

# ResNet-18 e Local Binary Pattern como abordagem para o diagnóstico de Alzheimer

1<sup>st</sup> Arthur Gonçalves  
Instituto de Ciência e Tecnologia  
Universidade Federal de São Paulo  
São José dos Campos, Brasil  
goncalves01@unifesp.br

2<sup>nd</sup> Guilherme Nicolasi  
Instituto de Ciência e Tecnologia  
Universidade Federal de São Paulo  
São José dos Campos, Brasil  
guilherme.nicolasi@unifesp.br

## I. RESUMO

Este trabalho tem por objetivo a análise de neuroimagens de ressonância magnética como contribuição à detecção precoce da Doença de Alzheimer (DA), por meio das técnicas deliberadas durante a eletiva de Processamento de Imagens (PI), ministrada pelo Prof. Dr. Fabio Faria, da Universidade Federal de São Paulo, bem como o estudo comparativo entre o algoritmo de aprendizado de máquina supervisionado Support Vector Machine (SVM) e o descritor visual de textura Local Binary Pattern (LBP), voltado ao domínio de pesquisa proposto.

Para melhorar a qualidade das Imagens de Ressonância Magnética (MRIs), oriundas da base de dados Alzheimer's Dataset (4 class of Images), disponibilizado pela Kaggle, e adquirir mais informações subjacentes, aplicamos, durante o pré-processamento, uma equalização de histograma e convertimos as imagens, ainda em tons de cinza, em imagens com pseudocor. Assim, ao colorir as imagens, enfatizamos regiões de interesse que podem nos ajudar a identificar as principais áreas afetadas pela DA.

Seguindo, utilizamos o LBP em cada imagem, extraíndo seus padrões, posteriormente, estruturados em features (vetores de características), que, conseguinte, foram processados pelo SVM para o diagnóstico do paciente.

Para o treinamento da Rede Neural Convolucional (CNN, da tradução de Convolutional Neural Network), selecionamos 5 (cinco) conjuntos com 90% de imagens randomizadas, então usadas no modelo, passando por 5 épocas, e, conseguinte, obtivemos as features de mínimos locais, com a descida de gradiente estocástico.

Por fim, inserimos todas as features, extraídas através de ambos os métodos, em uma SVM, separando-as em classe LBP e ResNet. Após aplicar o teste no modelo obtido, observamos que a ResNet-18 apresentou uma acurácia muito superior ao LBP, evidenciando que as CNNs, nesse contexto, são mais eficientes do que os métodos de Processamento de Imagens aplicados para analisar as informações presentes na imagem.

## II. INTRODUÇÃO

Uma das formas mais comuns de demência neurodegenerativas é o Alzheimer. Segundo o Ministério da Saúde (MS), cerca de 1,2 milhão de brasileiros sofrem dessa doença, e estima-se por volta de 35,6 milhões de pessoas no mundo. Seus níveis variam do estado leve ao severo e cada grau traz, consigo, algumas características, como a perda de

memória recente, dificuldade de encontrar palavras, prejuízo gravíssimo da memória, incontinência urinária, disfunção motora, entre outros. O diagnóstico antecipado pode proporcionar um retardo do avanço da doença e a redução dos sintomas.

Para uma doença que afeta não só a pessoa acometida, mas, também, os familiares, principalmente na área emocional, o descobrimento antecipado proporciona tanto uma esperança a essas pessoas quanto uma qualidade de vida melhor para todos. Portanto, esse trabalho tem como objetivo facilitar e agilizar todo esse processo de diagnóstico da doença.

A diferença que a doença traz ao cérebro, visualmente, é significativa, principalmente em condições mais avançadas. No entanto, para que seja possível notar tal diferença, é necessário ter uma amostra do cérebro do paciente. Para isso, são feitas ressonâncias magnéticas, ferramenta que nos ajuda a projetar o objeto de estudo. O revés, da utilização dessa tecnologia, surge a partir daí já que em MRI (Magnetic Resonance Images) o que era perceptivo passa a ser difícil de identificar, logo, é necessário métodos que possibilitam analisar tais imagens obter diagnósticos precisos e confiáveis.

