



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA



MARIANA ALAMINO CAMOLESI

Treinamento de autômatos celulares para reconhecimento de padrões em imagens de textura

Campinas
21/08/2020

MARIANA ALAMINO CAMOLESI

Treinamento de autômatos celulares para reconhecimento de padrões em imagens de textura*

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação do(a) Prof. Dr. João Batista Florindo.

*Este trabalho foi financiado pelo Instituto Serrapilheira, projeto Agosto/2019.

Resumo

Neste projeto, o problema de classificação de imagens de textura é abordado. Para tratar este problema, dois métodos são utilizados. O primeiro método combina conceitos de modelagem de corrosão e autômatos celulares para gerar um descritor de textura, definido pela massa cumulativa de produto corroído a cada iteração. O segundo método parte do treinamento de uma rede neural convolucional, aplicando esta mesma rede sobre as imagens para extrair os vetores de descritores, os quais contêm os valores calculados em cada neurônio na penúltima camada da rede. No final, ambos descritores são usados para classificação de imagens e seus resultados são comparados. Os resultados foram satisfatórios, exibindo uma maior taxa de acerto para os descritores extraídos pela rede neural, confirmando a eficiência das redes neurais convolucionais na análise de imagens.

Conteúdo

| | | |
|----------|---------------------------------------------------------------------------|-----------|
| 1 | Introdução | 5 |
| 2 | Redes Neurais | 6 |
| 2.1 | Uma ideia geral sobre as redes neurais artificiais | 6 |
| 2.2 | Redes neurais convolucionais | 7 |
| 3 | Autômatos celulares inspirados em corrosão | 10 |
| 3.1 | Corrosão localizada de metais | 10 |
| 3.2 | Autômatos celulares (CAs) | 10 |
| 3.2.1 | Estado das células | 11 |
| 3.2.2 | Tipos de vizinhança | 11 |
| 3.3 | Descritores CITA (<i>Corrosion-inspired texture analysis</i>) | 12 |
| 4 | Metodologia | 13 |
| 4.1 | Extraindo os descritores CITA | 14 |
| 4.2 | Extraindo os descritores com a rede neural | 14 |
| 5 | Resultados | 15 |
| 6 | Conclusão | 18 |

1 Introdução

Inspiradas no cérebro humano, as redes neurais artificiais são sistemas computacionais aptos a reconhecer padrões complexos, com uma grande capacidade de "aprenderem sozinhas". Essas redes usam diferentes camadas de processamento para entender e aprender cada vez mais as informações dadas, com aplicações que vão desde auxílio em diagnósticos médicos até reconhecimento de faces em imagens [4]. Graças à flexibilidade de uso e eficiência nos resultados, o interesse nas redes neurais vem crescendo significativamente, em especial nas redes neurais convolucionais. Essas redes têm demonstrado alto desempenho na classificação de imagens, como por exemplo na análise de imagens médicas [5].

Esse projeto tem como foco a abordagem das redes neurais em classificação de imagens de textura, em especial, reconhecimento de padrões nessas imagens. O campo de reconhecimento de padrões trabalha com a descoberta automatizada de padrões e regularidades nos dados por meio do uso de algoritmos de computador, fazendo uso desses padrões para realizar ações como classificar os dados em diferentes categorias [1].

A classificação é feita a partir de vetores de características, descritores responsáveis por carregar informações sobre a imagem. Ao longo do projeto, serão utilizados o método de descritores por corrosão de imagens apresentado no artigo "*Improved texture image classification through the use of a corrosion-inspired cellular automaton.*" [3], além do método de extração de descritores por redes neurais, ambos aplicados na base de imagens de textura de *Brodatz*.

2 Redes Neurais

2.1 Uma ideia geral sobre as redes neurais artificiais

Redes neurais são modelos computacionais compostos por nós interconectados que transmitem informações processadas de forma similar aos neurônios do cérebro humano. São principalmente utilizadas para criar sistemas de inteligência artificial, capazes de aprender informações e padrões com a base de dados. [9]

A estrutura de uma rede neural consiste em 3 tipos principais de camadas: a camada de entrada (para a classificação de imagens deste projeto, a entrada dos neurônios são os valores RGB/Escala cinza de cada pixel), camadas intermediárias responsáveis pelo processamento das informações (camadas ocultas) e a camada final (resultados da rede).

