

Classificação de Regiões Singulares em Imagens de Impressão Digital por Meio de Redes Neurais Convolucionais

Paulo Ricardo P. da Silva¹, Leonardo Vidal Batista¹, João Janduy Brasileiro²

¹Centro de Informática – Universidade Federal da Paraíba (UFPB)

²Centro de Engenharia Elétrica e Informática - Universidade Federal de Campina Grande (UFCG)

{paulosilvajp0, jjanduy}@gmail.com, leonardo@ci.ufpb.br

Abstract. *Fingerprint recognition is one of the most used biometrics approach, since its high accuracy and low cost make the system more affordable. However, fingerprint recognition is still an open problem, since false acceptance and false rejection errors can be found in matching algorithm. The fingerprint matching process uses the contrastive local information minutiae as the main attribute, but, because of problems like noise in sensing stage or deterioration on fingerprint, sometimes this attribute is not enough for that task. Therefore, singularities of type core and delta can collaborate in this stage for decreasing the error. This work proposes a method to classify singularities from fingerprint images, that is based on Convolutional Neural Network. To evaluate the performance of the proposed algorithm, the database FVC2002-1 was used on which it reached the value of 100% of Accuracy.*

Resumo. *O reconhecimento por impressão digital é uma das abordagens biométricas mais utilizadas, visto que sua alta precisão e seu baixo custo tornam os sistemas mais acessíveis. No entanto, o reconhecimento por impressões digitais ainda é um problema em aberto, uma vez que erros de falsa aceitação e falsa rejeição ainda são encontrados nos algoritmos de comparação de impressões digitais. 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 baseia-se em redes neurais convolucionais. Para avaliar a efetividade do algoritmo proposto, foi utilizada a base de dados FVC2002-1 sobre a qual o modelo alcançou acurácia de 100%.*

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

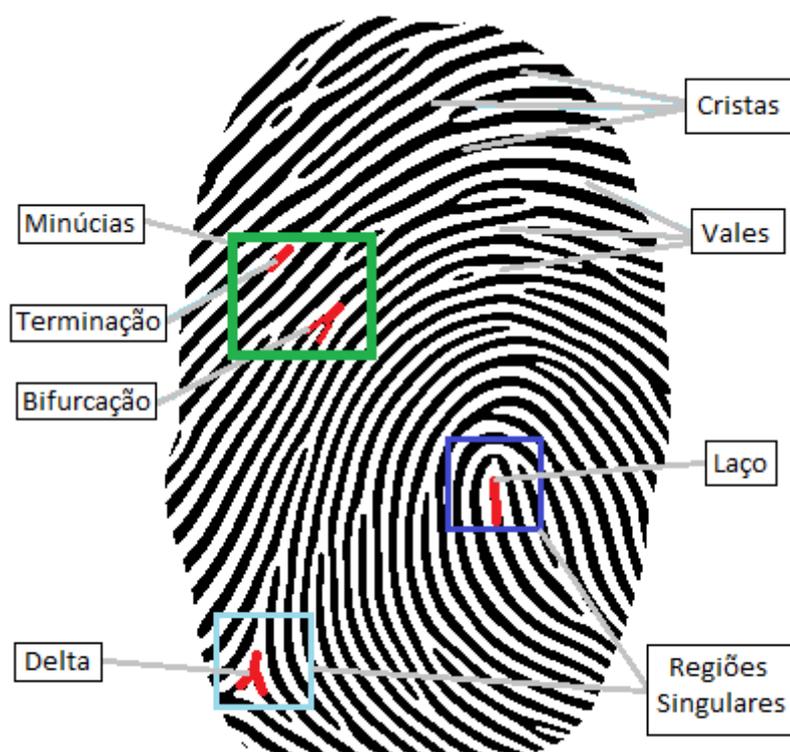


Figura 1. Exemplos de atributos em uma imagem de impressão digital.

usavam outros nomes para, ao ser recapturado, escaparem de penalidades mais severas, aplicadas a reincidentes [Ross et al. 2006].