Portanto, para que isso seja possível há vários caminhos a serem tomados, por exemplo:

- A. [1] *Deteção Precoce de Alzheimer Usando Machine Learning*, as pesquisadoras Nathalia Paiva e Tatiana Escovedo estudaram o desempenho de certas técnicas de Machine Learning para ajudar com a detecção precoce da demência. Com o uso de algoritmos supervisionados, as pesquisadoras analisaram o dataset OASIS-1, contendo tanto amostras acometidas, com alguns graus diferentes, quanto não acometidas da doença. Juntamente a essas imagens, temos alguns outros atributos relacionados, como id, gênero, se o indivíduo é destro ou canhoto, idade e nível de educação. Com base nesses atributos, Nathalia e Tatiana aplicaram uma correlação de Pearson, em um intervalo de 1 a -1, no qual 1 os atributos estão fortemente ligados e -1 não há nenhuma relação. Rumo ao Machine Learning, elas utilizaram de 5 algoritmos dentre eles: Regressão Logística, Random Forest, Naive Bayes, Multilayer Perceptron e KNN.
- B. [2] *Diagnosis of Alzheimer's Disease Severity with fMRI Images Using Robust Multitask Feature Extraction Method and Convolutional Neural Network (CNN)*, os pesquisadores Morteza Amini, Mir Mohsen Pedram, AliReza Moradi e Mahshad

Ouchani, da Universidade Shahid Beheshti, optaram por reter propriedades de imagens de ressonância magnética T1 com técnicas de pré-processamento, como a suavização gaussiana e redução de ruído por Quantum Matched-Filter Technique; e aplicar algoritmos distintos de classificação (KNN, SVM, DT, LDA, RF e CNN) para identificar os estados de desenvolvimento da doença sobre cada amostra analisada, demonstrando resultados promissores para os algoritmos Support Vector Machine, Random Forest e para a Rede Neural Convolutacional.

- C. [5] *Suporte ao Diagnóstico da Doença de Alzheimer a partir de Imagens de Ressonância Magnética*, o pesquisador Bruno Tavares Padovese trouxe, em seu trabalho, uma abordagem de recuperação de imagem de forma nova, utilizando descritores de propósito geral sem nenhuma ação de pré-processamento nas imagens e um pós-processamento não supervisionado. Ao que aparenta, há a capacidade de alcançar resultados satisfatórios para o propósito mor, diagnóstico da doença.

Nesse material de estudo, optamos por utilizar uma CNN, Resnet-18, e um descritor visual de texturas, LBP; comparando-os com os resultados extraídos de uma SVM, ou seja, verificando qual possibilita um acerto maior quanto à detecção da doença de Alzheimer.

### III. FUNDAMENTAÇÃO TEÓRICA

Como dito, anteriormente, alguns métodos foram utilizados para a realização desse trabalho. Aqui estão algumas definições e explicações para entender melhor como foram feitas as abordagens. O código para o processamento de imagem, em si, utilizou de funções prontas do OpenCV, exceto o LBP, mas cada técnica pode ser feita por meios matemáticos convencionais.

#### A. Abordagens em Processamento de Imagens

- Realce por Equalização: transformação a qual modifica-se o histograma da imagem original de forma que a imagem resultante possua uma distribuição mais uniforme dos seus níveis de cinza, portanto, seus níveis de cinza devem ficar, aproximadamente, com as mesmas frequências. Abaixo temos, na Fig.1, uma foto equalizada;

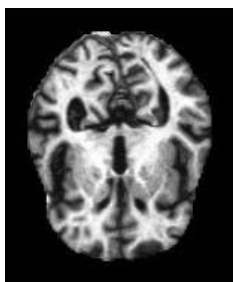


Fig. 1

- Realce Pseudocor: essa transformação utiliza um conjunto de cores para realçar regiões de interesse,

auxiliando a interpretação das imagens. Podendo ser aplicada em imagens com tons de cinza ou até em imagens coloridas. Apesar de, às vezes, o resultado não corresponder com as cores verdadeiras, podemos aumentar a qualidade das imagens, significativamente, e incluir novas informações. Abaixo vemos, na Fig. 2, uma imagem com Pseudocor e, na Fig. 3, uma imagem normal;

