

Aplicação de inteligência computacional para classificação de imagens utilizando descritores de textura.

Frederico Portilho¹, Sandro Roberto Fernandes²

¹Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais –
Campus Juiz de Fora
Rua Bernardo Mascarenhas, 1283 – Fábrica – 36.080-001 – Juiz de Fora – MG – Brasil

²Núcleo de Informática - Instituto Federal de Educação, Ciência e Tecnologia do
Sudeste de Minas Gerais – Campus Juiz de Fora

jgportilho@gmail.com, sandro.fernandes@ifsudestemg.edu.br

Abstract. *Back in 2015 was developed the SDPI software (Desenvolvimento de uma ferramenta computacional para classificação de padrões em imagens), in Instituto Federal Sudeste de Minas Gerais – Campus Juiz de Fora, with the function of extracting texture descriptors of a group of images. This paper shows how SDPI was enhanced with an Artificial Neural Network to train and test the descriptors extracted on the software. On developing was used the Encog library, and on validation three databases groups was tested, a breast cancer database, a group of four databases with several different texture formats and a database with two leaves species. All bases had the results expected on image classification literature, which proves the network as functional.*

Resumo. *Em 2015 no Instituto Federal Sudeste de Minas Gerais – Campus Juiz de Fora, foi desenvolvido o software SDPI (Desenvolvimento de uma ferramenta computacional para classificação de padrões em imagens), capaz de extrair descritores de textura de um grupo de imagens. Este trabalho apresenta como o SDPI foi aprimorado com uma rede neural artificial capaz de treinar e testar os descritores de textura extraídos pelo software. Para o desenvolvimento da rede foi utilizada a biblioteca Encog, e para sua validação foram testadas três grupos de dados, uma base de dados de câncer mamário, grupo de base de dados contendo diferentes formatos de textura e uma base de dados com duas espécies diferentes de folha. Todas as bases obtiveram resultados compatíveis com os esperados pela literatura, o que comprova o funcionamento da rede.*

1. Introdução

Chama-se classificação de imagens o problema onde um algoritmo deve ser capaz de reconhecer padrões em uma série de imagens, e distribuí-las em um número pré-determinado de grupos, com base na semelhança em seus padrões. A classificação de imagens é considerada um promissor campo na área de processamento de imagens digitais, encontrando aplicações em diferentes setores como agricultura, medicina e indústria, por exemplo. Infelizmente é também uma das mais difíceis tarefas, havendo diferentes abordagens que buscam diminuir o custo computacional e utilizar os dados extraídos em diferentes domínios de aplicações (PEDRINI; SCHWARTZ, 2008).

Dentre as possíveis diferentes maneiras de se classificar, uma delas é utilizando descritores de textura de Haralick (1973). Haralick definiu textura como uniformidade, densidade, aspereza, regularidade, intensidade, entre outras características da imagem.

Em 2015 no Instituto Federal Sudeste de Minas Gerais foi desenvolvido o software SDPI (Sistema de Detecção de Padrões em Imagens) tendo como função a partir da leitura de uma base de imagens, extrair um vetor de descritores de textura. O SDPI

utiliza como método uma matriz de co-ocorrência que retorna os seguintes descritores: Segundo Momento Angular, Entropia, Contraste, Variância, Correlação e Homogeneidade.

Entretanto, como é necessário um algoritmo de inteligência computacional para a classificação das imagens, o SDPI ainda era limitado por softwares de terceiros. Considerando as vantagens da inclusão de um algoritmo capaz de classificar as imagens no próprio software, como por exemplo torna-lo mais acessível a profissionais de diversas áreas, que é parte do objetivo do SDPI, foi decidido pela inclusão de uma Rede Neural Artificial (RNA) no próprio software.

Este trabalho apresenta como o software SDPI foi aprimorado com uma RNA, que a partir dos descritores de textura já gerados pelo software consiga identificar padrões em imagens e classifica-las em dois grupos distintos.

2. Fundamentação Teórica

2.1 Descritores de Textura

A textura é uma das principais características usadas pelo sistema humano para interpretar informações visuais (PEDRINI; SCHWARTZ, 2008). Métodos de classificação por textura se baseiam na criação de medidas qualitativas, como, por exemplo suavidade, rugosidade e regularidade, para classificar imagens inspirados pela visão humana.

Uma das abordagens mais diretas e simples para classificação de imagens por textura é através de histograma de níveis de cinza de uma imagem, ou região da imagem. Entretanto esse método apresenta limitações, pois não considera a posição relativa dos pixels em relação aos outros. Para solução desse problema o método utilizado utiliza medidas que mostram não somente a distribuição de intensidades, mas também as posições dos pixels com valores de intensidade iguais ou similares (FERNANDES; ASSIS, 2014)

O método utilizado cria uma matriz, chamado matriz intermediária, onde cada elemento representa a frequência com que um pixel de nível de cinza X e outro de nível de cinza Y ocorrem na imagem, separados por uma distância d e um ângulo θ (HARALICK et al., 1973). Exemplificado na Figura 01 abaixo.