Uma impressão digital é uma reprodução da aparência exterior da epiderme do dedo. A estrutura mais característica da impressão digital é o padrão intercalado de cristas e vales ou sulcos. Em uma imagem de impressão digital, as cristas são as áreas escuras e os vales são as áreas mais claras (Figura 1). Os detalhes das cristas são descritos em uma ordem hierárquica de diferentes níveis: Nível 1 (padrão global de disposição das cristas ou singularidades), Nível 2 (minúcias) - há o Nível 3 (poros e formatos das cristas), mas como essas informações não usadas neste trabalho, ele não será detalhado. No Nível 1 (nível global), encontram-se as singularidades, que podem ser definidas como estruturas compostas de cristas, nas quais estas seguem um fluxo padrão em determinadas regiões ao longo da digital - os principais tipos de singularidade são laço e delta. No Nível 2 (nível local) existem as minúcias, que descrevem as várias formas de descontinuidade das cristas. Se uma crista é descontinuada repentinamente, surge uma terminação, caso ela se divida em duas cristas, surge uma bifurcação - existem outros tipos minúcias, porém, esses são os principais [Maltoni et al. 2009].

Devido ao potencial das impressões digitais na identificação de criminosos, as agências de segurança investiram em um estudo rigoroso sobre o tema, desenvolvendo métodos científicos para a comparação visual das digitais e instituindo programas para treinamentos de especialistas na área [Maltoni et al. 2009]. 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 [Maltoni et al. 2009].

Em geral, os AFIS usam minúcias do tipo terminação e bifurcação como atributo discriminante [Dorizzi et al. 2009]. 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 [Goodfellow et al. 2016].

Redes convolucionais vêm ganhando destaque pelo seu desempenho na solução de problemas de classificação e detecção relacionados a imagens. No campo da biometria têm surgido trabalhos baseados nessa tecnologia para classificar impressão digital [Wu et al. 2019], detectar singularidades [Qin et al. 2017], e detectar minúcias [Nguyen et al. 2018].

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 deslocando-se em bloco de tamanho 50×50 a um passo p . Este trabalho é uma variação do trabalho proposto em [Silva et al. 2020], que utilizou imagens provenientes do banco FVC2006. Além disso, neste trabalho são usadas as informações extraídas pelo *VeryFinger SDK*, da empresa *Neurotechnology*[©], para criação do banco de singularidades usado no treinamento e teste. Há também diferença na configuração da arquitetura da rede, neste foram usados 64 filtros de tamanho 5×5 nas duas primeiras camadas. E, a quantidade de épocas usadas no treinamento do modelo deste trabalho foi reduzida para 30.

2. Trabalhos Relacionados

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 baseia-se no *Plindex*, um campo vetorial e uma curva que envolve esse campo, foi proposto por [Kawagoe and Tojo 1984]. 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 baseiam-se no *Plindex* ou usam-no como parte do processo de detecção e classificação de singularidades, como em [Jin and Kim 2010, Zhou et al. 2007].

Alguns algoritmos agrupam as orientações da imagem de orientação de acordo com suas similaridades, formando conjuntos, e estes são separados por linhas que determinam suas fronteiras. As interseções entre essas linhas fornecem a localização das singularidades, como apresentado em [Ramo et al. 2001, Huang et al. 2007, Gupta and Gupta 2015]. Puneet e Phalguni [Gupta and Gupta 2015] 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, como em [Nilsson and Bigun 2003, Jain et al. 2000]. O estado da arte em [Zhu et al. 2016]. Awad e Baba [Awad and Baba 2012] 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 reposta 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, como em [Chen et al. 2011, Qi and Liu 2014] - o trabalho [Qi and Liu 2014], considerado estado da arte em [Zhu et al. 2016], 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 [Qin et al. 2017], 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 [Qi and Liu 2014] 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 a qualidade da imagem. No entanto, ele apenas classifica as singularidades, não detecta. Como os métodos encontrados na literatura classificam e detectam singularidades conjuntamente, não há nenhuma comparação com outros trabalhos.

