

Classificação de Singularidades em Imagens de Impressão Digital Baseada em Redes Neurais Convolucionais

Paulo Ricardo P. da Silva¹, Leonardo Vidal Batista¹, Arnaldo Gualberto Silva², João Janduy Brasileiro²,
Diogo Ventura Dantas¹

¹Centro de Informática - Universidade Federal da Paraíba

²Centro de Engenharia Elétrica e Informática - Universidade Federal de Campina Grande
pauloricardo@ppgi.ci.ufpb.br, leonardo@ci.ufpb.br, {arnaldo.g12, jjanduy, dioogoven}@gmail.com

Resumo: A Biometria oferece um mecanismo de autenticação confiável utilizando traços (físicos ou comportamentais) que permitam identificar usuários baseados em suas características naturais. O processo de comparação de impressão digital utiliza como atributo discriminante informações locais como minúcias. Porém, devido a problemas como ruído na captura ou desgaste nas impressões digitais, este atributo nem sempre é suficiente para a realização desta tarefa. Portanto, singularidades do tipo laço e delta podem contribuir nesta etapa para reduzir a taxa de erro. Este trabalho propõe um método para classificar singularidades em imagens de impressão digital, que se baseia em redes neurais convolucionais. Para avaliar a efetividade do algoritmo proposto foi utilizada a base de dados FVC2006-500 sobre a qual o modelo alcançou acurácia de 98%.

Palavras-chave: *Impressão digital; Singularidades; Redes Neurais Convolucionais.*

1. Introdução

Desde 1893, o Departamento de Segurança do Reino Unido assume que dois indivíduos não possuem a mesma impressão digital. Pouco depois da descoberta de Alphonse Bertillon, que afirmava que uma pessoa poderia ser identificada a partir de um conjunto de atributos antropomórficos, muitos dos principais departamentos de aplicação da lei perceberam o potencial das impressões digitais na identificação de criminosos reincidentes que usavam outros nomes para, ao ser recapturado, escaparem de penalidades mais severas, aplicadas a reincidentes [1]. As agências de segurança investiram em um estudo rigoroso das impressões digitais, desenvolvendo métodos científicos para a comparação visual das digitais e instituindo programas para treinamentos de especialistas na área [2].

A monotonicidade e as cargas de trabalho cada vez maiores decorrentes do aumento da demanda de serviços de reconhecimento das impressões digitais levaram ao desenvolvimento dos primeiros Sistemas de Identificação Automática de Impressões Digitais (*Automatic Fingerprint Identification Systems - AFIS*), aproximadamente 40 anos após o início do uso de impressões digitais para identificação biométrica [2].

Em geral, os AFIS usam minúcias do tipo terminação e bifurcação como atributo discriminante [3]. Atributos globais como laço e delta podem ser usados para reduzir a taxa de erro e classificar impressões digitais em *arch* ou arco plano, *tented arch* ou arco angular, *left loop* ou presilha externa, *right loop* ou presilha interna, e *whorl* ou verticilo. Essas classes são usadas na indexação das amostras de impressão digital, o que ajuda a acelerar o tempo de resposta desses sistemas.

Redes neurais convolucionais (*Convolutional Neural Network - CNN*) são um tipo específico de redes neurais que processam dados representados na forma de matriz, como séries temporais (1-D), imagens (2-D) e

vídeo (3-D). Como o nome sugere, ao contrário das redes neurais que operam por meio de multiplicação de matrizes, redes convolucionais aplicam um tipo específico de operação matemática linear chamada de convolução [4]. Redes convolucionais vem ganhando destaque pelo seu desempenho na solução de problemas de classificação e detecção relacionados à imagens, no campo da biometria tem surgido trabalhos baseado nessa tecnologia para classificar impressão digital [5], detectar singularidades [6], e detectar minúcias [7].

Este trabalho propõe um método para classificar singularidades entre as classes laço, delta e não singularidade em imagens de impressão digital baseado em CNN. Porém ele não é completo porque não detecta os pontos singulares, mas pode ser usado para detectar se combinado com um algoritmo de janela deslizante que passa por toda imagem se deslocando em bloco de tamanho 50x50 a um passo p .

2. Fundamentação Teórica

Em geral, os trabalhos descritos na literatura científica baseiam-se na imagem de orientação para calcular a localização e o tipo dos pontos singulares. Os algoritmos de detecção e classificação de singularidades podem ser categorizados como: baseado no *Poincaré index (Plindex)*, baseado no particionamento da imagem de orientação, baseado em *template*, baseado na curvatura da orientação de regiões de pontos singulares, e, mais recentemente, baseado em CNN.