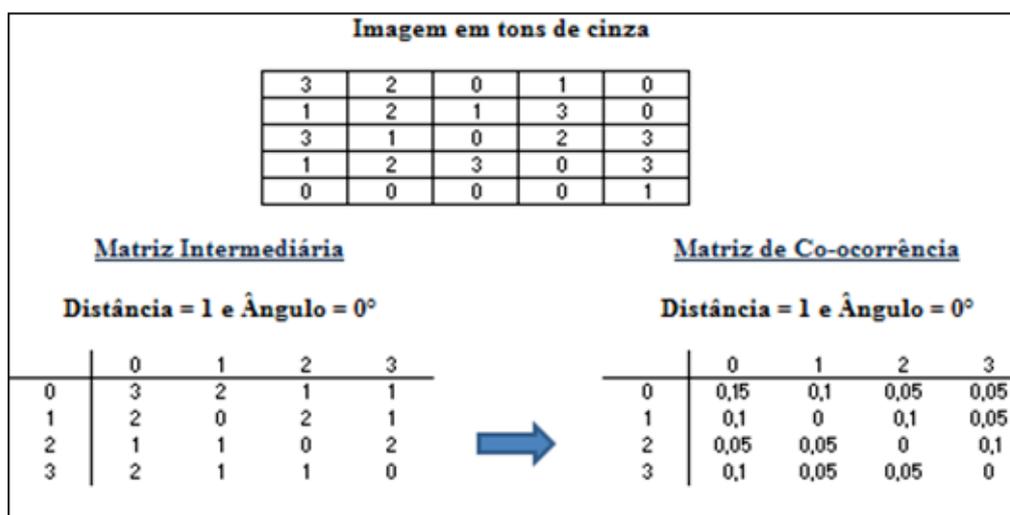


Figura 01. Matriz Intermediária e Matriz de Co-ocorrência.

Em seguida então é feita uma matriz de co-ocorrência, que mostra a probabilidade de determinado valor de cinza ocorrer na posição indicada. O SPDI utiliza por padrão o ângulo 0° e a distância de um pixel, uma vez que são os mais utilizados em exemplos encontrados na literatura.

Os resultados gerados pela matriz de co-ocorrência, podem ser utilizados para diversos cálculos estatísticos de segunda ordem, além de serem eficientes para a análise e classificação de imagens, mostrando ser uma assinatura de um padrão definido pela região de estudo. (HARALICK et al., 1973).

Haralick em seu estudo definiu quatorze medidas estatísticas, a serem calculadas a partir das matrizes de co-ocorrência. As mais relevantes entre elas são Segundo Momento Angular, Entropia, Contraste, Variância, Correlação e Homogeneidade (PEDRINI; SCHWARTZ, 2008).

2.2 Redes Neurais Artificiais

Uma Rede Neural Artificial é um modelo computacional inspirado pelo funcionamento do sistema nervoso central. Seu modelo mais simples, chamado *Perceptron*, consiste de um único neurônio que combina um vetor de entrada x recebido com um conjunto de pesos y gerando assim uma saída que classifica a entrada como pertencente a uma saída y binária.

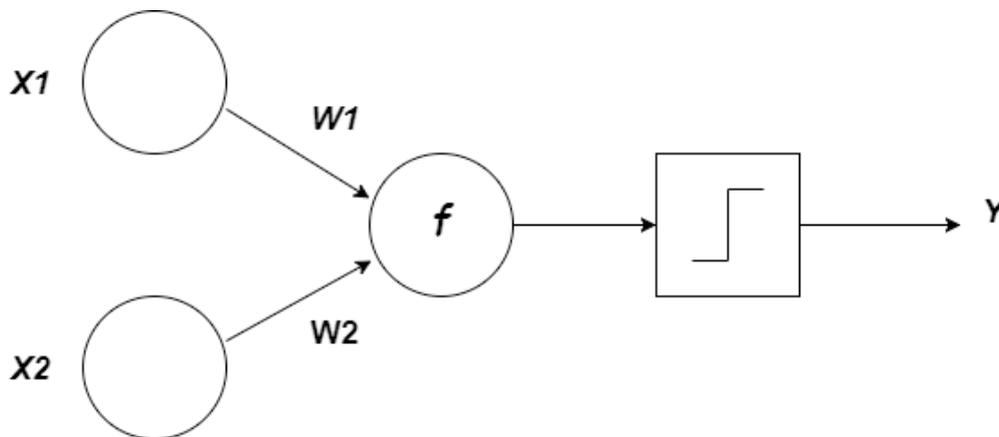


Figura 02. Estrutura de uma rede neural Perceptron.

Entretanto este é um modelo que apresenta limitações, como por exemplo, a necessidade do conjunto de dados ser linearmente separável. Devido a esse problema tarefas mais complexas utilizam modelos de *Perceptron* de múltiplas camadas (MLP).

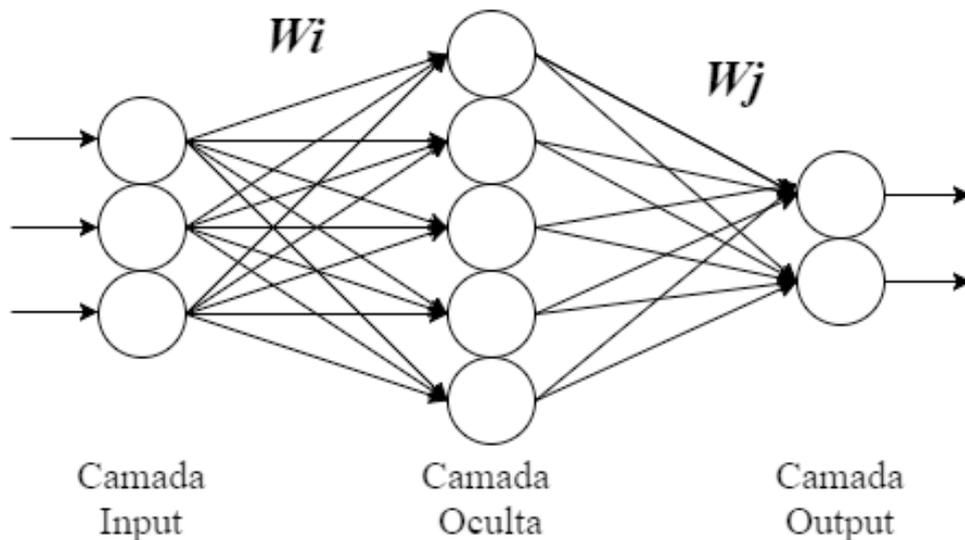


Figura 03. Estrutura de uma rede neural com múltiplos neurônios.

