

Reconhecimento de Padrões por Textura em Imagens Mamográficas

Ana Laura Oliveira
Ciência da Computação
Pontifícia Universidade Católica
de Minas Gerais
Belo Horizonte, Minas Gerais,
Brasil
analauroferoli@gmail.com

Larissa Gomes
Ciência da Computação
Pontifícia Universidade Católica
de Minas Gerais
Belo Horizonte, Minas Gerais,
Brasil
1201304@sga.pucminas.br

Pedro Henrique Carvalho
Ciência da Computação
Pontifícia Universidade Católica
de Minas Gerais
Belo Horizonte, Minas Gerais,
Brasil
pedrocarvalho@outlook.com

INTRODUÇÃO

O câncer de mama é um dos mais registrados na atualidade. De acordo com a American Cancer Society (2022), estima-se que cerca de 30% dos tumores diagnosticados anualmente em mulheres sejam de mama [6]. Neste cenário, o diagnóstico precoce se mostra importante para o aumento da sobrevida e chance de cura das pacientes. Tal diagnóstico ocorre principalmente por meio de exames de imagem do tecido mamário [7].

A mama é formada por diferentes componentes, entre eles glândulas secretoras de leite, ductos epiteliais, tecido adiposo e estroma [4]. O exame de densidade mamográfica avalia a proporção entre os tecidos adiposo, epitelial e estromal [4, 7].

A literatura apresenta a correlação entre o aumento da densidade mamária com a prevalência de lesões malignas em mulheres [4, 5]. Um tecido mamário mais denso está relacionado a uma maior dificuldade em detectar lesões em fase inicial, prejudicando o diagnóstico precoce da patologia e impactando diretamente no sucesso do tratamento [4, 7].

Tecnologias para reconhecimento de padrões têm sido utilizadas como ferramentas auxiliares no diagnóstico e estudo de lesões malignas [8, 9]. No contexto do câncer de mama, identificar tecidos mamários mais densos poderia contribuir para classificar o risco e estabelecer orientações específicas para a paciente, aumentando a vigilância e a chance de diagnosticar precocemente lesões malignas [7].

Este trabalho apresenta como objetivo desenvolver um classificador que utiliza como entradas descritores de textura para identificar tecidos mamários discriminando-os entre as quatro classes de densidade mamária (BIRAD).

MATERIAIS E MÉTODOS

Para classificar a classe de BIRAD de uma imagem, é necessário analisar as características de textura. Utilizando os descritores de Haralick, é possível extrair informações deste quesito.

A partir de uma imagem, foram calculadas matrizes de co-ocorrência circulares para os raios 1, 2, 4, 8 e 16. Foi feita a fusão das matrizes de mesmo raios de uma imagem e a partir dessa fusão para cada combinação de pixels foi calculada sua probabilidade. A partir dessa matriz normalizada, os cálculos para extrair os descritores de entropia, homogeneidade e energia foram realizados. Dessa forma, para cada imagem é retornada uma matriz com a seguinte configuração:

Matriz	Homogeneidade	Entropia	Energia
C1	x	x	x
C2	x	x	x
C4	x	x	x
C8	x	x	x
C16	x	x	x

Figura 1: **Formato Matriz de Características de Textura Extraída de uma imagem**

Após extrair os descritores de várias imagens é necessário treinar classificadores para que seja possível efetuar a classificação de outras imagens posteriormente. Dois tipos de classificadores foram implementados neste trabalho: Rede Neural e SVM.

Tanto para o treinamento da Rede Neural, quanto da SVM, uma matriz de 4 dimensões é recebida com descritores de várias imagens. A primeira dimensão se refere a classe, a segunda se refere às imagens de cada classe, a terceira possui 5 matrizes, com os descritores gerados por cada matriz de co-ocorrência e na quinta dimensão é possível acessar esses valores.

Inicialmente os dados são tratados para separar os elementos de teste e treino. As duas últimas dimensões são transformadas em um vetor linear e rotuladas conforme a sua classe, esses vetores passam por uma função de *shuffle*, a fim de treinar/testar a rede com dados escolhidos de forma aleatória. 75% das imagens de cada classe são escolhidas para treino e 25% para teste.

Classificador Rede Neural

O classificador de Rede Neural foi implementado com a biblioteca *Keras* (Sessão c.2), biblioteca que implementa o aprendizado de máquina por *Deep Learning*. Esta técnica utiliza dados rotulados e, por uma quantidade pré-definida de épocas, ajusta os pesos de cada neurônio para classificar elementos recebidos para treino.