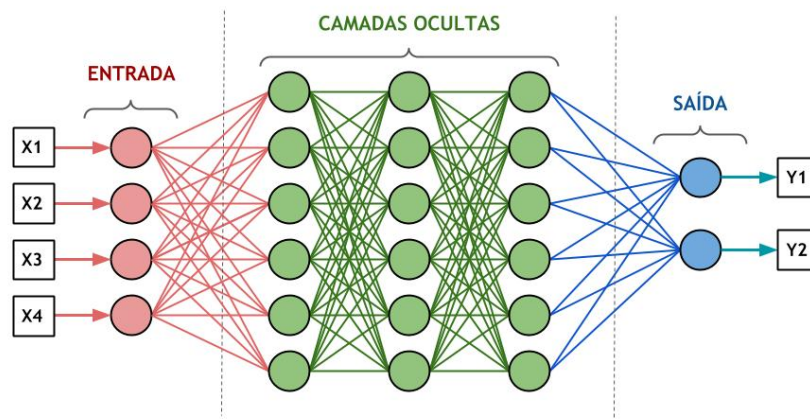


Figura 1: Esquema simplificado de uma rede neural.

Cada neurônio recebe todos os valores das entradas, que são multiplicadas por pesos w e somadas entre si junto com um *bias* b . A soma ponderada $z = \sum_{i=1}^N x_{i,j} w_i + b$ é utilizada em uma função de ativação $a = \sigma(z)$ (ex.: sigmoide, ReLU, etc.) que, por sua vez, determina o valor de saída do neurônio.

O objetivo é a rede aprender quais valores de pesos e *biases* entregam os melhores resultados desejados. Para isso, o modelo faz uso de uma função de custo, que serve para quantificar a qualidade da rede em prever as saídas já conhecidas, a partir de uma base de treinamento. O objetivo é minimizar essa função através do *método do gradiente descendente* para o mais próximo possível de zero e ajustar os pesos e *biases* a partir do

algoritmo de *backpropagation* [7]. Assim, a cada etapa, os parâmetros são atualizados até atingir o critério de parada escolhido e esses pesos são utilizados em uma base de teste, cujas saídas não são conhecidas, afim de melhorar a generalização da rede.

Infelizmente, esse tipo de rede não é a mais adequada para o problema de classificação de imagens. Devido ao fato dessas redes não levarem em conta a estrutura espacial das imagens, tratando pixels tanto próximos quanto distantes da mesma forma, sua performance se torna insuficiente. Assim, temos outra arquitetura de rede que vem sendo aplicada com sucesso no processamento e análise de imagens digitais.

2.2 Redes neurais convolucionais

Quando se trata de classificação de imagens, as chamadas Redes Neurais Convolucionais (CNN) são preferidas. Isso se deve ao fato de que, diferente da arquitetura de redes neurais vista anteriormente, a arquitetura da CNN considera a estrutura espacial de uma imagem. Essas redes são capazes de aprender sozinhas filtros e características sobre as imagens, atribuindo importância a diversos aspectos da imagem, o que torna o pre-processamento muito menor quando comparada a outros algoritmos de classificação de imagens [6]. As redes convolucionais são inspiradas nos processos biológicos da organização do córtex visual dos animais. Elas são organizadas por 3 ideias básicas:

- **Campos receptivos locais:** Um conjunto de neurônios de entrada, tamanho escolhido pelo usuário, conectado a cada neurônio da próxima camada oculta. Para cada campo receptivo local, existe um neurônio oculto diferente. Esse processo é realizado para todos os campos da entrada, movendo para o lado com um tamanho de passo pré-definido. Cada conexão aprende um peso e o neurônio oculto aprende um *bias* geral. Esquema exemplificado na Figura 2.
- **Pesos e *biases* compartilhados:** São através dos campos receptivos locais que os *filtros* agem nas imagens e geram os chamados *features maps*. Um filtro é definido pelos pesos e vieses compartilhados, responsável por mapear toda a imagem e “devolver” características dela. Ou seja, o filtro passa por toda a imagem, através de campos receptivos locais, e por convolução gera um mapa de características (*feature map*).