Como observado na Figura 03, a estrutura MLP possui três camadas sendo uma camada de Input, uma camada oculta e uma de output, onde cada uma está conectada a seguinte e possui uma função diferente. A camada de Input corresponde a entrada da rede possuindo um neurônio para cada atributo que a rede neural irá utilizar para a classificação. A camada oculta responsável pelo processamento da rede, quanto maior o número de neurônios, maior a adaptabilidade da RNA. E por fim a camada Output, a camada final da rede, aquela que recebe o resultado do processamento, possuindo um número de neurônios correspondente a quantidade de classes do problema.

Ainda é importante decidir sobre o algoritmo de treino da rede. O algoritmo mais comum de treino em redes de arquitetura *Feed-forward* (estrutura linear) é o *Backpropagation*. O *Backpropagation* consiste de duas etapas, a *propagação*, e a *retropropagação*. Na propagação a rede percorre seu caminho normal, no sentido dos inputs para os outputs, onde um vetor de entrada é aplicado aos nós da rede produzindo um conjunto de saídas como resposta. Logo após, na retropropagação, a rede percorre o caminho inverso, com os pesos sinápticos sendo ajustados de acordo com a regra de correção de erro. Então esse processo se repete até que o erro se encontre dentro de um limite aceitável.

Entretanto o *Backpropagation* tradicional possui algumas limitações, como com relação ao tempo de execução, por exemplo (RIEDMILLER; BRAUN 1993). Uma versão mais eficiente do *Backpropagation* é o *Resilient Backpropagation*.

O algoritmo de treino *Resilient Backpropagation* se destaca por possuir em cada peso um valor de atualização individual, sendo este valor adaptável, evoluindo durante a aprendizagem.

2.2.1 Validação cruzada

Durante o treinamento de uma RNA, os pesos sinápticos são ajustados da melhor maneira possível para atender ao conjunto de treinamento. Entretanto isso pode causar problemas com o modelo final da rede ficando excessivamente condicionado ao próprio conjunto treinado, situação essa que recebe o nome de *overfitting*.

Uma maneira comum de evitar o *overfitting* é a validação cruzada. A validação cruzada é uma técnica estatística que busca validar o modelo obtido durante o treinamento da rede, separando um conjunto de dados diferentes dos usados e usando-os como parâmetros durante o treinamento (HAYKIN, 2001). Dessa forma a base é dividida em dois conjuntos diferentes, sendo um o que será usado para o treinamento e o outro para validação, conseguindo assim resultados mais consistentes.

Uma das formas de se realizar a validação cruzada é o método *k-fold*. Nesse método a base é dividida em k partes, onde o modelo é treinado excluindo uma das partes para a validação. Esse processo é repetido então até que cada uma das partes k tenha sido usada para a validação. Ao fim do método elege-se o modelo que conseguiu o melhor desempenho, dessa maneira sendo possível através da validação cruzada eleger modelos mais eficientes.

3. Metodologia

No Instituto Federal Sudeste de Minas Gerais, campus Juiz de Fora, em 2014, Esmério e Fernandes desenvolveram um modelo para classificar imagens baseado em descritores de textura. Desenvolvido no software Matlab, da empresa Mathworks, o algoritmo programado utilizando uma matriz de co-ocorrência extraia descritores de textura de um vetor de imagens.

O trabalho através de testes comprovou a viabilidade do algoritmo de classificação de imagens através de descritores de textura, obtendo resultados compatíveis com os esperados pela literatura. Entretanto existem vantagens ao se programar um algoritmo de mais baixo nível como por exemplo Java ou c++, como, por exemplo, o uso de *multithreading*, desempenho e integração. Para isso em 2015 iniciou-se o processo de desenvolvimento do SDPI, com o objetivo de implementar em Java um software capaz de extrair descritores e treinar uma rede neural capaz de classificar imagens, e apresentá-lo em uma interface acessível a profissionais de outras áreas.

Em 2015, Teixeira e Fernandes desenvolveram a primeira parte do SDPI, implementando a funcionalidade de extrair os descritores de textura, usando como entrada um vetor de imagens. O algoritmo foi novamente testado, utilizando a mesma base de dados de imagens, e apresentando resultados semelhantes.

Com a funcionalidade de extração dos descritores resta a implementação da rede neural, que é apresentada nesse trabalho. Para o desenvolvimento da rede foi escolhida a biblioteca Encog. O Encog é uma biblioteca que permite a construção de aplicações que modelos são mais adequados a base de dados, o que permite que ela se comporte de maneira mais eficiente que muitas outras bibliotecas em Java (HEATON, 2015).

3.1 Encog

No desenvolvimento da rede foi utilizado o Encog. O Encog é uma biblioteca que dispõe ferramentas automatizadas para seleção de modelo, além de oferecer eficientes algoritmos *multithred* o que reduzem o tempo de processamento (HEATON, 2015). Essas características permitem que o Encog funcione de maneira mais eficiente que maioria das outras bibliotecas em Java e C#.

A base de dados é codificada pelo Encog, em um formato chamado *VersatileMLDataSet*, a partir de um arquivo CSV. Nele são definidos quais são as entradas da rede neural, e também qual a saída esperada.

