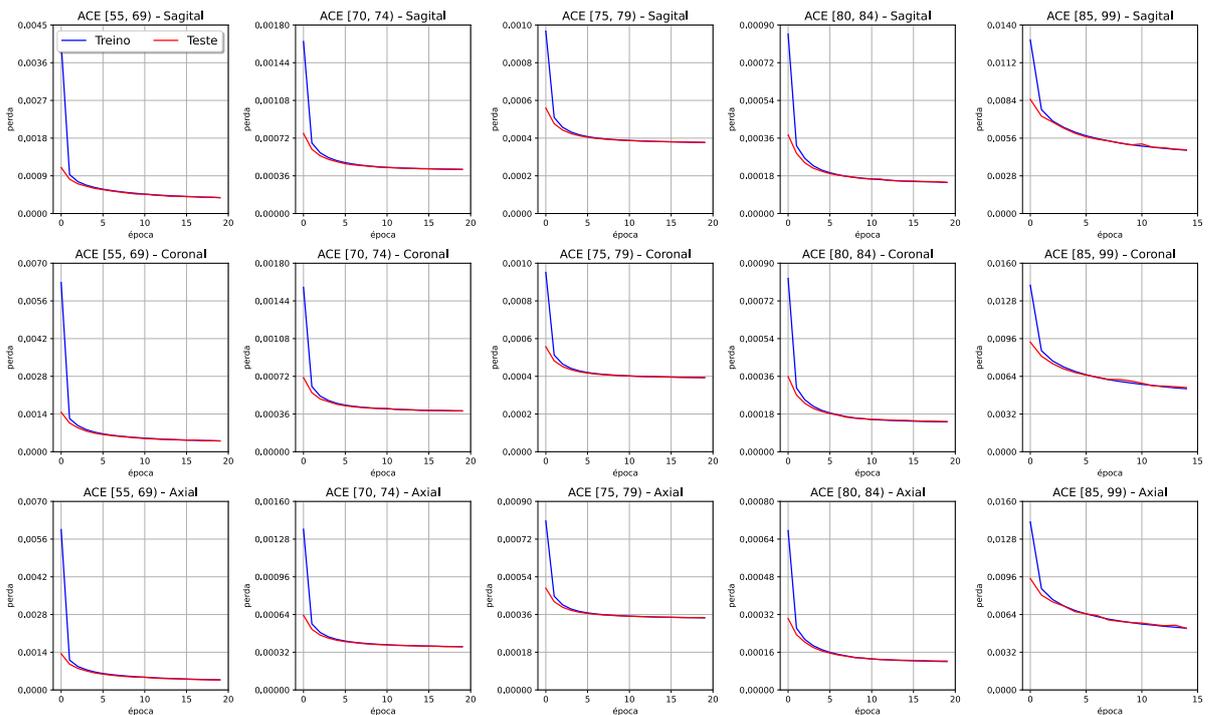
Figura 17 – Autoencoders Convolucionais treinados por camada.



Fonte: elaborada pelo autor.

Figura 18 – *Fine-tuning* dos Autoencoders Convolucionais Empilhados.

Fonte: elaborada pelo autor.

Figura 19 – *Fine-tuning* dos Autoencoders Convolucionais Empilhados com estratificação por idade.

Fonte: elaborada pelo autor.

## 4.2 Resultados da Classificação

Os autoencoders treinados na seção 4.1 foram utilizados para extração de características das imagens para um vetor de 196 atributos. Considerando as diversas formas de aquisição desse vetor apresentadas na seção 3.4, os resultados resumidos são apresentados na Tabela 6. Todas as arquiteturas foram capazes de classificar corretamente as imagens. Os melhores valores de acurácia,  $F_1$  e precisão foram obtidos pelo autoencoder convolucional empilhado especializado por plano e faixa etária com concatenação dos planos (D).

Tabela 6 – Resultados da classificação com XGBoost. Média geral para cada grupo.