Após o tratamento dos dados, o modelo de rede neural é montado, possuindo uma camada oculta intermediária, com a função de ativação *ReLU* e com 42 neurônios. Há também uma camada de saída, que possui a função de ativação *Softmax* e 4 neurônios, um para cada classe de BIRAD. O treino da rede é feito em 500 épocas.

Classificador SVM (Support Vector Machine)

O classificador SVM foi implementado utilizando a biblioteca *Sklearn* (Sessão c.3). A Support Vector Machine considera cada elemento um ponto no espaço e a partir de separação por hiperplanos classifica cada elemento. O treinamento desse classificador ajusta por regressão linear qual será a melhor posição da função de classificação que irá separar estes planos.

O modelo de SVM que foi utilizado, tem o valor de $\gamma = 0.1$, dessa forma ele utiliza grande parte das entradas, e não apenas entradas próximas à função de classificação para ajustar os limites de cada hiperplano. Logo pontos mais distantes são influentes para definição do hiperplano. Ao parâmetro de regularização C foi atribuído o valor 1, para que a função de classificação não seja muito complexa e pouco flexível, assim evitando que o treino faça a máquina memorizar as posições de cada objeto ao invés de reconhecer por características generalizáveis.

Interface gráfica (TKinter)

A interface gráfica do software foi construída utilizando a biblioteca TKinter [3] de Python. Nessa interface estão presentes os menus: Opções, Matrizes Treino, Rede Neural, SVM e Imagem. Por meio destes ocorre a interação do usuário com o software.

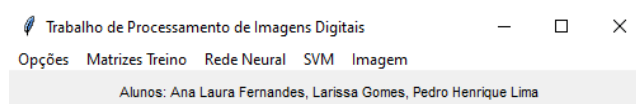


Figura 2: Menu da interface gráfica

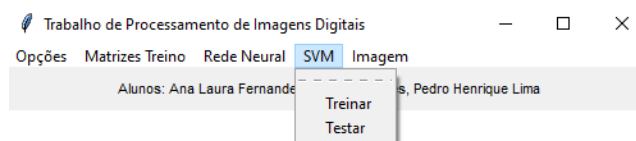


Figura 3: Submenu dos classificadores (Igual para SVM e Rede Neural)

Para realizar o treinamento de um dos classificadores, o usuário interage com o respectivo menu e seleciona a opção 'Treinar' (Figura 3). Caso seja o primeiro treino, aparecerá a opção para selecionar um diretório de imagens contendo os 4 subdiretórios correspondentes a cada classe de BIRAD. Após selecionado, popups informativos aparecerão indicando o seguinte fluxo: Cálculo das matrizes de coocorrência com o tempo de execução exibido ao final, cálculo dos descritores de Haralick (Entropia, Homogeneidade, Energia e Contraste) e informação sobre o treino do classificador. O processo é o mesmo tanto para a Rede Neural quanto para a SVM.

Ao final, as matrizes calculadas são salvas em arquivo de extensão .pkl, visando diminuir o tempo para posteriores treinos. Como mostrado na seção de resultados, o cálculo das matrizes de co-ocorrência depende o maior tempo de execução do software. Ao selecionar novamente a opção de 'Treinar', apenas os descritores são recalculados, utilizando os valores das matrizes salvos no arquivo. O cálculo das matrizes pode ser realizado novamente pelo menu "Matrizes Treino".

Para exibir os resultados do treino, utiliza-se a opção 'Testar' no menu do classificador treinado anteriormente (Figura 3). Em uma janela auxiliar são exibidas as métricas de especificidade média, sensibilidade média, tempo de execução e matriz de confusão onde a linha é a classe correta e a coluna a classe estimada. Um exemplo é mostrado na Figura 4. A matriz de confusão é exibida com um mapa de calor indicando quais foram as classes mais frequentemente estimadas para cada caso.