Um método sofisticado e prático, que se baseia no *Plindex*, um campo vetorial e uma curva que envolvem esse campo, foi proposto por Kawagoe e Tojo [8]. Por meio da imagem de orientação, em uma janela deslizante, é calculado o somatório das diferenças dos ângulos de cada pixel em torno de uma vizinhança para determinar o tipo e a localização dos pontos singulares de uma imagem de impressão digital. Há outros trabalhos que se baseiam no *Plindex* ou usam-no como parte do processo de detecção e classificação de singularidades, como em [9, 10].

Alguns algoritmos agrupam as orientações da imagem de orientação de acordo com suas similaridades, formando conjuntos separados por linhas que determinam suas fronteiras. As interseções entre essas linhas fornecem a localização das singularidades [11, 12, 13]. Puneet e Phalguni [13] propuseram um método que calcula um conjunto de singularidades candidatas por meio do algoritmo de particionamento da imagem de orientação, esse conjunto é validado pela análise do *Plindex*. Para refinar a localização das singularidades e detectar possíveis deltas que não foram descobertos, é realizado um pós-processamento, e assim, apenas singularidades genuínas são detectadas e classificadas.

Nos métodos baseados em *template*, para cada tipo de singularidade, há um filtro (ou *template*) que é convolucionado sobre a imagem de impressão digital para extrair singularidades [14, 15]. Considerado o estado da arte [16], Awad e Baba [17] apresentaram um método baseado em 2 filtros complexos que captura as propriedades de simetria de laço e delta, respectivamente, então convoluciona cada filtro com a imagem de orientação e o ponto que obtiver a resposta mais alta do filtro é considerado ponto singular. Os algoritmos fundamentados na curvatura da orientação de regiões de pontos singulares são bons para detectar e classificar pontos singulares, visto que as áreas onde encontram-se tais pontos são marcadas pela grande mudança de orientação [18, 19]. Neste sentido, o trabalho de Qi e Liu [19], considerado estado da arte [16], apresenta um método sensível a ruído e falsas singularidades são extraídas.

O único trabalho encontrado e acessível, baseado em CNN, sobre detecção e classificação de singularidades foi o de Qin [6], que apresentou um método que combina CNN e um modelo de probabilidade. Primeiramente, eles treinam um classificador com sub-imagens ou blocos rotulados em 3 classes: laço, delta e não singularidade. Então, eles usam uma CNN para estimar se o centro de um bloco é uma singularidade ou não.

Há outras formas de extrair singularidades (p. ex. aprendizagem de máquina e modelos matemáticos). Qi e Liu [19] propuseram um método baseado em um modelo polinomial complexo (*Zero-pole model*) e uma janela deslizante para detectar e classificar singularidades. *Zero-pole model* da imagem de orientação é essencialmente um polinômio racional complexo cujos *zero* e *pole* são considerados como pontos singulares laço e delta, respectivamente.

O método proposto neste trabalho, diferente dos métodos citados como estado da arte, não depende da imagem de orientação, que tem elevado custo computacional e é sensível à qualidade da imagem. No entanto, ele apenas classifica as singularidades, não detecta. Como os métodos encontrados na literatura classificam e detectam singularidades ao mesmo tempo, não há nenhuma comparação com outros trabalhos.

3. Metodologia

Este trabalho propõe um método de classificação de singularidades em imagens de impressão digital baseado em rede neural convolucional, que é dividido em 2 etapas:

1. Pré-processamento
 - (a) Equalização de histograma;
 - (b) Realce baseado em filtros de *Gabor* [20];
 - (c) Binarização;
 - (d) Extração manual de amostras (imagens 50x50 pixels);
2. Classificação por meio da Rede Neural Convolucional.

Para remover ruídos e melhorar a qualidade das imagens, são aplicadas a equalização de histograma, o realce baseado em filtros de *Gabor* e a binarização - veja a Figura 1. Como o objetivo deste trabalho é apenas classificar, para localizar as singularidades e não singularidades (ou Neg), foram extraídas manualmente subimagens (50x50 pixels) para compor a base de dados usada pelo método proposto.

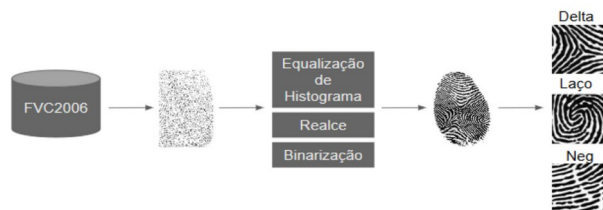


Figura 1. Visão geral do pré-processamento.