3. Materiais e Métodos

Abaixo é apresentado o ambiente de desenvolvimento (Subseção 3.1), informações sobre a base de dados (Subseção 3.2), a ferramenta usada para gabaritar a classe Neg (Subseção 3.3), métricas de avaliação (Subseção 3.4) e o detalhamento do método de classificação proposto neste trabalho (Subseção 3.5).

3.1. Ambiente de Desenvolvimento

Foi utilizada a linguagem de programação *Python* 3.6, junto com a API de rede neural Keras 2.2.0, a biblioteca Tensorflow GPU 1.8.0 e a aplicação *web Jupyter Notebook*, tudo instalado em um computador com processador Intel *Core™* i5-3537U 1.8 GHz, 8GB de memória RAM DDR3, placa de vídeo da Nvidia GeForce GT 730M, e sistema operacional Microsoft® Windows 10 de 64 bits.

3.2. Base de dados

A base de dados FVC2002-1 foi concedida pela competição de verificação de impressão digital (*Fingerprint Verification Competition*) [Dorizzi et al. 2009]. Essa base possui 800 imagens que foram coletadas por meio de sensores óticos com resolução de 500 dpi, em escala de cinza de 256 níveis, e dimensão 640x480. A Tabela 1 apresenta a quantidade de singularidades presente na base de dados FVC2002-1.

Tabela 1. Quantidade de singularidades presente na base de dados antes e depois de aplicar o *ImageDataGenerator* do Keras.

	Laço	Delta	Neg	Total
Antes	846	283	1074 ¹	2203
Depois	2138	2131	2124	6393

¹ Essas amostras foram extraídas por meio do Marcador de Singularidades (Subseção 3.3)

3.3. Marcador de Singularidades

Foi desenvolvida uma ferramenta de código aberto com objetivo de gerar os bancos de dados para treinar o modelo que classifica singularidades - *Fingerprint Label Marker* [Silva 2019]. O marcador de singularidades possibilita o usuário marcar, rotular e salvar uma região da imagem original, gerando novas imagens, de menor dimensão, a partir da imagem original - essa região é definida a partir do ponto da imagem que recebe o clique do usuário, ele é usado como ponto central de um quadrado de dimensão 50x50 pixels. Os rótulos de cada imagem marcada pelo usuário depende do botão do mouse que é clicado: o botão esquerdo do mouse marca imagens como Laço (na cor azul); o botão central como Delta (na cor verde) e o direito como Neg (na cor vermelho) - conforme a Figura 2. Todas as imagens geradas pelo usuário são salvas em pastas cujo nome corresponde ao rótulo da classe marcada pelo usuário. Além disso, é gerado um arquivo .txt contendo informações (x, y, rótulo) para cada região definida pelo usuário. Para baixar a ferramenta ou obter mais informações, acesse a página do Github.

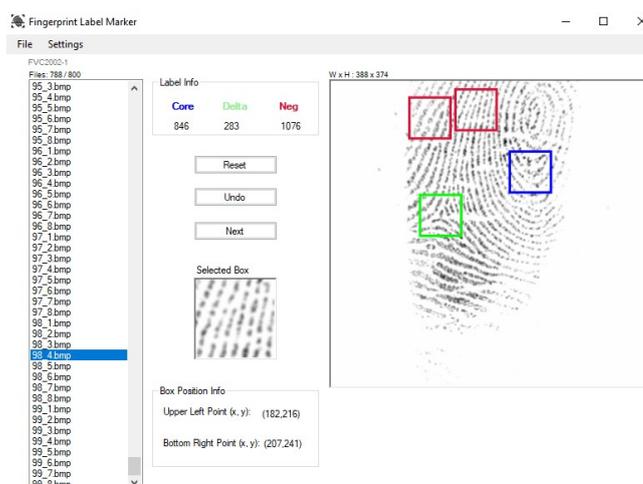


Figura 2. Marcador de singularidades *Fingerprint Label Marker*.