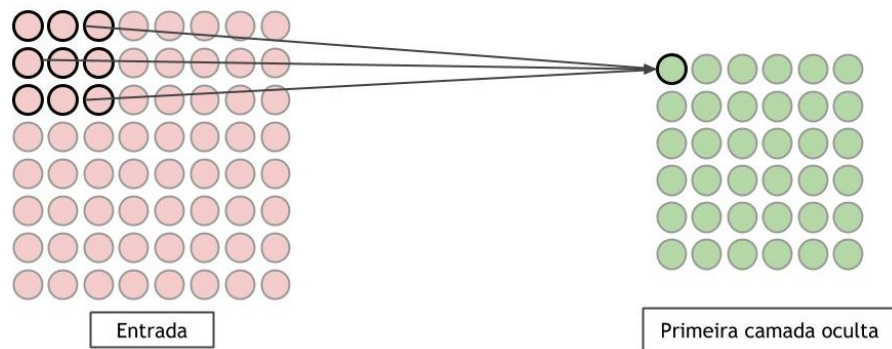


Figura 2: Campo receptivo local de tamanho 3x3.

- **Condensamento (pooling):** A camada de condensamento (*pooling layer*) tem por objetivo combinar características similares escolhendo uma unidade responsável por resumir a informação da área (submatriz) em um único valor, gerando uma nova matriz de menor dimensão. Existem alguns métodos para isso, sendo um deles o *max pooling*, no qual apenas o maior número da unidade é passado para a saída. Ou então o *average pooling*, que tem como saída o valor médio das entradas das submatrizes.

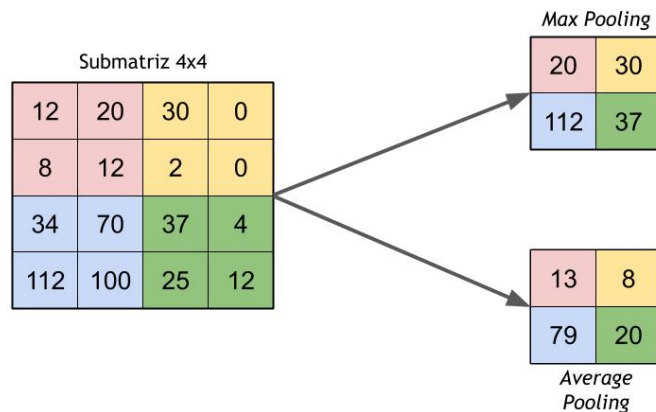


Figura 3: Representação de dois tipos de *pooling* em uma submatriz 4x4.

O processo de extrair características pelos *feature maps*, aplicar convoluções e condensar matrizes diversas vezes caracteriza o chamado *deep learning*, também conhecido como aprendizagem profunda.

Geralmente as arquiteturas das redes neurais convolucionais (Figura 4) intercalam as camadas convolucionais e camadas de condensamento, cujo papel é a extração dos descritores. As últimas camadas ficam responsáveis pela classificação, geralmente contendo um vetor de neurônios totalmente conectados (*fully connected layer*) com os valores de saída.

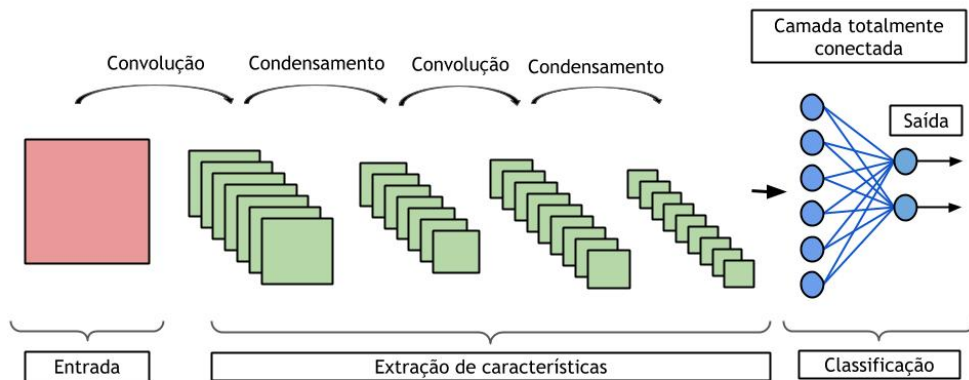


Figura 4: Esquema simplificado de uma rede neural convolucional.