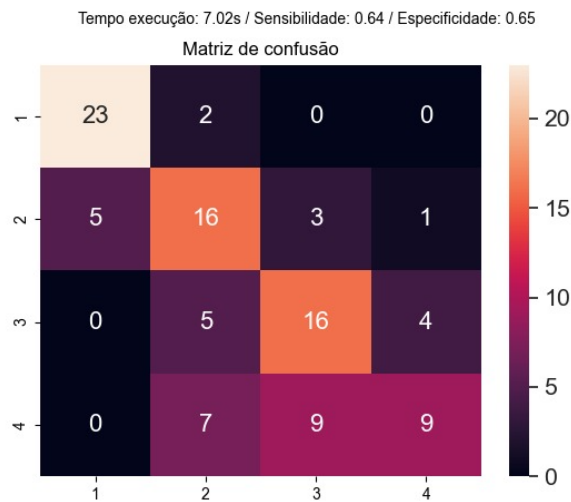


Figura 4: Exemplo da janela auxiliar de resultados exibida quando a opção 'Testar' é selecionada no submenu de um dos classificadores

No menu 'Imagem' (Figura 2) é possível selecionar uma imagem e esta é exibida no canvas, redimensionada para 500x500 para uma melhor visualização. Para os cálculos é utilizada a imagem no seu tamanho original.

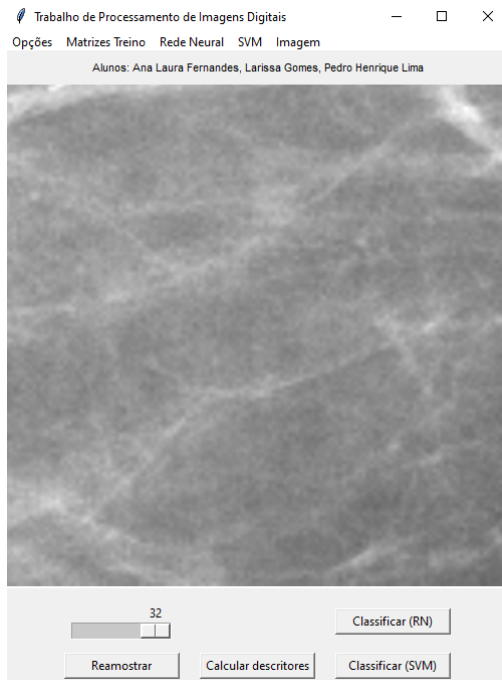


Figura 5: Exemplo de imagem exibida no canvas após ser selecionada no menu 'Imagem'

Após aberta, a imagem pode ser reamostrada para número de tons de cinza entre 2 (binarização) e 32. Por padrão são utilizados 32 tons de cinza. A imagem após a reamostragem é exibida em uma janela auxiliar (Figura 6). Esta reamostragem é considerada no momento dos cálculos dos descritores da imagem.

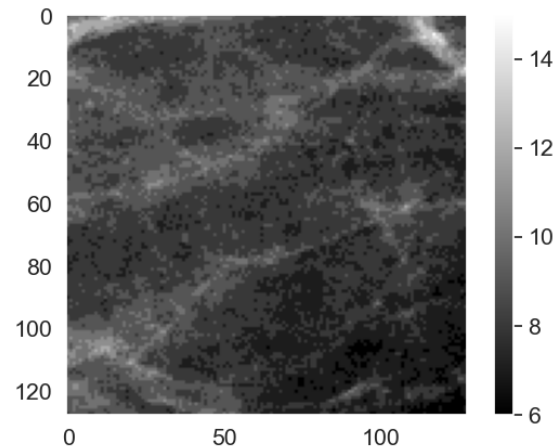


Figura 6: Exemplo de imagem reamostrada em 16 tons de cinza exibida na janela auxiliar

Após o número de tons de cinza ser selecionado, pode-se calcular os descritores para a imagem clicando no botão "Calcular descritores". Após o cálculo finalizar, é exibida uma janela auxiliar com os valores obtidos (Figura 7).

Descritores de Haralick da imagem selecionada

Tempo de execução: 3.57 segundos

Matriz de Co-ocorrência	Homogeneidade	Entropia	Energia	Contraste
C1	0.6985920355600965	4.977370254273282	0.04769984566858297	1.0196997549019613
C2	0.6299887764427701	5.341867217027089	0.03819605259345643	1.9035433070866143
C4	0.5822219696891605	5.572368734982281	0.0328172933541966	2.9542878632655944
C8	0.5347863483844942	5.76494933519676	0.0283026628879062	4.521457784345583
C16	0.49144473036973785	5.880436069323515	0.02517046747889244	5.88876709648835

Figura 7: Exemplo de valores para os descritores de Haralick da imagem selecionada

Por fim, a imagem poderá ser classificada por meio da rede neural ou da SVM. Ao selecionar um dos botões, um popup informando a classe estimada será exibido (Figura 8).