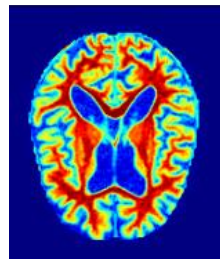


Fig. 2

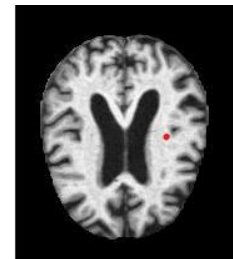


Fig. 3

- Descritor LBP: sendo uma técnica responsável por discretizar a imagem, ou seja, representar algo aparentemente não numérico (objetos, por exemplo) em números, ou melhor dizendo, vetores de características, esse descritor se especializa em “encontrar” padrões de texturas nas imagens. Tal procedimento é bem simples. Uma máscara, de tamanho 3x3, é passada pela imagem inteira e, com base em um pixel central, verifica-se se os outros pixels dentro dessa máscara são maiores ou menores que o mesmo, caso maior recebe valor 1, caso contrário valor 0. Assim, escolhe-se um padrão para se obter um número binário dessa máscara, que, passa por uma conversão de binário para decimal e o valor obtido é incrementado num histograma (tons de cinza  $\times$  ocorrências). Abaixo vemos o resultado do LBP aplicado sobre uma imagem com Pseudocor.



Fig. 4

#### B. Abordagens em Aprendizado de Máquina

- K-fold Cross-Validation: Neste método, o conjunto de dados é dividido em dois segmentos: treino, para formar o modelo, e teste, para validá-lo. A forma mais comum de validação cruzada é a validação cruzada k-fold, em que os dados são divididos em k conjuntos de mesmo tamanho. Consequentemente, as k iterações de formação e validação são realizadas de tal forma que

dentro de cada iteração é extraído um conjunto diferente de dados para a validação, enquanto o restantes k-1 conjuntos são utilizadas para a aprendizagem. O desempenho do algoritmo de aprendizado é então analisado através de uma métrica de desempenho, tal como a precisão, e os resultados podem ser utilizados para fazer uma comparação com outros modelos. Abaixo, na figura 5, temos um exemplo de validação cruzada 10-fold;

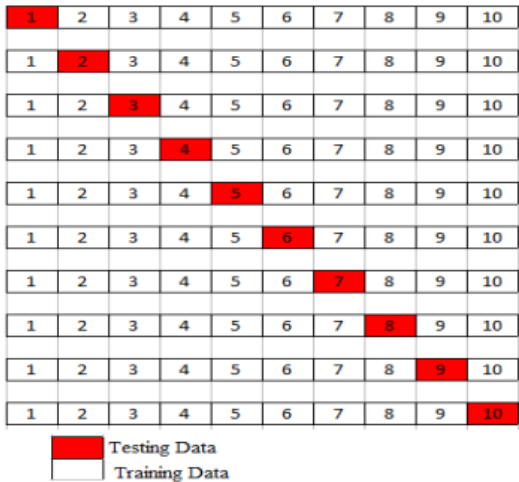


Fig. 5

• Resnet-18: A ResNet, abreviação de Residual Network, é uma arquitetura de rede neural profunda caracterizada por apresentar camadas residuais, criadas para evitar o problema do gradiente de desaparecimento, permitindo que a rede aprenda representações mais profundas e complexas, e conhecida por seu ótimo desempenho em tarefas de classificação e reconhecimento de imagens. Atualmente, existem várias versões da ResNet na literatura, incluindo a ResNet-18, ResNet-34, ResNet-50, entre outras. A ResNet-18, por exemplo, que é a versão mais simples do modelo, possui 18 camadas, divididas em 5 blocos, o que a torna mais fácil de treinar e mais rápida para executar. Cada bloco consiste em camadas de convolução, seguidas por camadas de normalização e camadas de ativação ReLU. A camada de convolução é responsável por extrair características das imagens de entrada, enquanto a normalização ajuda a estabilizar o treinamento, e a camada de ativação ReLU a introduzir não-linearidade à rede. A camada residual, caso de uso dessa arquitetura, é uma camada adicional que segue cada bloco e permite que a informação dos blocos anteriores seja adicionada à camada atual, possibilitando que a rede aprenda representações mais profundas. Além dessas camadas principais, a ResNet também contém camadas de pooling, para reduzir a dimensionalidade dos dados de entrada de cada camada, e a fully connected, também conhecida como camada densa, que é a última camada da rede neural, responsável por realizar a classificação das imagens. Segue, abaixo, nas figuras 6 e 7, a

arquitetura e versões de redes residuais, respectivamente.