3.4. Métricas

Neste trabalho é usada matriz de confusão e as métricas de avaliação acurácia (*Acc*) e medida-F da biblioteca *scikit-learn*.

Tabela 2. Matriz de confusão.

		Classe Predita	
		Positivo	Negativo
Gabarito	Positivo	VP	FN
	Negativo	FP	VN

A matriz de confusão permite visualizar o desempenho de métodos de classificação em que a visão da perspectiva das linhas representa as classes do gabarito e a visão da perspectiva das colunas representa as classes preditas pelo método de classificação.

Os elementos da matriz de confusão (Tabela 2) VP, FP, FN e VN são verdadeiro positivo, falso positivo, falso negativo e verdadeiro negativo, respectivamente. Estes elementos são usados na formulação das métricas acurácia (*Acc*) e *medida-F* abaixo.

A acurácia (*Acc*) é definida por:

$$acurácia = \frac{VP + VN}{VP + FP + FN + VN} \quad (1)$$

Como a acurácia não é uma métrica robusta, foi utilizada a medida-F, que é definida por:

$$precisão = \frac{VP}{VP + FP} \quad (2)$$

$$revocação = \frac{VP}{VP + FN} \quad (3)$$

$$medida-F = 2 \times \frac{precisão \times revocação}{precisão + revocação} \quad (4)$$

A *medida-F* varia entre [0, 1] e quanto maior forem os valores de *precisão* e *revocação*, maior será o seu valor, indicando que o método alcançou desempenho elevado.

3.5. Método Proposto

Este trabalho propõe um método de classificação de singularidades em imagens de impressão digital baseado em rede neural convolucional. A arquitetura da rede LeNet [LeCun et al. 1998] foi usada como base no desenvolvimento do classificador - Figura 3. O método proposto é dividido em 4 etapas:

1. Baseado nas informações de coordenadas da posição e tipo de singularidade, x , y e t , respectivamente, que são extraídas pelo *VeryFinger SDK* da empresa *Neurotechnology*[©], são geradas novas imagens, de dimensão 50x50, a partir do ponto central (x,y) , que irão formar a base de dados para os tipos (t) ou classes Laço e Delta;
2. Extração de imagens, de dimensão 50x50, de singularidades do tipo Neg por meio do marcador de singularidades *Fingerprint Label Marker* (Subseção 3.3);
3. Por meio do *ImageDataGenerator* do *Keras* [Chollet et al. 2015], as bases de dados são ampliadas, estaticamente, conforme os parâmetros definido na Tabela 3;
4. O treinamento do classificador por meio de Rede Neural Convolucional do *Keras* [Chollet et al. 2015].

Tabela 3. Parâmetros usados no *ImageDataGenerator*.

Parâmetro	Valor
<i>rotation_range</i>	180
<i>horizontal_flip</i>	True
<i>vertical_flip</i>	True
<i>zoom_range</i>	0.2

A partir da base da FVC2002-1 e por meio do gabarito obtido pelo *VeryFinger SDK* da empresa *Neurotechnology*[©], é gerada a base de dados com imagens de dimensão 50x50 para as classes Laço e Delta - as imagens da classe Neg são geradas manualmente por meio do Marcador de Singularidades (Subseção 3.3). Após a criação da base de dados para as três classes, para aumentar o número de amostras e equilibrar a distribuição de imagens entre as classes, é utilizado, uma única vez de maneira estática ou *offline*, o *ImageDataGenerator* do *Keras* configurado com os parâmetros definidos na Tabela 3.