Grupo	Modelo	Média (%)				
		$acc_{treino}$	$acc_{teste}$	$F_1$	$prec$	$revc$
(A)	AE MSE	76.4 ± 7.2	73.5 ± 5.7	76.3 ± 4.5	69.7 ± 6	<b>85 ± 6.5</b>
	AE SSIM	82.4 ± 3.6	76 ± 3.1	76.9 ± 3	74.2 ± 3.3	79.9 ± 4.1
	ACE	<b>89.6 ± 2.4</b>	<b>81 ± 2.8</b>	<b>81.4 ± 2.8</b>	<b>79.8 ± 3.2</b>	83.1 ± 3.8
(B)	AE MSE	84.3 ± 3.6	80.8 ± 3.1	82.2 ± 2.5	76.5 ± 3.9	<b>89.2 ± 3.4</b>
	AE SSIM	87.8 ± 2.6	81.6 ± 2.4	82.3 ± 2.2	78.9 ± 2.7	86.2 ± 3.2
	ACE	<b>91.3 ± 2.1</b>	<b>84.9 ± 2.1</b>	<b>85.2 ± 2</b>	<b>83.1 ± 2.6</b>	87.7 ± 3.1
(C)	AE MSE	80.4 ± 7.9	76.9 ± 6.9	79.7 ± 6.2	74.5 ± 7.4	86.4 ± 8.6
	AE SSIM	84 ± 5.5	79.4 ± 5.7	81.6 ± 5.3	77.1 ± 6.1	<b>87.2 ± 7.8</b>
	ACE	<b>91.5 ± 5.9</b>	<b>82.7 ± 5.6</b>	<b>84 ± 5.4</b>	<b>81.8 ± 6.3</b>	86.8 ± 7.9
(D)	AE MSE	87.5 ± 0.8	83.1 ± 1	84.1 ± 2.2	79.9 ± 1.6	89.5 ± 4
	AE SSIM	90 ± 1	84.4 ± 0.8	85.3 ± 1.8	81.8 ± 1	<b>89.6 ± 4.4</b>
	ACE	<b>96 ± 0.8</b>	<b>87.2 ± 1.3</b>	<b>87.4 ± 1.7</b>	<b>86.9 ± 2.4</b>	88.5 ± 4.8

Fonte: elaborada pelo autor.

As tabelas a seguir detalham os resultados obtidos na Tabela 6 para cada plano das imagens.

Tabela 7 – Resultados da classificação com XGBoost para o plano axial.

Grupo	Modelo	Média do Plano Axial (%)				
		$acc_{treino}$	$acc_{teste}$	$F_1$	$prec$	$revc$
(A)	AE MSE	74.6 ± 6.6	72.2 ± 5.6	75.2 ± 4.6	68.3 ± 5.4	<b>84.1 ± 6.9</b>
	AE SSIM	80.9 ± 3.8	73.9 ± 2.8	74.6 ± 2.6	72.7 ± 3.1	76.8 ± 3.6
	ACE	<b>88 ± 2.1</b>	<b>78.9 ± 2.5</b>	<b>79.3 ± 2.5</b>	<b>78 ± 3</b>	80.9 ± 3.7
(C)	AE MSE	78.3 ± 7.1	75.6 ± 6.7	78.6 ± 6	73.2 ± 7	<b>85.6 ± 8.5</b>
	AE SSIM	84 ± 5.5	79 ± 5.6	80.6 ± 5.3	76.3 ± 6.1	86 ± 7.9
	ACE	<b>90 ± 6</b>	<b>80.7 ± 5.5</b>	<b>82 ± 5.3</b>	<b>80.2 ± 6.1</b>	84.6 ± 8.2

Fonte: elaborada pelo autor.

Tabela 8 – Resultados da classificação com XGBoost para o plano coronal.

Grupo	Modelo	Média do Plano Coronal (%)				
		$acc_{treino}$	$acc_{teste}$	$F_1$	$prec$	$revc$
(A)	AE MSE	80.4 ± 5	76.4 ± 3.5	78.4 ± 2.4	72.9 ± 4.6	<b>85.3 ± 4.9</b>
	AE SSIM	82.3 ± 2.9	76.9 ± 2.9	78 ± 2.7	74.5 ± 3.2	81.9 ± 3.5
	ACE	<b>90.8 ± 2.1</b>	<b>81.8 ± 2.4</b>	<b>82.1 ± 2.3</b>	<b>80.9 ± 2.9</b>	83.5 ± 3.2
(C)	AE MSE	85.1 ± 7.4	79.4 ± 6.1	81.7 ± 5.4	77.2 ± 7.1	87.4 ± 7.7
	AE SSIM	82.2 ± 4.6	79 ± 5.7	81.3 ± 5.2	76.6 ± 6.1	87.2 ± 7.7
	ACE	<b>93.1 ± 5.6</b>	<b>83.8 ± 5.4</b>	<b>85.1 ± 5</b>	<b>82.5 ± 6.3</b>	<b>88.4 ± 7.1</b>

Fonte: elaborada pelo autor.

Tabela 9 – Resultados da classificação com XGBoost para o plano sagital.