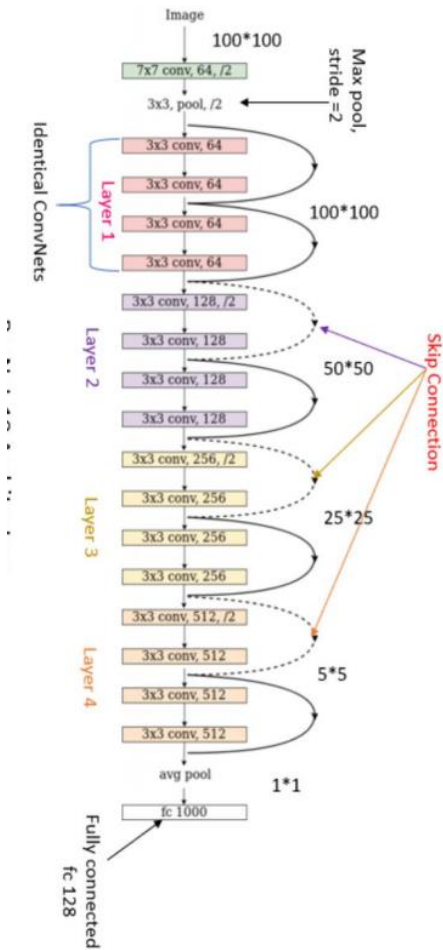


Fig. 6

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2,x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3,x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4,x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5,x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
1×1		average pool, 1000-d fc, softmax				
FLOPs		1.8×10 <sup>9</sup>	3.6×10 <sup>9</sup>	3.8×10 <sup>9</sup>	7.6×10 <sup>9</sup>	11.3×10 <sup>9</sup>

Fig. 7

• Support Vector Machine: O Support Vector Machine (SVM) é um método de aprendizado de máquina supervisionado não-paramétrico, ou seja, que utiliza parte ou todo o conjunto de treinamento para realizar uma tarefa, ótimo para problemas de classificação e regressão em domínios não-linearmente separáveis (de alta dimensionalidade). O SVM usa uma função kernel, que transforma os dados de entrada em um espaço de características de maior dimensão, para projetar os dados de um espaço em outro, criando hiperplanos, superfícies que dividem o espaço de

características e servem como thresholds (limiares) para separar os conjuntos de dados. O objetivo desse algoritmo é encontrar um hiperplano que maximize a margem de distância entre os dados separáveis e construir uma fronteira de decisão linear através de vetores de suporte, amostras de dados que estão mais próximas ao hiperplano, definindo, também, a tolerância a erros do modelo.

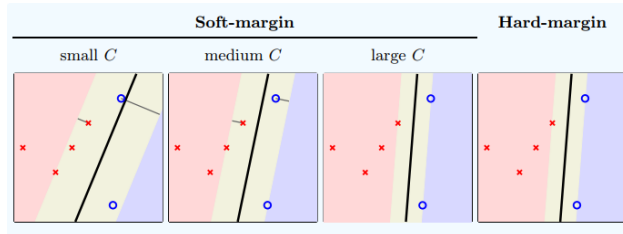


Fig. 8

#### IV. METODOLOGIA PROPOSTA

Conforme acordado, no início do trabalho, definimos um esquema padrão presente na literatura para todo o procedimento de Processamento de Imagens. Abaixo, temos esse esquema representado na Fig 9.