Em seguida a base de dados é associada a uma classe *EncogModel*. Essa classe, utilizando a base de dados como referência cria a estrutura da rede, definindo o número

de neurônios em cada camada. É nessa classe que também se encontram os métodos auxiliares para a normalização da base de dados, e para o treinamento da rede.

O próximo passo é o treino da rede neural, onde utiliza-se também a classe *EncogModel*. O Encog foi programado para treinar a rede com um modelo de validação cruzada, onde o treino é executado cinco vezes sendo armazenado o melhor resultado. O algoritmo de treino utilizado foi *Resilient Backpropagation* e a função de ativação foi a tangente hiperbólica.

Por tratar-se de um problema de classificação, onde as entradas devem ser associadas a um valor não numérico, o resultado do treino deve ser armazenado em uma classe *MLClassification*. É nela que se encontram os valores referentes aos pesos e a estrutura da rede, como demonstrado na Figura 04, e esse o arquivo que é salvo para o uso posterior da rede.

```
encog, BasicNetwork, java, 3.3.0, 1, 1499020508634
[BASIC]
[BASIC:PARAMS]
[BASIC:NETWORK]
beginTraining=0
connectionLimit=0
contextTargetOffset=0,0,0
contextTargetSize=0,0,0
endTraining=2
hasContext=f
inputCount=6
layerCounts=2,11,7
layerFeedCounts=2,10,6
layerContextCount=0,0,0
layerIndex=0,2,13
output=-0.6890132151,0.8690586665,-0.9841589835,-0.8963321457,-0.4344011502,-0.9999999912,-0.9714461261,0.9127792208,-0.9992571992,-1,1,0.9280305521,1,-1,1,
outputCount=2
weightIndex=0,22,92
weights=1.1774848684,1.0652957984,0.7135335154,0.4282319368,0.5533931766,-0.5287692399,-2.3303392181,-0.5028478485,-1.0363248154,0.8311710037,0.4595661052,-
biasActivation=0,1,1
[BASIC:ACTIVATION]
"org.encog.engine.network.activation.ActivationTANH"
"org.encog.engine.network.activation.ActivationTANH"
"org.encog.engine.network.activation.ActivationLinear"
```

Figura 04. Classe MLClassification

É também na classe *MLClassification* que se encontra o método *Classify* que é usado tanto para a validação quanto para a classificação de novas imagens.

3.2 SDPI

Ao SDPI foram acrescentadas duas funções. Sendo a primeira a leitura de um vetor de descritores para o treino de uma rede neural, e a segunda o uso de uma rede previamente treinada para classificação de um novo grupo de imagens.

3.2.1 Ler um vetor de descritores e treinar uma rede neural

A primeira função deve ser utilizada quando o objetivo é a treinar a rede a partir dos descritores extraídos pelo software.

Inicialmente seleciona-se o arquivo de texto que contém os descritores de textura previamente extraídos pelo software. Conforme mostra a Figura 05 abaixo.

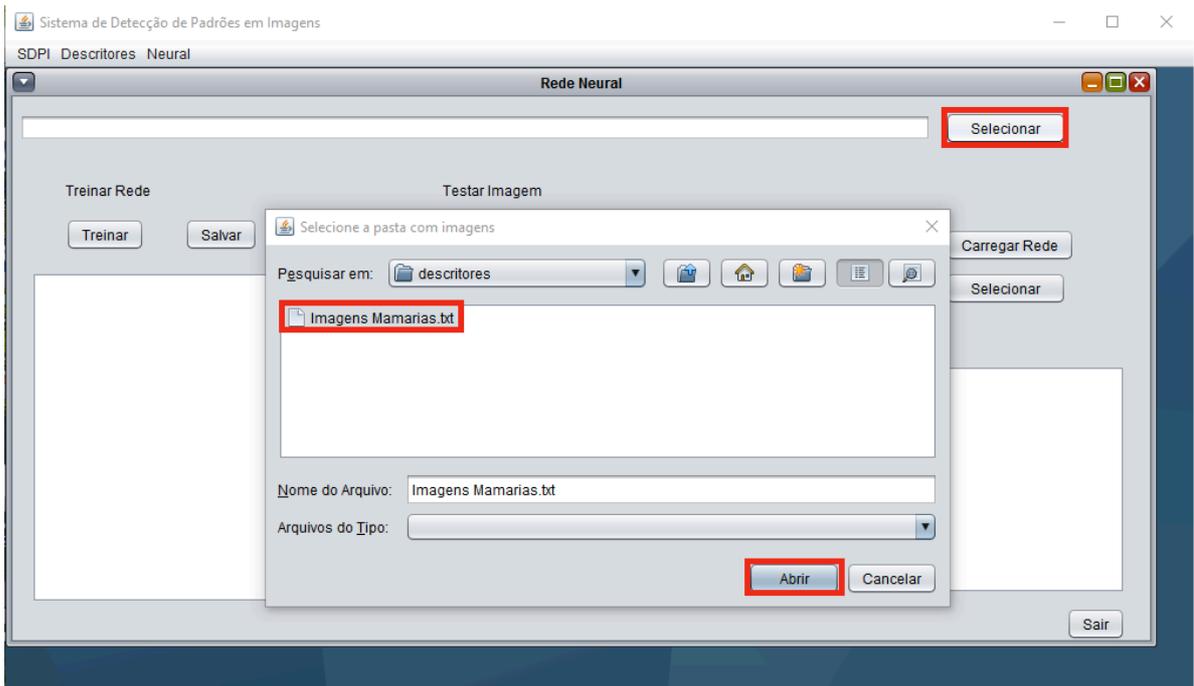


Figura 05. Selecionando o arquivo que contém os descritores.