Apesar da eficiência das redes neurais convolucionais, estas redes exigem uma larga base de dados para um treinamento bem feito. Entretanto, nem sempre há essa disponibilidade de dados. Afim de contornar esse problema, utiliza-se uma técnica chamada *Transfer Learning*. Nela, a rede é pré-treinada em um conjunto de dados bem maior (ex.: ImageNet para classificação de imagens) e, a partir dos pesos aprendidos nesse conjunto, segue-se para o treinamento no conjunto de dados de interesse. Esse procedimento de transferência de conhecimento pode ser realizado de duas formas: pelo "ajuste fino", em que a inicialização dos pesos é feita a partir dos pesos otimizados da rede pré-treinada; ou pela "extração fixa de características", em que se congelam todos os pesos da rede, exceto os da última camada. Essa camada é substituída por uma nova com pesos aleatórios e apenas ela é treinada. Para este trabalho, escolheu-se trabalhar com o ajuste fino, partindo da rede pré-treinada Resnet50.

3 Autômatos celulares inspirados em corrosão

O pré-processamento das imagens usado neste trabalho se dá por meio de autômatos celulares (CAs) inspirados no fenômeno de corrosão da superfície de metais, que posteriormente dá origem ao descritor CITA (*Corrosion-Inspired Texture Analysis*). Os mecanismos básicos por trás dessa reação que ocorre entre os metais e seu ambiente servem como inspiração para desenvolver um modelo baseado em CA [3].

Esse modelo de corrosão é aplicado em cada imagem da base de dados, resultando em imagens de textura "corroídas". A partir da massa de corrosão acumulada total, obtemos uma matriz de descritores, usada futuramente para a classificação das imagens.

3.1 Corrosão localizada de metais

Corrosão é a desintegração de metais ou ligas metálicas devido à reação de seus elementos com o meio ambiente. Neste trabalho utilizaremos o termo *pitting corrosion*, que se trata de um tipo de corrosão localizada, onde toda ou boa parte da perda de metal ocorre de forma concentrada em certas áreas, como pequenos buracos na superfície do metal.

3.2 Autômatos celulares (CAs)

Autômatos celulares são construções matemáticas usadas para modelar a evolução temporal de sistemas, cujo espaço, tempo e estado possuem domínio discreto. O arranjo das células parte de uma configuração inicial e cada componente do sistema tem sua evolução temporal baseada no estado atual de sua vizinhança e em como interagem.

Definição 1 *Autômatos celulares bidimensionais podem ser representados como $C = \langle T; S; s; N; \phi \rangle$, onde:*

- *T é a grade bidimensional das células.*
- *S é o conjunto finito dos estados das células.*
- *A função de saída s corresponde ao estado de cada célula a cada passo de tempo.*

- A função de vizinhança N define as células vizinhas de cada célula, incluindo ela mesma.
- A função de transição ϕ dá o estado $s(c, t + 1)$ de cada célula no tempo seguinte, baseado no seu estado e vizinhança atual.

3.2.1 Estado das células

A imagem original é considerada a configuração inicial de um CA. Ou seja, a imagem bidimensional é tratada como uma grade T de dimensões definidas pelo tamanho da imagem com cada pixel sendo uma célula, no qual o valor do pixel corresponde ao estado inicial da mesma.

3.2.2 Tipos de vizinhança

Há diferentes tipos de vizinhanças que podem ser definidas para uma grade 2D, sendo as mais importantes a vizinhança de Moore e a de von Neumann, como mostrado na Figura 5. Para o desenvolvimento deste projeto utilizaremos a vizinhança de Moore.

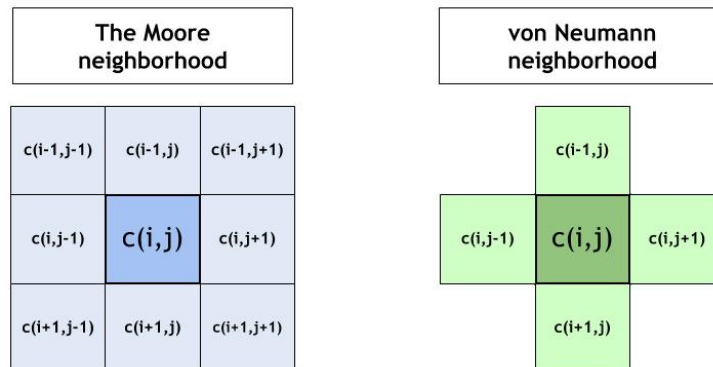


Figura 5: Esquema dos dois importantes tipos de vizinhança para CAs.