Após o pré-processamento, com as imagens separadas em 3 classes (laço, delta e neg), elas são carregadas para um *NumPy Array* [22] de dimensão 2866x50x50x1. Os rótulos das classes são representados seguindo o formato *one hot encoding* e armazenados em um *NumPy Array* de dimensão 2866x3. Todos os valores dos pixels das imagens são convertidos de inteiro para real (*float32*) e normalizados entre [0,1] pela divisão por 255. Para tratar possíveis problemas de superajuste, os dados são divididos em 3 conjuntos: treino, teste e validação. Esse particionamento é feito por meio do método *train test split* da biblioteca *scikit-learn* [23], que faz a divisão de forma aleatória e estratificada, ou seja, mantendo as proporções de cada classe em todas as partições.

Então, a CNN é instanciada e as camadas são adicionadas e configuradas com o inicializador de pesos ou *kernel initializer glorot uniform* e a função de ativação *relu* nas primeiras camadas e na última a função *softmax*. Assim, a CNN é compilada e configurada com a função de perda ou *loss categorical crossentropy*, o otimizador ou *optimizer adam* e com a métrica ou *metrics accuracy* - vide Figura 2 e Tabela 1.

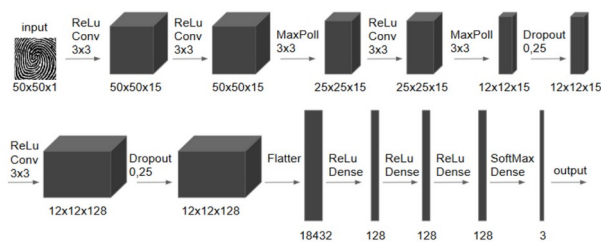


Figura 2. Arquitetura do modelo baseada na LeNet [21].

Tabela 1. Alguns dos hiper-parâmetros usados no modelo.

Nome	Valor
<i>batch_size</i>	128
<i>epochs</i>	50
<i>kernel_initializer</i>	<i>glorot_uniform</i>
<i>loss</i>	<i>categorical_crossentropy</i>
<i>optimizer</i>	<i>Adam</i>
<i>metrics</i>	<i>accuracy</i>

No treinamento, a CNN é configurada com os valores dos hiper-parâmetros *batch size* e *epochs*, 128 e 50, respectivamente, e é utilizado o método *imageDataGenerator* do *Keras* [24] para aumentar a quantidade de imagens. Com o objetivo de evitar perda de informação, este método foi configurado apenas para rotacionar (até 45°), girar (*flip*) horizontal e verticalmente.

Para avaliar a capacidade de generalização do modelo, em um dos experimentos, foi aplicada a técnica de validação cruzada *k-fold* com $k = 10$. Foi utilizado o método *StratifiedKfold* da biblioteca *scikit-learn* que, a cada iteração, aleatoriamente divide os dados em 10 conjuntos, deixando 1 para teste e os outros 9 para treino. Esses 9 conjuntos foram divididos em treino e validação, e então foi realizada a codificação ou *encoding* das classes alvo e usado o aumentador de imagens do *Keras*, como descrito acima.

4. Apresentação e Análise dos Resultados

O primeiro experimento foi usado para encontrar a arquitetura da rede, a configuração dos hiper-parâmetros e o pré-processamento mais adequados a solução do problema. Esta etapa foi realizada de forma empírica, testando diferentes valores de hiper-parâmetros da Tabela 1, camadas da rede e tamanho das imagens extraídas manualmente. Então, após testar algumas configurações diferentes, chegou-se à arquitetura apresentada na Figura 2. O modelo alcançou o índice de acurácia exatidão global de aproximadamente 95% no teste (Tabela 2).

Tabela 2. Matriz de confusão.

	Laço	Delta	Neg	
Classe verdadeira	Laço	93	0	1
Delta	0	38	3	
Neg	1	2	6	
	Classe Predita			

Nesse experimento, o modelo apresentou erro na classificação de 7 imagens, e em todas elas a classe neg estava envolvida. Um fator importante foi que não houve erros entre as classes laço e delta, o que mostra que o modelo consegue distinguir as duas classes principais. A Figura 3 mostra as imagens erradas pelo modelo, e em todas elas não é simples identificar a classe. É possível que esses erros tenham sido causados devido a pouca quantidade de imagens das classes delta e neg, visto que a classe laço, que é a que tem mais amostras, esteve envolvida em apenas 2 erros (ver as Figuras 3b e 3c).