Fig. 9

##### A. Aquisição e Preparação

Para adquirir as imagens, buscamos por um dataset na Kaggle, plataforma que disponibiliza inúmeras bases de dados para pesquisas, experimentos, aprendizados de máquina entre outros; escolhemos, portanto, o dataset Alzheimer's Dataset (4 class of Images), que já se encontrava estruturado e, portanto, não precisamos modificá-lo. A base continha duas pastas: uma para teste e outra para treino, sendo que ambas apresentavam outras 4 pastas (Mild, Very Mild, Moderate e Non), que seriam os níveis da doença, "Leve", "Muito Leve", "Moderado" e "Nenhum", respectivamente. Optamos, também, por juntar todas as imagens que referenciavam qualquer tipo de demência em uma lista "Demented" e o restante em outra "Non Demented", ou seja, sem demência alguma.

##### B. Pré-Processamento

O pré-processamento é uma etapa de extrema importância, pois é nela onde temos a possibilidade de realçar determinadas características, antes imperceptíveis na imagem. Todas as técnicas citadas, brevemente, foram realizadas por meio dos métodos disponibilizados pela própria OpenCV.

Portanto, dado que as imagens não estavam todas com os mesmos contrastes, consideramos que o Equalizador (função `cv2.equalizeHist()`) seria necessário para padronizar as entradas, independente do grupo a qual pertencia. No entanto,

fizemos uma coloração por meio da técnica de Pseudocolor (função `cv2.applyColorMap(imagem, tipoCor)`), esta que é o meio ao qual permite, como dito, anteriormente, ajudar visualmente a localização de regiões de interesse e melhorar a qualidade da imagem.

##### C. Descrição (LBP)

Partindo para o LBP, há uma função que recebe a imagem em seu estado de colorido e a transforma em tons de cinza, pois não é necessário, para o Local Binary Pattern, reconhecer as cores da imagem, assim, para a conclusão do procedimento do descritor, o esquema utilizado para adquirir o binário da máscara é em sentido horário, começando no pixel do canto esquerdo acima e terminando no pixel abaixo do inicial.

Como retorno dessa função, inicialmente dita, temos um vetor com valores de 0 a 255, onde cada índice do vetor é incrementado a partir do momento que tal pixel, com determinada intensidade, é localizado. Esse histograma, portanto, é o que chamamos de features, uma descrição da imagem.

##### D. Análise das informações (ResNet-18)

Na etapa da ResNet-18, foi selecionado 90% do conjunto de dados para cinco experimentos, então dividimos em dois outros conjuntos (treino e teste), usando a classe KFold da biblioteca Scikit-Learn para realizar a validação cruzada, e, depois, os resultados foram alocados em duas listas (train\_sets e test\_sets), para o treino e teste do modelo.

Após esse processo, algumas transformações, como dimensão e normalização, são definidas para as amostras de dados, antes de serem enviadas para a rede. Em seguida, é definido o número de épocas e o tamanho dos lotes, e o código entra em um laço de repetição para acessar cada conjunto de treinamento e teste. Dentro do laço, uma ResNet-18 pré-treinada é instanciada e o modelo é carregado para o dispositivo (cuda ou cpu) usando o método `to()`. Com isso, inicia-se um outro laço para cada época, a rede entra em modo de treino, com o método `train()`, e o otimizador Stochastic Gradient Descent e a função de perda Cross Entropy são definidas, respectivamente, usando os métodos `torch.optim.SGD` e `torch.nn.CrossEntropyLoss`. Após isso, o modelo faz a retropropagação com o método `backward()`, para encontrar o gradiente da função de perda e o otimizador então atualizar os parâmetros da rede com o método `step()`. Por fim, o modelo entra em modo de avaliação, com método `val()`, o código faz o cálculo da acurácia, verificando os resultados das previsões de teste, e as features das imagens de treino e teste processadas são extraídas e alocadas em duas listas (train\_features e test\_features).

##### E. Resultados

Nesta etapa, tivemos que pegar todas as features recuperadas pelo LBP (os histogramas das imagens) e pela CNN, além de inseri-las na SVM, como dois conjuntos de entrada distintos, e, a partir de seus resultados, comparar as acurácias e verificar qual das duas técnicas é mais adequada, ou seja, mais efetiva quanto ao diagnóstico do paciente.



## V. EXPERIMENTOS E DISCUSSÃO

A dimensão das imagens, contidas na base de dados, é 176x208 e, para o experimento, as amostras foram divididas em destinadas para o treino (2560 não dementes e 2569 dementes) e as destinadas para o teste (639 não dementes e 640 dementes).