Assim, utilizando uma função de transição, é possível calcular o estado seguinte de uma célula baseado em seu estado atual e nos estados de sua vizinhança.

3.3 Descritores CITA (*Corrosion-inspired texture analysis*)

Tratamos a imagem original I como uma superfície de metal e o estado s de cada célula como a profundidade da corrosão (0 para nenhuma e 255 como máxima em $t = 0$). Além disso, para que ocorra processo corrosivo, uma diferença mínima $d_{i,j}$ de altura entre a célula $c_{i,j}$ e suas vizinhas é necessária. Essa diferença $d_{i,j}$ é dada por:

$$d_{i,j} = s(c_{i,j}, t) - \min(s * (N(c_{i,j}, t))), \quad (1)$$

em que $\min(s * (N(c_{i,j}, t)))$ é o conjunto dos estados da vizinhança.

Para iniciar o processo corrosivo, é necessário um **parâmetro de rugosidade da superfície** ν . Esse parâmetro dita se o local sofrerá corrosão ou não, seguindo as condições:

- Se $d \leq \nu$, a corrosão é insignificante.
- Se $d \geq 255$, a corrosão tende a zero.

E o cálculo do nível de corrosão aplicada ao local é dado por:

$$Q(d_{i,j}, \gamma) = (255 - d_{i,j}) * \gamma, \quad (2)$$

em que $\gamma \in [0, 1]$ é o *pitting power* (resistência do material à corrosão) e Q o menor valor inteiro. Esse valor é utilizado no cálculo do total de massa que sofreu o processo de corrosão. Assim, a função de transição ϕ estabelece o estado de $c_{i,j}$ no tempo seguinte de acordo com:

$$s(c_{i,j}, t + 1) = \begin{cases} s(c_{i,j}, t) + Q(d_{i,j}, \gamma), & \text{se } 255 \geq d \geq \nu \\ s(c_{i,j}, t), & \text{se } d < \nu \text{ ou } d > 255 \end{cases} \quad (3)$$

A cada iteração, após atualizar o estado da célula, a massa que sofreu corrosão naquela iteração é adicionada ao total corroído na iteração anterior. Ao final, o total de massa corroída relativo ao número total de pixels é usado como descritor de textura da imagem. Portanto, o número total de características que compõem o vetor de descritores da imagem será o total de iterações feitas pelo modelo.

4 Metodologia

Para este trabalho foi utilizada a base de imagens de textura *Brodatz*, composta por 1776 imagens divididas em 111 classes (tipos de texturas naturais únicas). As imagens possuem tamanho 128 x 128 pixels em escala cinza.

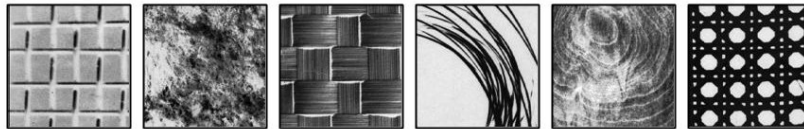


Figura 6: Exemplos de imagens presentes na base Brodatz.

Aplicaram-se diferentes níveis de corrosão na base de imagens *Brodatz* a fim de aprofundar a análise. Optou-se por usar três diferentes níveis de corrosão, classificando as imagens como: **Pouco corroídas** ($\gamma = 0.01$, $\nu = 1$ e 30 iterações); **Corroídas** ($\gamma = 0.05$, $\nu = 5$ e 90 iterações); **Muito corroídas** ($\gamma = 0.05$, $\nu = 1$ e 158 iterações).

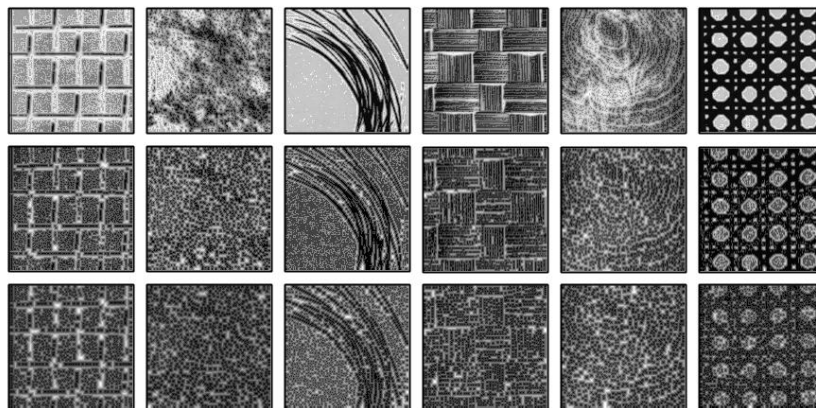


Figura 7: Exemplos dos níveis de corrosão. As fileiras representam, respectivamente, as imagens pouco corroídas, corroídas e muito corroídas.