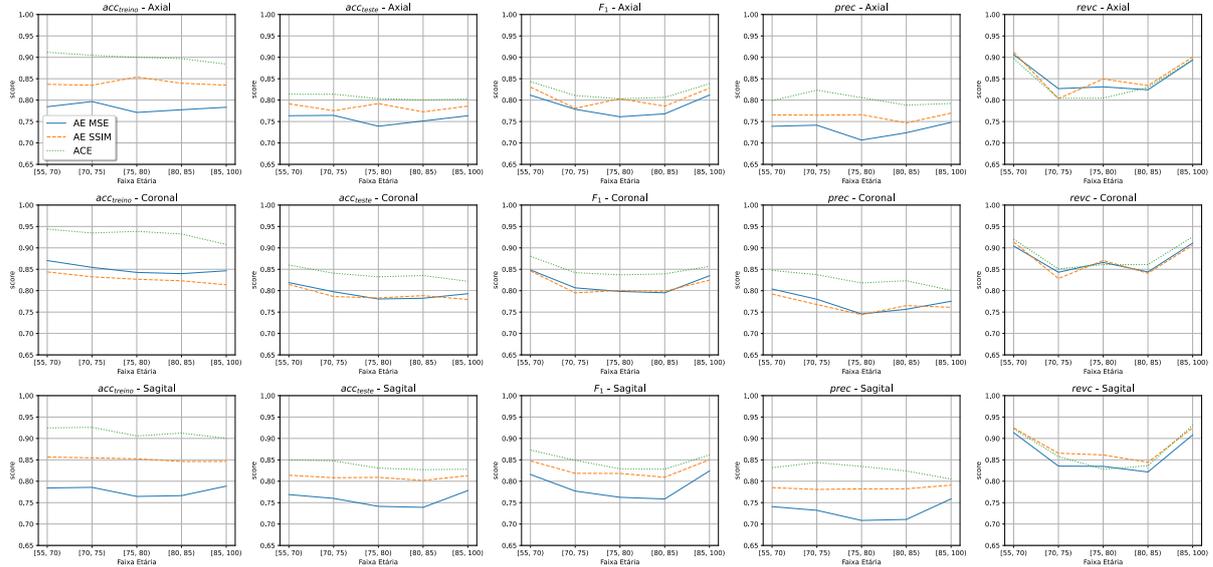
Grupo	Modelo	Média do Plano Sagital (%)				
		$acc_{treino}$	$acc_{teste}$	$F_1$	$prec$	$revc$
(A)	AE MSE	74.1 ± 7.7	71.9 ± 6.3	75.3 ± 5.1	67.9 ± 6.5	<b>85.5 ± 7.5</b>
	AE SSIM	84.1 ± 3.3	77.3 ± 2.6	78.1 ± 2.4	75.5 ± 3.1	81.1 ± 3.1
	ACE	<b>90 ± 2.1</b>	<b>81.8 ± 2.4</b>	<b>82.1 ± 2.3</b>	<b>80.9 ± 2.9</b>	83.5 ± 3.2
(C)	AE MSE	77.8 ± 7.1	75.7 ± 7.1	78.8 ± 6.7	73 ± 7.4	86.3 ± 9.5
	AE SSIM	85.1 ± 5.4	80.9 ± 5.3	82.9 ± 5	78.4 ± 5.9	<b>88.4 ± 7.6</b>
	ACE	<b>91.4 ± 5.5</b>	<b>83.7 ± 5.4</b>	<b>84.8 ± 5.2</b>	<b>82.8 ± 6.1</b>	87.5 ± 7.9

Fonte: elaborada pelo autor.

Tendo em vista a neurodegeneração natural e aumento de casos de demência em pessoas de idade avançada, torna-se necessário que o classificador seja capaz de manter uma boa acurácia com o aumento

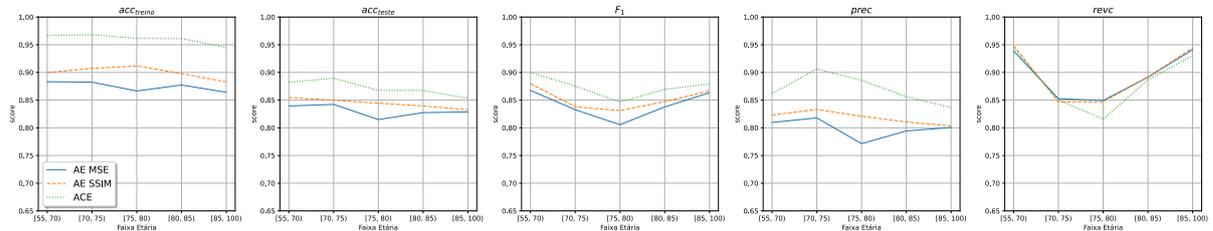
na idade dos pacientes. Todavia, o declínio cognitivo natural em decorrência da idade pode ser confundido com os estágios iniciais de demência. Ao observar os gráficos da Figura 20 e Figura 21, é possível concluir que para todas as idades os autoencoders foram capazes de extrair as características necessárias para o classificador obter boa acurácia.