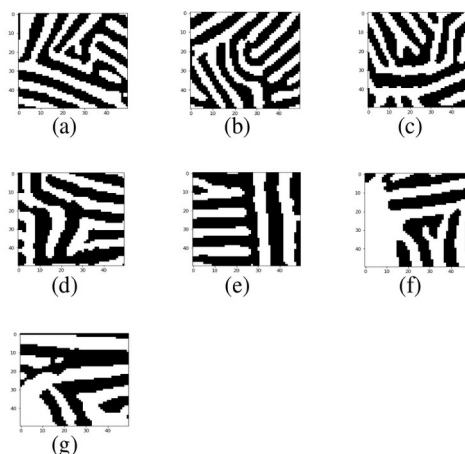


Figura 3. Imagens classificadas erradas pelo modelo no teste do primeiro experimento (gabarito | resultado do modelo): (a) Delta | Neg; (b) Laço | Neg; (c) Neg | Laço; (d) Neg | Delta; (e) Neg | Delta; (f) Delta | Neg; (g) Delta | Neg.

O segundo experimento foi desenvolvido para testar a capacidade de generalização do modelo. Apesar de validação cruzada não ser uma prática comum em modelos baseados em CNN por causa do alto custo computacional, ela foi adotada devido à falta de outros bancos pré-processados e do uso de processamento em GPU. Este experimento durou aproximadamente 4h para concluir as 10 rodadas de validação cruzada, e os resultados obtidos nos testes podem ser vistos na Tabela 3. As matrizes de confusão e as imagens erradas nos testes foram omitidas devido ao espaço que ocupariam neste trabalho.

Tabela 3. Resultados obtidos durante as 10 rodadas de teste de validação cruzada.

	Média	Mediana	Desvio padrão
Acurácia	0.9850	0.9850	0.0024
Perda	0.0469	0.0469	0.0083

O modelo gerou resultados promissores, visto que apresentou o índice de acurácia exatidão global médio de aproximadamente 98%. Esses resultados são preliminares e não garantem que o modelo já está pronto para ser usado como parte de um método de casamento de impressões digitais com desempenho semelhante, pois é necessário testá-lo em outras bases de dados.

5. Conclusão

Apesar de os resultados serem promissores, o modelo precisa ser refinado: é possível reduzir o número de camadas e tornar mais equilibrado o número de amostras das classes. Além disso, é necessário testar o modelo sobre outras bases de dados e aplicar o pré-processamento sobre elas. As pretensões futuras são de desenvolver um modelo baseado em redes neurais convolucionais para fazer tanto a detecção quanto a classificação de singularidades. Mas, para isso ser alcançado, é necessário obter mais bases de imagens para o treinamento do modelo.