A implementação foi feita com o notebook [Google Colab](#) utilizando a linguagem Python e as bibliotecas *Scikit-learn* e *Pytorch*, além de outras bibliotecas auxiliares. Todos os códigos podem ser [encontrados aqui](#).

4.1 Extrair os descritores CITA

Para este método, os descritores foram extraídos pelo [modelo CA](#), conforme a Subseção 3.3. Esse processo foi realizado para cada nível de corrosão, tendo no final três matrizes de tamanhos: **pouco corroídas** (1776, 30); **corroídas** (1776, 90); **muito corroídas** (1776, 158).

4.2 Extrair os descritores com a rede neural

A rede neural convolucional utilizada foi baseada no tutorial de "*Transfer Learning*" da biblioteca *Pytorch* [2]. A rede foi treinada com os conjuntos de imagens corroídas, a fim de determinar os parâmetros com melhores resultados. Após o ajuste, usou-se a rede sobre as imagens para extrair os vetores de descritores. Esses vetores são os *feature maps* da rede, os quais contém os valores calculados em cada neurônio na penúltima camada da rede. Esse processo foi realizado para a imagem original e para a imagem corroída, concatenando os dois vetores no final visando melhorar a acurácia da classificação. Assim, obtivemos três matrizes de descritores de tamanho (1776, 4096), cada uma referente a um nível de corrosão.

Os dados obtidos em 4.1 e 4.2 foram divididos em treinamento e teste pelo esquema de validação cruzada estratificado (*Stratified K-Fold*). O método consiste em dividir os dados em K partes iguais, treinando o modelo com $K - 1$ partes, enquanto a parcela restante fica destinada ao teste. Esse processo mantém as proporções das classes em cada parte e é repetido K vezes (em cada momento uma partição diferente será o teste) [8]. Por fim, calcula-se a média e o desvio padrão dos resultados. Neste projeto optou-se por utilizar $K = 10$.

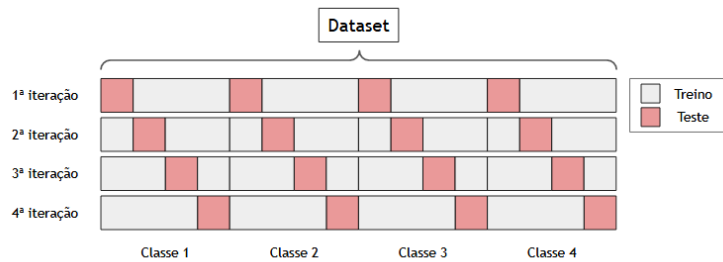


Figura 8: Esquema demonstrativo do método *Stratified K-Fold*.

5 Resultados

A fim de obter o melhor resultado, alguns classificadores foram utilizados, como: Análise Discriminante Linear (LDA); K-ésimo Vizinho mais Próximo (KNN); Árvore de Decisão (CART); Florestas Aleatórias (RF); Gaussian Naive Bayes (NB); Máquinas de Vetor de Suporte (SVC) [8]. Como pode ser visto abaixo, em todos os casos os melhores desempenhos ficaram por conta do LDA.

Para imagens pouco corroídas, os descritores CITA têm um desempenho de 68.3% contra 98.8% dos descritores da rede neural.