Após a seleção basta clicar no botão treinar para o resultado ser exibido na tela, como mostrado na Figura 06. Caso haja interesse em usar novamente a rede treinada para classificação de novas imagens, basta clicar em salvar.

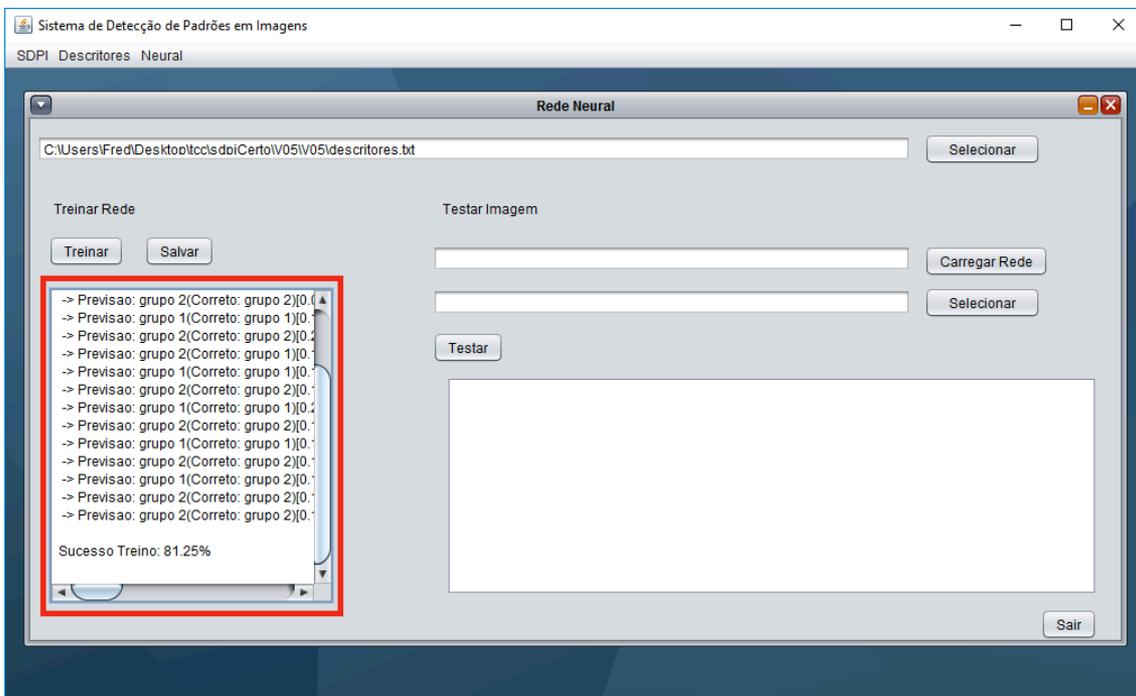


Figura 06. Resultado Treinamento.

3.2.2 Verificar a qual grupo pertence uma imagem utilizando uma rede previamente treinada.

É possível no SDPI usar uma rede previamente treinada para classificar novas imagens. Para isso basta clicar em carregar rede e selecionar ao arquivo com extensão *eg*, aonde estão contidas as informações da rede neural salva pelo Encog.

Além disso, também é necessário selecionar a pasta com as imagens a serem classificadas. Após isso basta clicar no botão Testar. O resultado exibido pelo software é exemplificado na Figura 07.

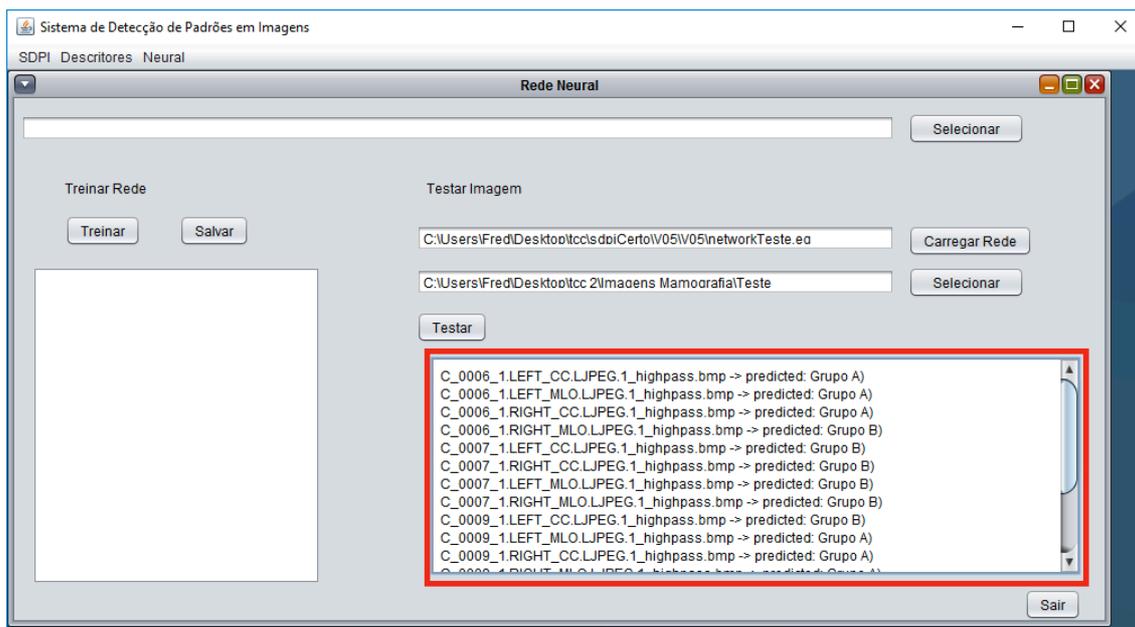


Figura 07. Resultado da Classificação

4. Testes e Resultados

Para os testes da ferramenta e validação da rede neural artificial desenvolvida foram escolhidos três grupos de dados. O primeiro uma base de dados de câncer mamário cedida por pelo site da *University of South Florida - Digital Mammography*. O segundo conjunto é um grupo de imagens contendo diferentes tipo possíveis de textura encontradas na URL <http://www.cb.uu.se/~gustaf/texture>. O terceiro grupo é uma base de dados com duas espécies de folhas diferentes.