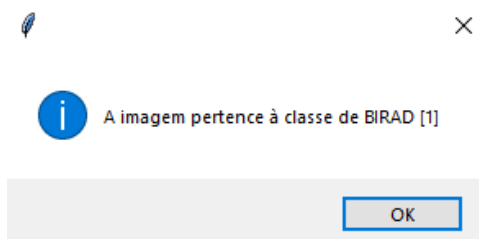


Figura 8: **Exemplo de classificação de imagem exibida no canvas**

c.2) Keras

A rede neural foi desenvolvida utilizando a biblioteca *Keras* [2]. Biblioteca de código aberto implementada utilizando *TensorFlow*. A biblioteca permite criar, treinar e testar uma rede neural.

c.3) Sklearn

O classificador SVM foi desenvolvido utilizando a biblioteca de código aberto *Sklearn* [1], de desenvolvimento de aprendizado de máquina. A biblioteca permite criar, treinar e testar uma SVM.

RESULTADOS

d) Medidas de tempo de execução para diversas imagens, descritores e hiperparâmetros do classificador.

d.1) Tempo de Cálculo - Matriz de Co-ocorrência e descritores:

Tempo médio gasto para o cálculo de matrizes de co-ocorrência e descritores de Haralick para 1 e 400 imagens em diferentes Sistemas Operacionais:

Entropia, Homogeneidade, Energia e Contraste

SO / Qtd. Imagens	1	400
Mac OS	1,93 segundos	12,62 minutos
Windows	3,65 segundos	26,29 minutos

Figura 9: **Tempo gasto para cálculo de matriz de co-ocorrência e 4 descritores de Haralick.**

Entropia, Homogeneidade e Energia

SO / Qtd. Imagens	1	400
Mac OS	1,88 segundos	12,58 minutos
Windows	4,29 segundos	25,15 minutos

Figura 10: **Tempo gasto para cálculo de matriz de co-ocorrência e 3 descritores de Haralick.**

d.3) Tempo de Treino e Teste - Rede Neural:

Tempo médio gasto para treino e teste da Rede Neural, alterando os hiperparâmetros de funções da bibliotecas *Keras*.

Alterando Quantidade de Camadas Ocultas

SO / Qtd. Camadas	1	2	3
Mac OS	22,40 segundos	25,75 segundos	28,75 segundos
Windows	31,21 segundos	36,42 segundos	46,40 segundos

Figura 11: **Tempo gasto para treino e teste da Rede Neural alterando a quantidade de camadas ocultas.**

A quantidade de camadas intermediárias foi muito influente no tempo de execução, mas quanto mais camadas foram adicionadas, pelo baixo volume de dados, a especificidade e sensibilidade tiveram a sua magnitude reduzida em 10%.

Alterando Otimizador

SO / Otimizador	Adam	NAdam
Mac OS	22,40 segundos	31,28 segundos
Windows	31,21 segundos	45,91 segundos

Figura 13: **Tempo gasto para treino e teste da Rede Neural alterando a função de otimização da Rede Neural.**

O tipo de otimizador é influente no tempo de execução da rede neural, sendo que a função Adam foi mais rápida que a NAdama, contudo não houve uma grande alteração na sensibilidade e especificidade após os testes com ambos otimizadores.

Alterando Quantidade de Épocas

SO / Qtd. Épocas	100	250	500
Mac OS	4,61 segundos	11,58 segundos	22,40 segundos
Windows	6,12	21,37	31,21

	segundos	segundos	segundos
--	----------	----------	----------

Figura 14: **Tempo para treino e teste da Rede Neural alterando a quantidade de épocas para o treinamento da Rede Neural.**

A quantidade de épocas é muito influente no tempo de execução da rede neural, contudo a sensibilidade e especificidade obtiveram seus valores prejudicados em cerca de 10% em relação às 500 épocas.

d.4) Tempo de Treino e Teste - SVM:

Tempo médio gasto para treino e teste da SVM, alterando os hiperparâmetros de funções da bibliotecas *Sklearn*.