Os parâmetros utilizados no pré-processamento foram mínimos, pois as funções disponibilizadas pelo OpenCV não demandavam entradas específicas, exceto pela Pseudocor, em que foi necessário definir o conjunto de cores para a colorização e, dentre os disponíveis, escolhemos o `cv2.COLORMAP_JET`, que ocupa o local do `tipoCor` na função exibida em IV.2.

Para a descrição das imagens, através do LBP, utilizamos uma máscara de dimensões 3x3, passando por toda a imagem e fazendo cálculos em relação ao pixel que se encontrava no centro da máscara. Esses cálculos comparam a magnitude dos pixels vizinhos, enquadrados na máscara, em relação ao pixel central, sendo o threshold igual ao valor do pixel central. Caso seus pixels vizinhos fossem maiores, ser-lhes-ia atribuído o valor 1, e 0 aos que fossem menores. Assim, dentro da máscara teríamos um número binário, denotado a partir de um método arbitrário. O número era formado a partir do pixel do canto superior esquerdo e seguia-se, em sentido horário, até que o último fosse o pixel abaixo do primeiro. Por fim, convertimos o número para o sistema numérico decimal e atribuímos o valor na posição do pixel central, gerando uma nova imagem. Abaixo, temos um dos histogramas adquiridos no processo.

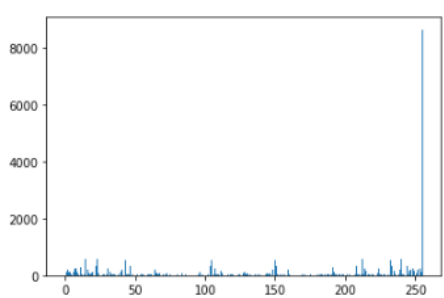


Fig. 10

Finalizando, partimos para o SVM com kernel linear, pegamos as features disponibilizadas tanto pelo LBP quanto pela CNN e as colocamos para treinamento. Como o trabalho visa a comparação, fizemos uma SVM para as features da IA e outro para as features do LBP. O treinamento pegou, das features coletadas, 80% para treino e 20% para teste, sendo que foram escolhidos aleatoriamente e, em ambos os SVMs, utilizamos um randomState, arbitrariamente, igual a 42. Após o treinamento fizemos o teste e coletamos a acurácia obtida. Ao final, plotamos um gráfico usando o `plt.scatter` mostrando as regiões ocupadas por cada instância. Abaixo, na Fig. 11, temos o resultado da SVM para a ResNet-18 e, na Fig. 12, temos o resultado do SVM para o LBP.

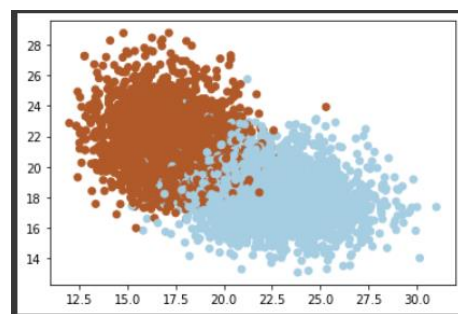


Fig. 11

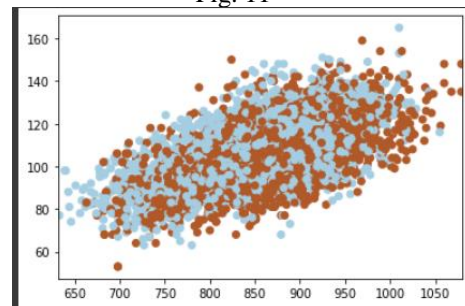


Fig. 12

## VI. CONCLUSÃO

O LBP mostrou-se bem aquém do que esperávamos, já que, em alguns trabalhos da literatura, foram obtidas acurácias melhores, com média entre 75% e 85%. No entanto, acreditamos que a queda de sua acurácia pode ser resultado do grande número de pixels de mais alta frequência. Por outro lado, a ResNet-18 cumpriu com seu trabalho e nos surpreendeu com sua acurácia entre 92% e 96%, mostrando, portanto, que o aprendizado de máquina é, certamente, muito importante e quase que indispensável para problemas de classificação na área da saúde.