4.1 Base de dados de câncer mamário

A base de dados de câncer mamário foi a mesma utilizada no trabalho Desenvolvimento de uma ferramenta computacional para classificação de padrões em imagens (TEIXEIRA; FERNANDES, 2015) que apresenta como foi desenvolvida a função de extrair descritores do SDPI.

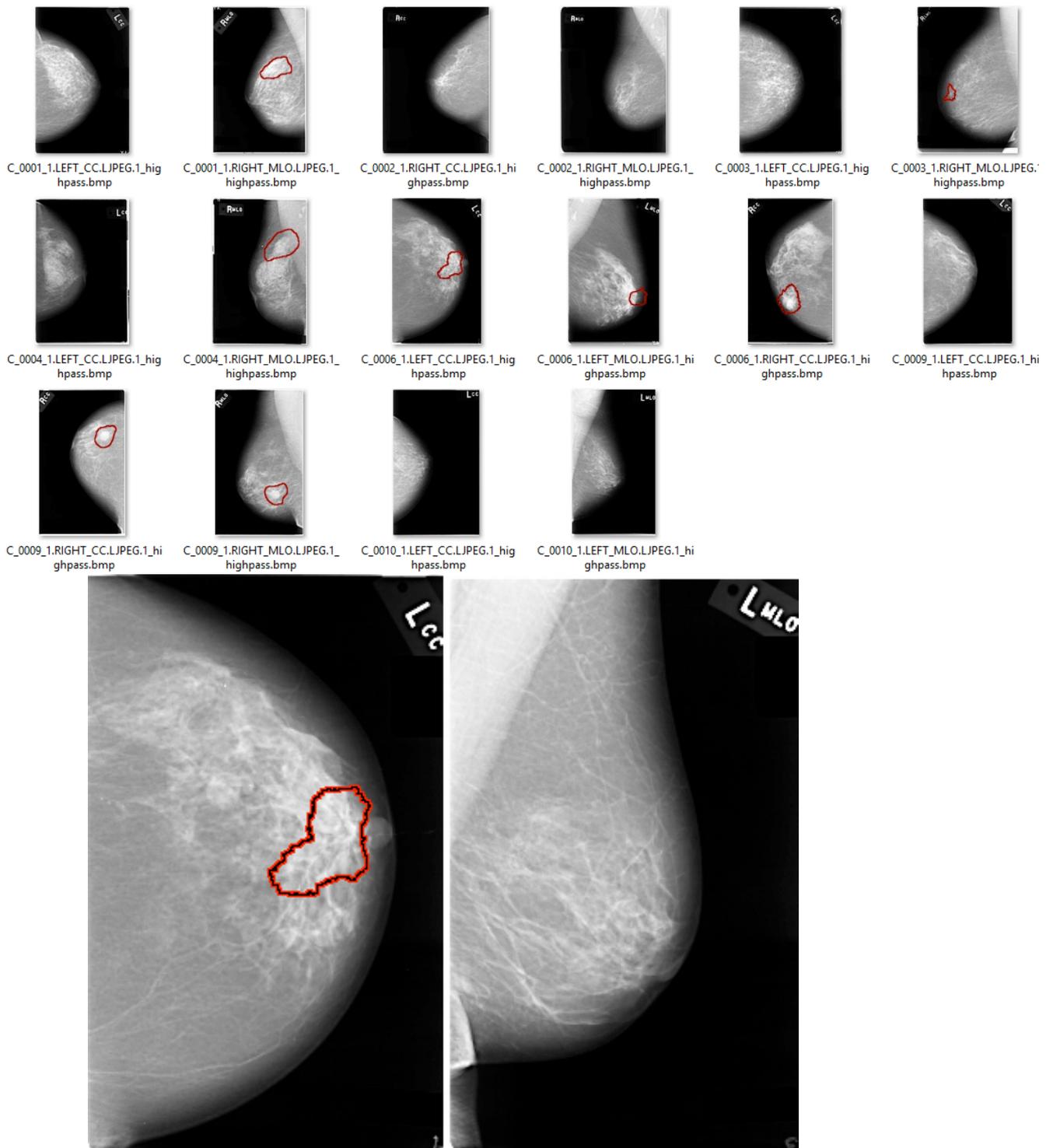


Figura 08. Amostra do grupo de imagens.

Foram utilizadas 16 imagens para o treinamento da rede, sendo essas oito imagens com lesão e oito imagens sem lesão, e a porcentagem de acerto obtido durante a validação utilizando as mesmas imagens usadas no treino foi de 87.5%. Embora o resultado seja inferior ao obtido nos testes anteriores feitos no MatLab por Esmerio e

Fernandes, que foi de 100%, uma das possíveis consequências da diferença de resultados pode ser atribuída ao uso de um maior número de imagens nos testes feitos em Java, uma vez que em testes anteriores foram utilizadas apenas oito imagens, diferente dos testes executados no próprio SDPI onde foram adicionadas oito imagens novas além das oito anteriores.