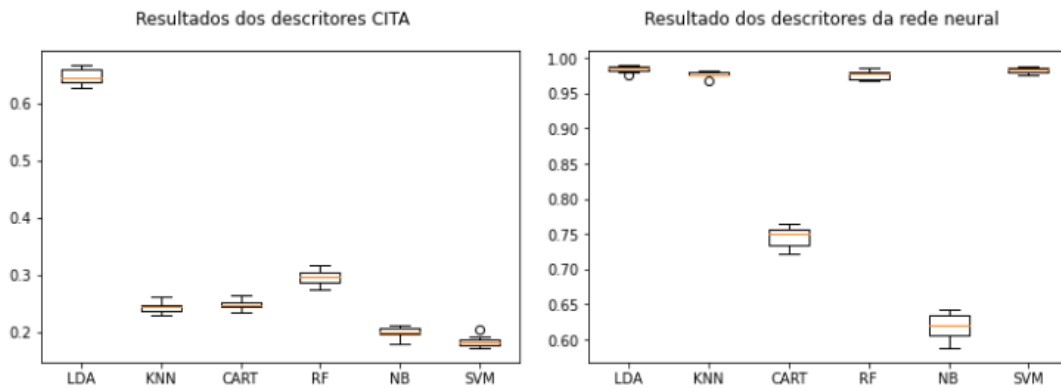


Figura 9: Boxplot dos descritores das imagens pouco corroídas.

| Classificadores | Taxa de acerto (\pm Desvio padrão) | |
|-----------------|---------------------------------------|-------------------|
| | Descritores CITA | Descritores CNN |
| LDA | 68.3 (\pm 1.6) | 98.8 (\pm 0.7) |
| KNN | 29.5 (\pm 2.7) | 97.9 (\pm 1.1) |
| CART | 29.1 (\pm 3.1) | 79.5 (\pm 1.9) |
| RF | 34.7 (\pm 3.7) | 98.1 (\pm 1.1) |
| NB | 21.1 (\pm 3.2) | 82.1 (\pm 3.2) |
| SVM | 8.6 (\pm 1.1) | 98.6 (\pm 0.8) |

Figura 10: Resultados dos descritores das imagens pouco corroídas.

Já para imagens mais corroídas, os descritores CITA têm um desempenho de 91.2% contra 97.2% dos descritores da rede neural.

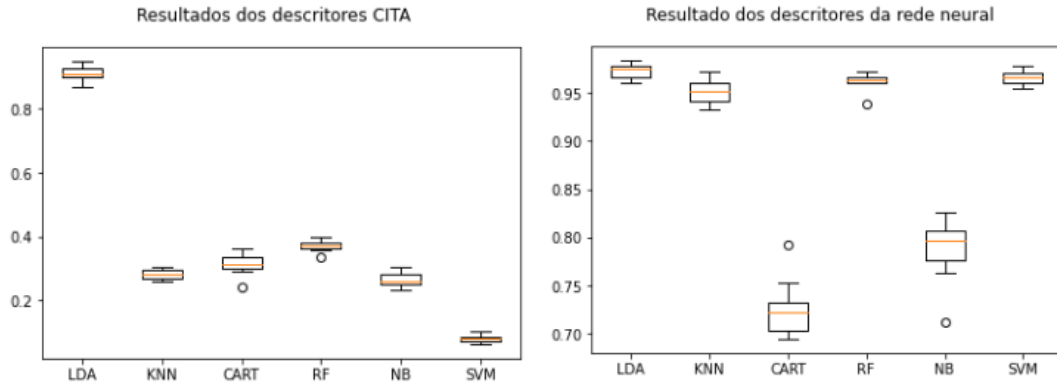


Figura 11: Boxplot dos descritores das imagens corroídas.

| Classificadores | Taxa de acerto (\pm Desvio padrão) | |
|-----------------|---------------------------------------|-------------------|
| | Descritores CITA | Descritores CNN |
| LDA | 91.2 (\pm 2.3) | 97.2 (\pm 0.8) |
| KNN | 28.1 (\pm 1.5) | 95.2 (\pm 1.2) |
| CART | 31.3 (\pm 3.2) | 72.6 (\pm 2.8) |
| RF | 36.9 (\pm 1.7) | 96.2 (\pm 0.9) |
| NB | 26.5 (\pm 2.3) | 78.8 (\pm 3.1) |
| SVM | 7.9 (\pm 1.1) | 96.5 (\pm 0.7) |

Figura 12: Resultados dos descritores das imagens corroídas.

Apesar do desempenho da rede neural ter uma leve queda entre o nível de corrosão anterior e esse, é possível observar que a acurácia da classificação pelos descritores CITA aumenta significativamente, com uma melhora de quase 23% na taxa de acerto.