Bibliografia

- [1] Ross, A. A.; Nandakumar, K.; Jain, A. (2006) Handbook of multibiometrics. Springer Science & Business Media 6.
- [2] Maltoni, D. et al. (2009) Handbook of fingerprint recognition. Second edition. Springer Science & Business Media.
- [3] Dorizzi, B. et al. (2009) Fingerprint and on-line signature verification competitions at ICB 2009. *Advances in Biometrics – Lecture Notes in Computer Science* 5558: 725–732. DOI: [10.1007/978-3-642-01793-3_74](https://doi.org/10.1007/978-3-642-01793-3_74)
- [4] Goodfellow, I.; Bengio, Y.; Courville, A. (2016) Deep Learning. MIT Press. Online: <http://www.deeplearningbook.org>.
- [5] Wu, F.; Zhu, J.; Guo, X. (2020) Fingerprint pattern identification and classification approach based on convolutional neural networks. *Neural Computing and Applications* 32:5725-5734. DOI: [10.1007/s00521-019-04499-w](https://doi.org/10.1007/s00521-019-04499-w)
- [6] Qin, J. et al (2017) Multi-scaling detection of singular points based on fully convolutional networks in fingerprint images. *Biometric Recognition – Lecture Notes in Computer Science* 10568: 221–230. DOI: [10.1007/978-3-319-69923-3_24](https://doi.org/10.1007/978-3-319-69923-3_24)
- [7] Nguyen, D.-L.; Cao, K.; Jain, A. K. (2018) Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge. In: IEEE. 2018 International Conference on Biometrics (ICB), p. 9–16. DOI: [10.1109/ICB2018.2018.00013](https://doi.org/10.1109/ICB2018.2018.00013)
- [8] Kawagoe, M.; Tojo, A. (1984) Fingerprint pattern classification. *Pattern Recognition* 17(3): 295-303. Elsevier. DOI: [10.1016/0031-3203\(84\)90079-7](https://doi.org/10.1016/0031-3203(84)90079-7)
- [9] Jin, C.; Kim, H. (2010) Pixel-level singular point detection from multi-scale gaussian filtered orientation field. *Pattern Recognition* 43(11): 3879–3890. Elsevier. DOI: [10.1016/j.patcog.2010.05.023](https://doi.org/10.1016/j.patcog.2010.05.023)
- [10] Zhou, J.; Gu, J.; Zhang, D. (2007) Singular points analysis in fingerprints based on topological structure and orientation field. *Advances in Biometrics – Lecture Notes in Computer Science* 4642: 261–270. DOI: [10.1007/978-3-540-74549-5_28](https://doi.org/10.1007/978-3-540-74549-5_28)
- [11] Ramo, P. et al. (2001) Optimized singular point detection algorithm for fingerprint images. In: Proc. Int. Conf. Image Processing 3: 242–245. DOI: [10.1109/ICIP.2001.958096](https://doi.org/10.1109/ICIP.2001.958096)
- [12] Huang, C.-Y.; Liu, L.-m.; Hung, D. D. (2007) Fingerprint analysis and singular point detection. *Pattern Recognition Letters* 28(15): 1937–1945. Elsevier. DOI: [10.1016/j.patrec.2007.04.003](https://doi.org/10.1016/j.patrec.2007.04.003).
- [13] Gupta, P.; Gupta, P. (2015) A robust singular point detection algorithm. *Applied Soft Computing* 29: 411–423, Elsevier. DOI: [10.1016/j.asoc.2015.01.027](https://doi.org/10.1016/j.asoc.2015.01.027).
- [14] Nilsson, K.; Bigun, J. (2003) Localization of corresponding points in fingerprints by complex filtering. *Pattern Recognition Letters* 24(13): 2135–2144. Elsevier. DOI: [10.1016/S0167-8655\(03\)00083-7](https://doi.org/10.1016/S0167-8655(03)00083-7).
- [15] Jain, A. K. et al. (2000) Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing* 9(5): 846–859. DOI: [10.1109/83.841531](https://doi.org/10.1109/83.841531).
- [16] Zhu, E.; Guo, X.; Yin, J. (2016) Walking to singular points of fingerprints. *Pattern Recognition* 56: 116–128. Elsevier. DOI: [10.1016/j.patcog.2016.02.015](https://doi.org/10.1016/j.patcog.2016.02.015).
- [17] Awad, A. I.; Baba, K. (2012) Singular point detection for efficient fingerprint classification. *Int. Journal on New Computer Architectures and Their Applications* 2(1): 1–7.
- [18] Chen, H. et al. (2011) Fingerprint singular point detection based on multiple-scale orientation entropy. *IEEE Signal Processing Letters* 18(11): 679–682. IEEE. DOI: [10.1109/LSP.2011.2169957](https://doi.org/10.1109/LSP.2011.2169957).
- [19] Qi, J.; Liu, S. (2014) A robust approach for singular point extraction based on complex polynomial model. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, p. 78–83. DOI: [10.1109/CVPRW.2014.17](https://doi.org/10.1109/CVPRW.2014.17).
- [20] Turrone, F.; Cappelli, R.; Maltoni, D. (2012) Fingerprint enhancement using contextual iterative filtering. In: Proc. 5th IAPR International Conference on Biometrics, p. 152–157. DOI: [10.1109/ICB.2012.6199773](https://doi.org/10.1109/ICB.2012.6199773).
- [21] Lecun, Y. et al. (1998) Gradient-based learning applied to document recognition. Proc. IEEE 86(11): 2278–2324. DOI: [10.1109/ICB.2012.6199773](https://doi.org/10.1109/ICB.2012.6199773).
- [22] Walt, S. v. d.; Colbert, S. C.; Varoquaux, G. (2011) The Numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering* 13(2): 22–30. DOI: [10.1109/MCSE.2011.37](https://doi.org/10.1109/MCSE.2011.37).
- [23] Pedregosa, F. et al. (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.
- [24] Chollet, F. et al. (2015) Keras. <https://keras.io>.