Figura 20 – Média dos resultados da classificação com XGBoost por plano e faixa etária, grupo (C).



Fonte: elaborada pelo autor.

Figura 21 – Média dos resultados da classificação com XGBoost por faixa etária, grupo (D).



Fonte: elaborada pelo autor.

Perante os resultados apresentados, conclui-se que, todos autoencoders foram capazes de extrair características relevantes das imagens de RM para o uso na classificação. O ACE com classificação do XGBoost obteve os melhores resultados para as métricas de acurácia,  $F_1$  e precisão, e resultados próximos aos demais para a revocação. Somado a isso, a estratégia do grupo (D) foi a que obteve os melhores resultados. Assim, a estratégia de transferência de aprendizado mostrou-se eficaz para a tarefa de especializar os autoencoders por faixa etária.



## 5 Conclusões

Neste projeto, foram apresentados três autoencoders especializados na classe CN: um AE com função de perda MSE e outro com SSIM, e um ACE. Os autoencoders realizaram o processo extração de características para gerar um vetor de 196 atributos a partir de imagens bidimensionais de RM dos planos axial, coronal e sagital. O objetivo dos autoencoders era aprender as características mais importantes das imagens para permitir a diferenciação e, conseqüentemente, classificação nas classes CN e DA com o classificador XGBoost.

A análise dos quantitativa dos resultados foi realizada por meio das métricas de acurácia,  $F_1$ -score, precisão e revocação. Nesse sentido, notou-se que o ACE, apresentou os melhores resultados para a classificação com todas as métricas, exceto revocação. Embora a média de revocação com ACE não tenha sido a melhor, os valores obtidos estão muito próximos dos demais autoencoders e dentro da margem de desvio padrão. Os AE com função de perda MSE e SSIM também apresentaram bons resultados mas a métrica SSIM obteve resultados superiores.

Ademais, os melhores resultados foram obtidos para os autoencoders que utilizaram a transferência de aprendizado para especialização por faixa etária e classificação com concatenação dos vetores de cada plano. Portanto, a estratificação por idade torna-se um processo fundamental ao trabalhar com imagens de RM visto que, o cérebro de pacientes de idade avançada passa por mudanças estruturais que não estão presentes em outras faixas etárias.

Por fim, conclui-se que, autoencoders especializados na classe CN são capazes de aprender as características necessárias para o classificador XGBoost identificar corretamente as classes CN e DA em imagens bidimensionais de RM. Além disso, a métrica SSIM é superior à MSE como função de perda para AE e os ACE apresentam melhores resultados que AE. A Tabela 10 apresenta a comparação dos melhor resultado obtido neste trabalho com outros na literatura.

Tabela 10 – Comparação entre resultados da literatura e o melhor resultado deste trabalho.

Referência	Dados de RM	Modelo	Acurácia (%) CN × DA
SUK, LEE e SHEN (2015)	93 atributos de regiões de interesse	AE e <i>Multi-kernel</i> SVM	98.8 ± 0.4
FERRI et al. (2021)	16 atributos de regiões de interesse	(i) AE com <i>softmax</i> (ii) AE especializado com erro de reconstrução	(i) 84.3 ± 6.9 (ii) 82.4 ± 7.3
YAGIS, SECO e CITI (2021)	Imagens 2D	ACE e Rede Neural Artificial	74.66 ± 0.01
PAYAN e MONTANA (2015)	Imagens 2D e 3D	AE e Rede Neural Convolutacional	95.39
HEDAYATI, KHEDMATI e GORJIKOLAIE (2021)	Imagens 3D	Onze ACE especializados por template e Rede Neural Convolutacional	95
Este trabalho	<b>Imagens 2D por plano</b>	<b>ACE especializado por plano e faixa etária com XGBoost (D)</b>	<b>86.2 ± 4.9</b>

Como sugestão para trabalhos futuros: é relevante o estudo da métrica SSIM em ACE, uma vez que apresentou resultados superiores à MSE para AE; estudar o desempenho dos modelos estudados neste projeto em imagens 3D e avaliar se o maior custo computacional provê melhores resultados.