Por fim, para imagens muito corroídas, os descritores CITA chegam em 92.3%, enquanto os da rede neural ficam em torno de 96.9%

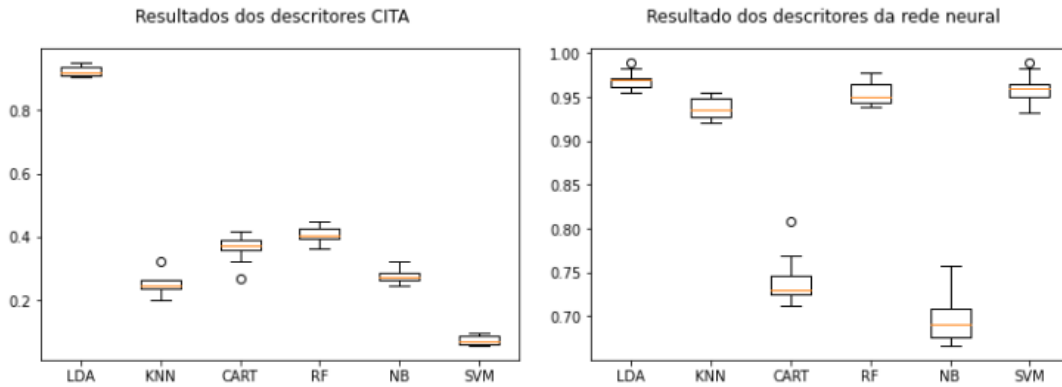


Figura 13: Boxplot dos descritores das imagens muito corroídas.

| Classificadores | Taxa de acerto (\pm Desvio padrão) | |
|-----------------|---------------------------------------|-------------------|
| | Descritores CITA | Descritores CNN |
| LDA | 92.3 (\pm 1.5) | 96.9 (\pm 1.0) |
| KNN | 24.9 (\pm 3.0) | 93.8 (\pm 1.2) |
| CART | 36.5 (\pm 4.0) | 73.9 (\pm 2.8) |
| RF | 40.9 (\pm 2.5) | 95.4 (\pm 1.2) |
| NB | 27.5 (\pm 2.1) | 69.8 (\pm 2.7) |
| SVM | 7.4 (\pm 1.4) | 96.0 (\pm 1.6) |

Figura 14: Resultados dos descritores das imagens muito corroídas.

É possível observar o ótimo desempenho dos descritores extraídos pela rede neural para todos os níveis de corrosão aqui abordados. Entretanto, para os descritores CITA, os resultados se mostram melhores apenas para os níveis de maior corrosão.

Infelizmente, os resultados dos descritores CITA durante o projeto se encontram abaixo daqueles expostos no artigo de referência [3]. Nesse artigo, são considerados os valores ótimos seguintes para os parâmetros: $\gamma = 0.05$, $\nu = 1$ e 158 iterações, os mesmos parâmetros usados aqui na categoria "imagens muito corroídas". A taxa de acerto para a classificação da base de imagens *Brodatz* obtida no artigo para esses parâmetros foi de 99%, desempenho melhor do que ao acerto de 92.3% dos descritores CITA desenvolvidos no projeto.

6 Conclusão

O projeto abordou alguns aspectos teóricos sobre aprendizado de máquinas e classificação de imagens, sendo possível estudar conceitos envolvendo redes neurais convolucionais e uma nova proposta de extração de descritores em imagens de textura a partir de autômatos celulares. Além dos aspectos teóricos, conseguimos observar o comportamento de dois diferentes métodos de extração de descritores, analisar seus desempenhos e como seus resultados são influenciados a partir do nível de corrosão aplicado. Assim, apesar das divergências entre os valores obtidos pelo projeto e pelo artigo "*Improved texture image classification through the use of a corrosion-inspired cellular automaton.*" [3], os resultados foram satisfatórios, mostrando o potencial dos descritores CITA.

Referências

- [1] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2016.
- [2] Sasank Chilamkurthy. Transfer learning for computer vision tutorial, 2017. https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html.
- [3] Núbia Rosa da Silva, Pieter Van der Weeën, Bernard De Baets, and Odemir Martinez Bruno. Improved texture image classification through the use of a corrosion-inspired cellular automaton. *Neurocomputing*, 149:1560–1572, Feb 2015. <http://dx.doi.org/10.1016/j.neucom.2014.08.036>.
- [4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [5] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, Dec 2017.
- [6] MissingLink.ai. Convolutional neural networks for image classification. <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-networks-image-classification/>.
- [7] Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. <http://neuralnetworksanddeeplearning.com/index.html>.
- [8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.
- [9] Wikipedia contributors. Neural network — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Neural_network&oldid=970977775, 2020.