Assim, analisando nosso trabalho e os trabalhos relacionados, a DA é bastante amparada na literatura quanto ao uso de IA (Inteligência artificial), seja pela inúmera quantia de trabalhos e artigos feitos ou pela eficácia da IA demonstrada para resolução do problema. Portanto, torna-se resoluto o uso complementar do Processamento de Imagens e do Aprendizado de Máquina, pois ambos trabalham de forma simbiótica e efetiva e podem melhorar nossas abstrações sobre os dados e, também, nossas tomadas de decisão.

## REFERÊNCIAS

- [1] N. Paiva, T. Escovedo, Detecção Precoce de Alzheimer Usando Machine Learning, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, [https://web.archive.org/web/20220519030843id/https://sbic.org.br/wp-content/uploads/2021/09/pdf/CBIC\\_2021\\_paper\\_2.pdf](https://web.archive.org/web/20220519030843id/https://sbic.org.br/wp-content/uploads/2021/09/pdf/CBIC_2021_paper_2.pdf)
- [2] M. Amini, M. Pedram, A. Moradi, M. Ouchani, Diagnosis of Alzheimer's Disease Severity with fMRI Images Using Robust Multitask Feature Extraction Method and Convolutional Neural Network (CNN), Shahid Beheshti University, Tehran, Iran, [Diagnosis of Alzheimer's Disease Severity with fMRI Images Using Robust Multitask Feature Extraction Method and Convolutional Neural Network \(CNN\) - PubMed \(nih.gov\)](https://pubmed.ncbi.nlm.nih.gov/35811111/)

- [3] Biblioteca Virtual da Saúde, <https://bvsmms.saude.gov.br/doenca-de-alzheimer-3/>
- [4] Biblioteca Virtual da Saúde, <https://bvsmms.saude.gov.br/21-9-dia-mundial-da-doenca-de-alzheimer-e-dia-nacional-de-conscientizacao-da-doenca-de-alzheimer/#:~:text=No%20Brasil%2C%20onde%20o%20Dia,com%20a%20Doen%C3%A7a%20de%20Alzheimer.>
- [5] Padovese, Bruno Tavares, Suporte ao Diagnóstico da Doença de Alzheimer a partir de Imagens de Ressonância Magnética, <https://docplayer.com.br/52780999-Campus-de-sao-jose-do-rio-preto-bruno-tavares-padovese-suporte-ao-diagnostico-da-doenca-de-alzheimer-a-partir-de-imagens-de-ressonancia-magnetica.html>
- [6] Dataset Alzheimer's Dataset (4 class of Images), <https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images>
- [7] Using a ResNet-18 Network to Detect Features of Alzheimer's Disease on Functional Magnetic Resonance Imaging,

[https://adni.loni.usc.edu/adni-publications/Nicholas\\_2022\\_using%20a%20ResNet-18%20network.pdf](https://adni.loni.usc.edu/adni-publications/Nicholas_2022_using%20a%20ResNet-18%20network.pdf)

[8] ResNets and Attention , <https://www.cs.toronto.edu/~lczhang/321/notes/notes17.pdf>

[9] Deep Residual Learning for Image Recognition , <https://arxiv.org/pdf/1512.03385.pdf>

[10] Resnet18 Model With Sequential Layer For Computing Accuracy On Image Classification Dataset , <https://ijcrt.org/papers/IJCRT2205235.pdf>

[11] Support Vector Machine, <https://edisciplinas.usp.br/pluginfile.php/5078086/course/section/5978681/chapSVM.pdf>

Ferreira , Eduardo Vargas, Support Vector Machines, [http://cursos.leg.ufpr.br/ML4all/slides/Support\\_Vector\\_Machines.pdf](http://cursos.leg.ufpr.br/ML4all/slides/Support_Vector_Machines.pdf)

Fletcher, Tristan Support Vector Machines Explained, [https://www.csd.uwo.ca/~xling/cs860/papers/SVM\\_Explained.pdf](https://www.csd.uwo.ca/~xling/cs860/papers/SVM_Explained.pdf)