## Referências

- BALDI, P. Autoencoders, Unsupervised Learning, and Deep Architectures. In: v. 27, p. 37–49.
- BANK, D.; KOENIGSTEIN, N.; GIRYES, R. Autoencoders. arXiv, 2021.
- BUADES, A.; COLL, B.; MOREL, J. Non-Local Means Denoising. **IPOL Journal**, Image Processing On Line, v. 1, p. 208–212, 2011.
- CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. In: PROCEEDINGS of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Cidade de Nova Iorque, Nova Iorque: Association for Computer Machinery, 2016. P. 785–794.
- DUDA, R.; HART, P.; STORK, D. **Pattern Classification**. [S.l.]: Wiley, 2000.
- FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC, 2021.
- FALCO, A. et al. Doença de Alzheimer: Hipóteses Etiológicas e Perspectivas de Tratamento. **Química Nova**, v. 39, n. 1, p. 63–80, 2016.
- FERRI, R. et al. Stacked autoencoders as new models for an accurate Alzheimer’s disease classification support using resting-state EEG and MRI measurements. **Clinical Neurophysiology**, v. 132, n. 1, p. 232–245, 2021.
- FRIEDMAN, J. Greedy function approximation: a gradient boosting machine. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189–1232, 2001.
- \_\_\_\_\_. Stochastic gradient boosting. **Computational Statistics & Data Analysis**, Elsevier, v. 38, n. 4, p. 367–378, 2002.
- GAUTHIER, S. et al. **World Alzheimer Report 2021**. Inglaterra, 2021.
- GONZALEZ, R.; WOODS, R. **Digital Image Processing**. Cidade de Nova Iorque, Nova Iorque: Pearson, 2018.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge, Massachusetts: The MIT Press, 2016.
- HEDAYATI, R.; KHEDMATI, M.; GORJIKOLAIE, M. Deep feature extraction method based on ensemble of convolutional autoencoders: Application to Alzheimer’s disease diagnosis. **Biomedical Signal Processing and Control**, Elsevier, v. 66, 2021.
- HOU, Y. et al. Ageing as a risk factor for neurodegenerative disease. **Nature Reviews Neurology**, v. 15, n. 10, p. 565–581, 2019.
- JOLLIFFE, I. **Principal Component Analysis**. Cidade de Nova Iorque, Nova Iorque: Springer, 2002.
- KNOPMAN, D. et al. Alzheimer’s disease. **Nature Reviews Disease Primers**, Springer Nature, v. 7, n. 33, 2021.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, 1998.
- MANJÓN, J. MRI Preprocessing. In: MARTÍ-BONMATÍ, L.; ALBERICH-BAYARRI, A. (Ed.). **Imaging Biomarkers**. [S.l.]: Springer, Cham, 2017. cap. 5, p. 53–63.
- NYUL, L.; UDUPA, J.; XUAN, Z. New variantes of a method of MRI scale standardization. **IEEE Transactions on Medical Imaging**, IEEE, v. 19, n. 2, p. 143–150, 2000.

- OURSELIN, S. et al. Reconstructing a 3D structure from serial histological sections. **Image and Vision Computing**, Elsevier, v. 19, n. 1, p. 25–31, 2001.
- PAYAN, A.; MONTANA, G. Predicting Alzheimer’s disease: a neuroimaging study with 3D convolutional neural networks. **Computer Research Repository (CoRR)**, arXiv, 2015.
- PRINCE, M.; BRYCE, R.; FERRI, C. **World Alzheimer Report 2011**. Londres, 2011.
- PRINCE, M. et al. **World Alzheimer Report 2015**. Londres, 2015.
- SUK, H.; LEE, S.; SHEN, D. Latent feature representation with stacked auto-encoder for AD/MCI diagnosis. **Brain Struct and Function**, Springer, v. 220, p. 841–859, 2015.
- TAKIZAWA, C. et al. Epidemiological and Economic Burden of Alzheimer’s Disease: A Systematic Literature Review of Data across Europe and the United States of America. **Journal of Alzheimer’s Disease**, IOS Press, v. 43, n. 4, p. 1271–1284, 2015.
- THIES, W.; BLEILER, L. 2012 Alzheimer’s disease facts and figures. **Alzheimer’s & Dementia**, Elsevier, v. 8, n. 2, p. 131–168, 2012.
- TUSTISON, N. et al. N4ITK: improved N3 bias correction. **IEEE Transactions on Medical Imaging**, IEEE, v. 29, n. 6, p. 1310–1320, 2010.
- 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.
- YAGIS, E.; SECO, A.; CITI, L. Convolutional Autoencoder based Deep Learning Approach for Alzheimer’s Disease Diagnosis using Brain MRI. In: 2021 IEEE 34th International Symposium on Computer-Based Medical Systems. [S.l.: s.n.], 2021. P. 486–491.