Alterando Parâmetro C

SO / Valor C	1	50	100
Mac OS	0,2 segundos	0,2 segundos	0,2 segundos
Windows	0,2 segundos	0,2 segundos	0,2 segundos

Figura 15: **Tempo gasto para treino e teste da SVM alterando o parâmetro C.**

Alterando Parâmetro Gamma

SO / Valor Gamma	0,1	0,5	1
Mac OS	0,2 segundos	0,2 segundos	0,2 segundos
Windows	0,2 segundos	0,2 segundos	0,2 segundos

Figura 16: **Tempo gasto para treino e teste da SVM alterando o parâmetro Gamma.**

Alterando Grau do Polinômio do Hiperplano

SO / Grau Polinômio	3	6	10
Mac OS	0,2 segundos	0,2 segundos	0,2 segundos
Windows	0,2 segundos	0,2 segundos	0,2 segundos

Figura 17: **Tempo gasto para treino e teste da SVM alterando o grau do polinômio.**

As alterações de hiperparâmetros da SVM foram irrelevantes para o tempo de execução. Como o volume de dados foi pequeno, não houveram grandes alterações na sensibilidade e especificidade.

e) Resultados obtidos nos testes dos classificadores

Foram realizados conjuntos de testes para cada classificador considerando inicialmente os descritores energia, entropia e homogeneidade e posteriormente acrescentando o contraste.

Considerando para cada conjunto 10 execuções teste/treino, para a SVM foi obtida sensibilidade média de 0,628 (62,8%) e especificidade média de 0,66 (66%) utilizando 3 descritores. Para 4 descritores foi obtida sensibilidade média de 0,581 (58,1%) e especificidade média de 0,654 (65,4%).

Para a rede neural, foi obtida sensibilidade média de 0,627 (62,7%) e especificidade média de 0,65 (65%) para o conjunto de testes com 3 descritores. Para 4 descritores foi obtida sensibilidade média de 0,598 (59,8%) e especificidade média de 0,641 (64,1%).

Observa-se que ambos os classificadores obtiveram melhores métricas sem considerarem o contraste nos cálculos. No entanto, a amostra foi pequena e isso possivelmente impactou nessa diferença.

e) Exemplos de erros e acertos dos métodos.