Tabela 1. Comparação entre Java e MatLab para imagens de câncer mamário o mesmo grupo de imagens do treino

Treinamento	Numero de Imagens	Numero de Acertos	Proporção
Mat Lab Imagens com Lesão	4	4	1
Mat Lab Imagens sem Lesão	4	4	1
Mat Lab Total de Imagens	8	8	1
Java Imagens com Lesão	8	7	0,875
Java Imagens sem Lesão	8	7	0,875
Java Total de Imagens	16	14	0,875

A fim de obter resultados mais consistentes, também foram feitos testes utilizando um grupo de imagens diferente daquelas usadas no treino da rede. Para esses testes foi usado um total de 16 imagens, sendo 12 imagens com lesão e 4 imagens sem lesão. Nos testes realizados o SDPI obteve uma taxa de acerto de 68,7%, em comparação aos 87,5% obtidos nos testes usando o MatLab.

Tabela 2. Comparação entre Java e MatLab para imagens de câncer mamário utilizando um grupo de imagens diferente

Testes	Número de Imagens	Número de Acertos	Proporção
Mat Lab Imagens com Lesão	4	4	1
Mat Lab Imagens sem Lesão	4	3	0,75
Mat Lab Total de Imagens	8	7	0,875
Java Imagens com Lesão	12	8	0,67
Java Imagens sem Lesão	4	3	0,75
Java Total de Imagens	16	11	0,69

4.2 Base de dados de textura

Como é objetivo do SDPI ser utilizado em diferentes aplicações da área de classificação de imagens, é importante que ele seja avaliado em diferentes tipos de problema. Por esse motivo foram escolhidas quatro grupos de bases de dados, contendo diferentes formatos de textura.

Tais bases foram escolhidas por tratarem-se de problemas simples onde seriam esperados resultados positivos, visando assim avaliar o funcionamento do software em diferentes possíveis problemas envolvendo textura.

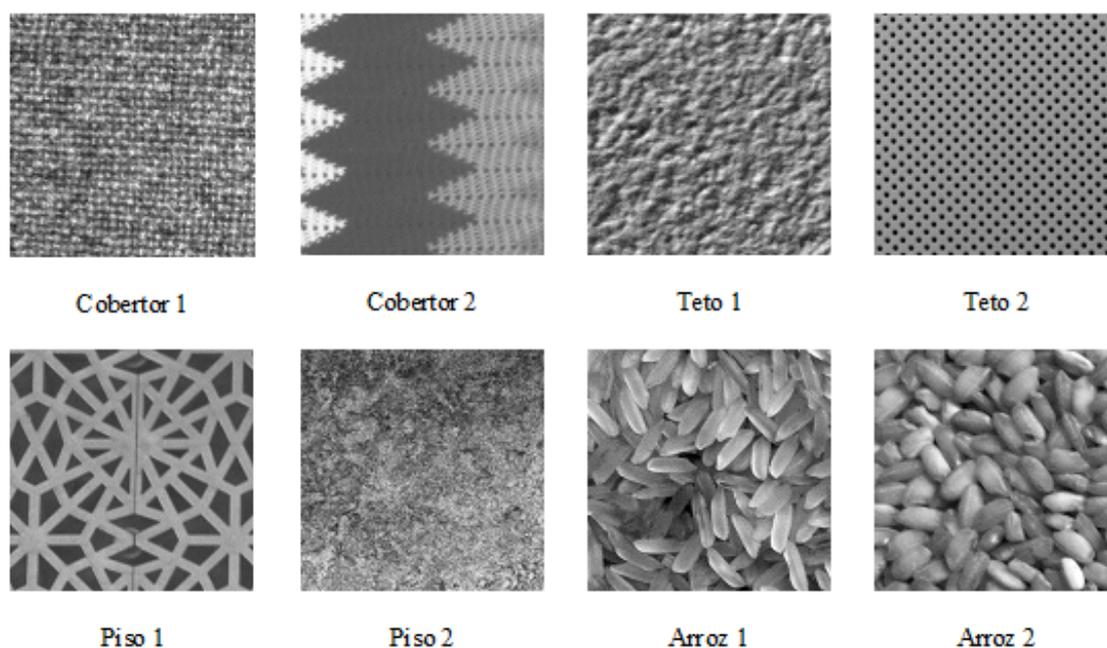


Figura 09. Imagens contendo diferentes tipos de textura.

As bases foram avaliadas par a par em seus respectivos grupos, sendo sempre o tipo 1 comparado ao tipo 2. Cada grupo consistia em total de 256 imagens, com 128 pertencendo ao tipo 1 e 128 ao tipo 2.

Nos testes realizados usando a mesma base de dados do treino, foi obtido um resultado de 100% nos grupos Cobertor, Teto e Piso e um acerto de 99.6% no grupo Arroz, que considerando o número de imagens usadas também pode ser aproximado a 100%.

Tabela 3. Resultados de testes em bases de textura usando o mesmo grupo de imagens do treino

Treino	Número de Imagens	Número de Acertos	Proporção
Cobertor 1	128	128	1
Cobertor 2	128	128	1
Cobertor Total	256	256	1
Teto 1	128	128	1
Teto 2	128	128	1
Teto Total	256	256	1
Piso 1	128	128	1
Piso 2	128	128	1
Piso Total	256	256	1
Arroz 1	128	128	1
Arroz 2	128	127	0,992
Arroz Total	256	255	0,996

Para os testes com outras imagens foram utilizadas 128 imagens de cada grupo, sendo sempre 64 do tipo 1 e 64 do tipo 2. Nos grupos Cobertor, Teto e Piso a taxa de acerto foi de 100%, já no grupo Arroz o resultado foi de 95%.