e.1) Acertos SVM

Classe 1 p_d_left_cc(24).png

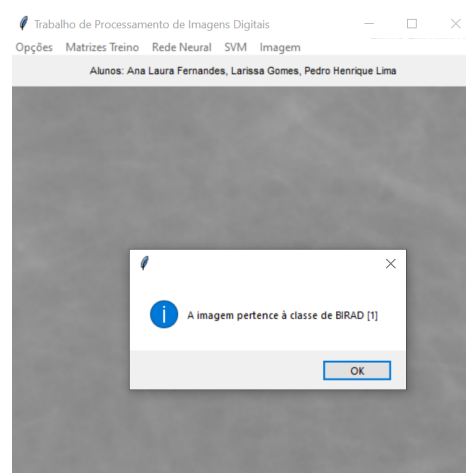


Figura 18: **Imagem BIRAD 1 classificada corretamente na SVM**

Classe 4 p_f_left_cc(36).png

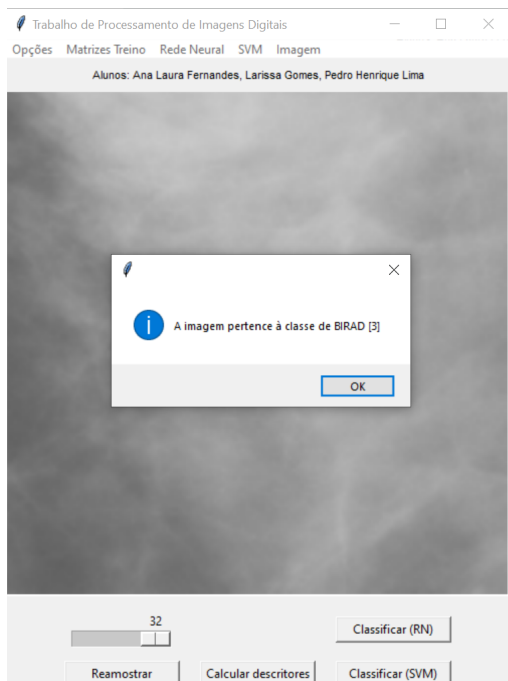


Figura 19: Imagem BIRAD 3 classificada corretamente na SVM

e.2) Erros SVM

Classe 4 p_g_left_cc(48).png

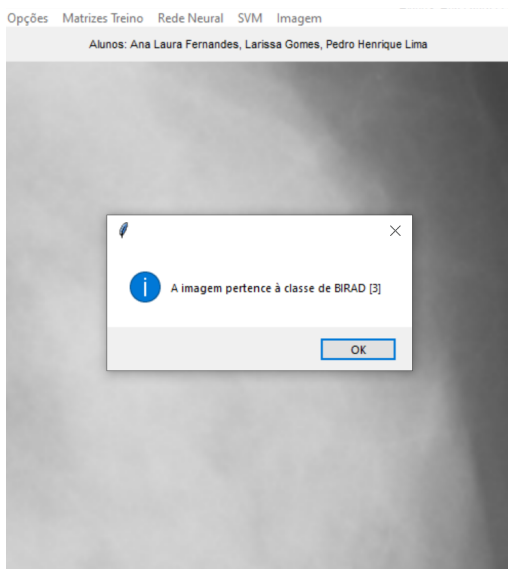


Figura 20: Imagem BIRAD 4 classificada incorretamente na SVM

Classe 2 p_e_left_cc(48).png

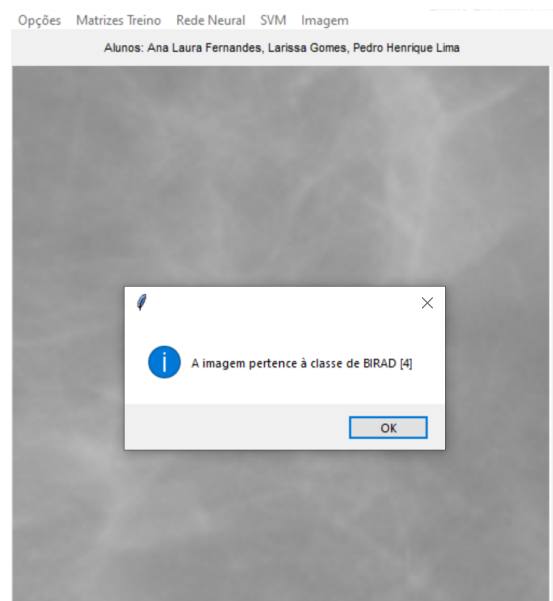


Figura 21: Imagem BIRAD 2 classificada incorretamente na SVM

e.3) Acertos Rede Neural

Classe 2 p_e_left_mlo(168).png

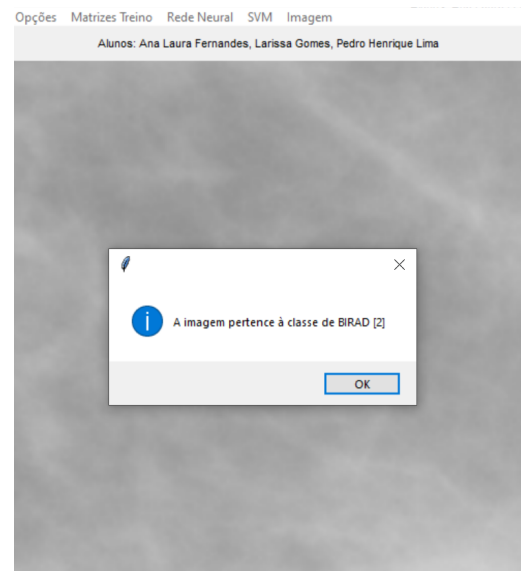


Figura 22: Imagem BIRAD 2 classificada corretamente na SVM

Classe 1 p_d_left_cc(108).png

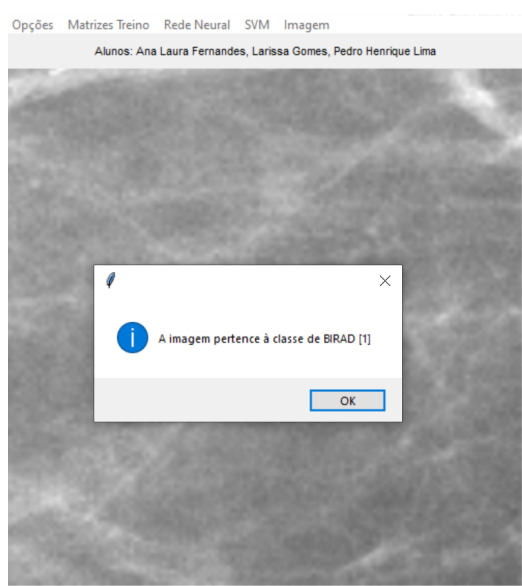


Figura 23: Imagem BIRAD 1 classificada corretamente na SVM

e.3) Erros Rede Neural

Classe 4 p_g_left_mlo(216).png

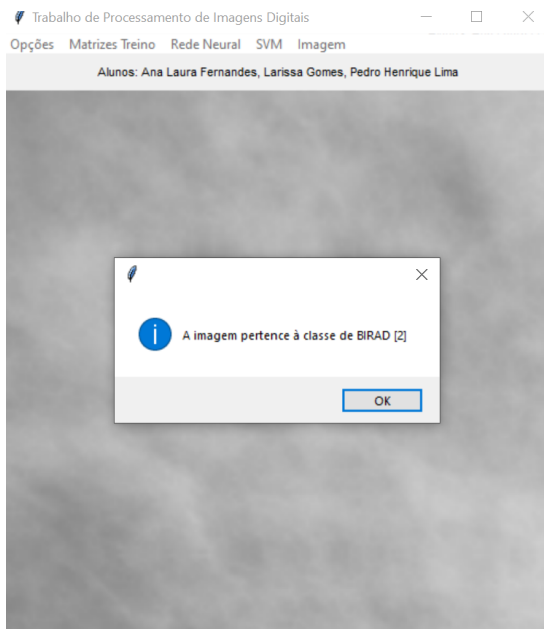


Figura 24: Imagem BIRAD 4 classificada corretamente na SVM

Classe 3 p_f_right_cc(24).png

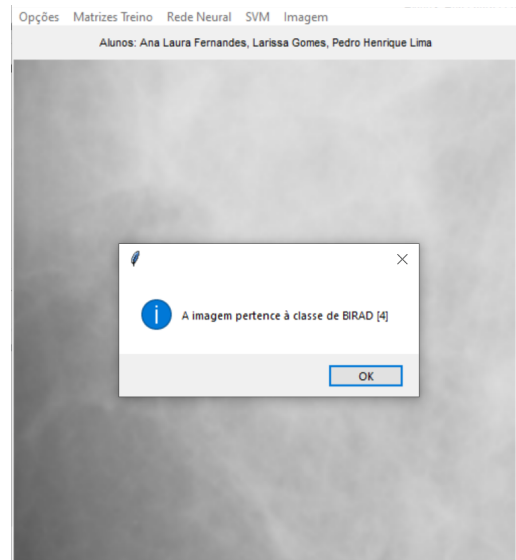


Figura 25: Imagem BIRAD 3 classificada corretamente na SVM

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, vol. 12, 2825--2830.
- [2] Chollet, François and others, 2015, Keras, <https://keras.io>.
- [3] Lundh, F. 1999. An introduction to tkinter. URL: www.pythonware.com/Library/Tkinter/Introduction/Index.Htm.
- [4] Nazari SS, Mukherjee P. An overview of mammographic density and its association with breast cancer. *Breast Cancer*. 2018 May. doi: 10.1007/s12282-018-0857-5
- [5] Northey JJ, Barrett AS, Acerbi I, et al. Stiff stroma increases breast cancer risk by inducing the oncogene ZNF217. *J Clin Invest*. 2020 Nov 2;130(11):5721-5737. doi: 10.1172/JCI129249. PMID: 32721948; PMCID: PMC7598051.
- [6] The American Cancer Society. Key Statistics for Breast Cancer. How common is breast cancer? URL: <https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html#:~:text=Overall%2C%20the%20average%20risk%20of,will%20never%20have%20the%20disease>. Accessed in : May 02 2022
- [7] Siu AL; U.S. Preventive Services Task Force. Screening for Breast Cancer: U.S. Preventive Services Task Force Recommendation Statement. *Ann Intern Med*. 2016 Feb 16;164(4):279-96. doi: 10.7326/M15-2886. Epub 2016 Jan 12. Erratum in: *Ann Intern Med*. 2016 Mar 15;164(6):448. PMID: 26757170.
- [8] Zhang H, Li W, Zhang H. An Image Recognition Framework for Oral Cancer Cells. *J Healthc Eng*. 2021 Oct 14;2021:2449128. doi: 10.1155/2021/2449128. PMID: 34691374; PMCID: PMC8536430.
- [9] Wang J, Liu X. Medical image recognition and segmentation of pathological slices of gastric cancer based on Deeplab v3+ neural network. *Comput Methods Programs Biomed*. 2021 Aug;207:106210. doi: 10.1016/j.cmpb.2021.106210. Epub 2021 May 29. PMID: 34130088.