Tabela 4. Resultados de testes em bases de textura usando um grupo de imagens diferente do treino.

Teste	Número de Imagens	Número de Acertos	Proporção
Cobertor 1	64	64	1
Cobertor 2	64	64	1
Cobertor Total	128	128	1
Teto 1	64	64	1
Teto 2	64	64	1
Teto Total	128	128	1
Piso 1	64	64	1
Piso 2	64	64	1
Piso Total	128	128	1
Arroz 1	64	58	0,906
Arroz 2	64	64	1
Arroz Total	128	122	0,953

4.3 Base de dados de Folha

Comparar espécies de folhas é um problema bastante comum em classificação de imagens. Por esse motivo foi escolhida uma base de dados comparando folhas da espécie *Ulmus carpinifolia* com a espécie *Salix aurita*.



Figura 10: Amostra imagens de folhas

Para o treinamento foram utilizadas 100 imagens, sendo 50 da espécie *Ulmus carpinifolia* e 50 da espécie *Salix aurita*. Os resultados gerais da validação foram de 87%.

Tabela 5. Resultados de testes em folhas usando o mesmo grupo de imagens do treino

Treinamento	Número de Imagens	Número de Acertos	Proporção
<i>Ulmus carpinifolia</i>	50	41	0,82
<i>Salix aurita</i>	50	46	0,92
Folha Total	100	87	0,87

Para os testes foram usadas 50 imagens, 25 da espécie *Ulmus carpinifolia* e 25 da espécie *Salix aurita*. Obtendo uma taxa de acerto de 58%.

Tabela 6. Resultados de testes em folhas usando um grupo de imagens diferentes do treino

Treinamento	Número de Imagens	Número de Acertos	Proporção
<i>Ulmus carpinifolia</i>	25	10	0,4
<i>Salix aurita</i>	25	19	0,76
Folha Total	50	29	0,58

5. Conclusões e Trabalhos Futuros

Através dos testes realizados no SDPI foi observado o funcionamento do software, e que este pode ser utilizado em diversas tarefas envolvendo descritores de textura. Os resultados também se mostraram compatíveis os testes previamente realizados no MatLab e com os esperados pela literatura.

Embora o resultado atingido tenha sido o esperado entende-se que ainda existe melhorias a ser feitas que poderiam melhorar o seu funcionamento. Como por exemplo a implementação de Deep Learning, um algoritmo de treinamento mais moderno onde cada entrada de dados na rede neural é considerada para o treinamento da entrada seguinte.

O software também se beneficiaria da inclusão de filtros capazes de tratar impurezas das imagens, além de testes com mais bases de dados a fim de um grau de informação maior sobre qual a expectativa de resultados em tipos diferentes de bases de dados.

Também é objetivo que o SDPI funcione como ferramenta de auxílio a profissionais de diversas áreas, e não como substituto a esses profissionais, havendo-se, portanto, ainda a necessidade de um especialista, para que os resultados obtidos sejam os de sua melhor forma.

Seguindo essas recomendações o SDPI pode ser considerado uma útil ferramenta na área de classificação de padrões em imagens.

References

- DDSM – Digital Database for Screening Mammography; University of South Florida - Maio 2008. Disponível em: <http://marathon.csee.usf.edu/Mammography/Database.html> Acesso em 14 jan. 2008.
- Esmério, w; Fernandes, s. Classificação de padrões em imagens utilizando descritores de textura. Juiz de fora: trabalho de conclusão de curso, 2014.
- Heaton, J. (2015). Encog: Library of Interchangeable Machine Learning Models for Java and C#. *Journal of Machine Learning Research*, 16, 1243-1247.
- G. Kylberg. The Kylberg Texture Dataset v. 1.0, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, External report (Blue series) No. 35. Available online at: <http://www.cb.uu.se/~gustaf/texture/> Acesso em 20 jun. 2017
- Hornik, k Multilayer Feedforward Networks are Universal Approximators. University of California, San Diego 1989. *Neural Networks*, Vol. 2, pp. 35Y-366, 198Y Printed in the USA.
- Heaton, Jeff. “Programming Neural Networks with Encog3 in Java”. Heaton Research, Inc, 2011.
- Teixeira, e; Fernandes, s. Desenvolvimento de uma ferramenta computacional para classificação de padrões em imagens. Juiz de fora: trabalho de conclusão de curso, 2014.
- Haralick, r. M.; shanmugam, k.; dinstein, i. “Textural features for image classification”. In *iee transactions on systems, man and cybernetics*, v.3, n. 6, 1973. Anais... P. 610-621.
- Pedrini, h.; schwartz, w. R. “Análise de imagens digitais: princípios, algoritmos e aplicações”. São paulo: thomson learning, 2008.
- Fernandes, s. Caracterização de imagens de microtomografia de raios x baseada em descritores de textura. 2012. 102 f. Tese (doutorado em modelagem computacional): iprjuerj, nova friburgo, rj, 2012.
- Haykin, s. “Redes Neurais: Princípios e Práticas”. Bookman Companhia Ed, 2001.
- Riedmiller, m.; Braun, h. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. University of Karlsruhe, 1993.
- Guimarães, a.; Mathias, i; Dias, a; Ferrari j; Cazelatto , c. Módulo de Validação Cruzada para treinamento de Redes Neurais Artificiais com algoritmos Backpropagation e Resilient Propagation. *UEPG Ci. Exatas Terra, Ci. Agr. Eng.*, Ponta Grossa, 14 (1): 17-24, abr. 2008.
- Oskar J. O. Söderkvist, “Computer vision classification of leaves from swedish trees,” Master’s Thesis, Linkoping University, 2001.