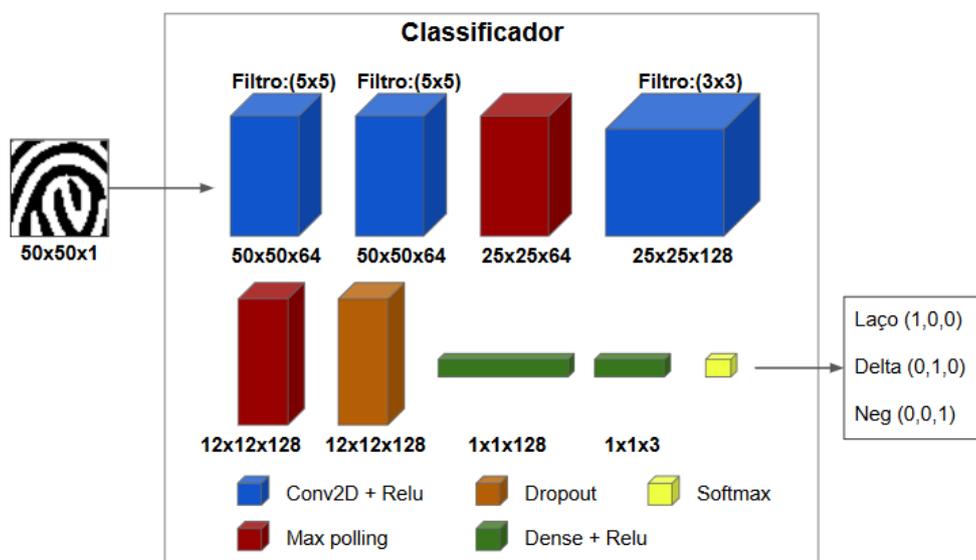


Figura 3. Visão geral do classificador.

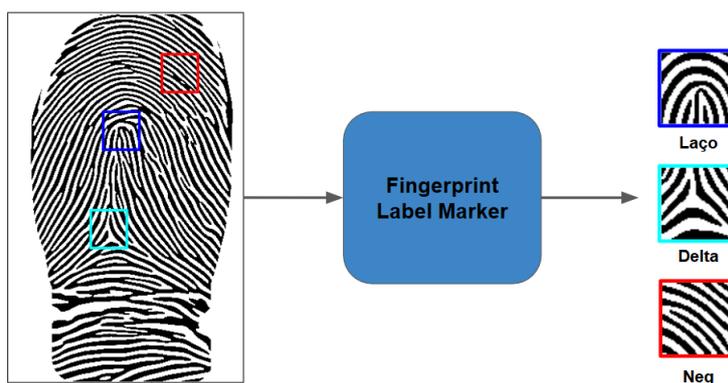


Figura 4. Extração de singularidades por meio do *Fingerprint Label Marker*.

Após a criação das bases aumentadas, para tratar possíveis problemas de superajuste, as imagens (50x50x1) são carregadas em memória e particionadas em 3 conjuntos: treino (90%), validação (5%) e teste (5%). Esta divisão é feita pelo método *train_test_split* da biblioteca *scikit-learn* [Pedregosa et al. 2011], que o faz de forma aleatória e estratificada, de modo que as proporções das classes (Laço, Delta e Neg) são mantidas em cada partição. Os rótulos das classes são representados seguindo o formato *one hot encoding* (Tabela 4) e armazenados em um *NumPy Array* [Oliphant 2006]. Em seguida, os valores dos pixels das imagens são convertidos de inteiro para real (*float32*) e normalizados entre [0,1] pela divisão por 255. Os conjuntos das imagens de treino e validação são usados para treinar o classificador, e o de teste para testar e calcular a matriz de confusão e as métricas acurácia (Equação 1), precisão (Equação 2), revocação (Equação 3) e medida-F (Equação 4).

Tabela 4. Representação dos rótulos (*encoding*) das classes Laço, Delta e Neg.

Classe	Representação
Laço	[1,0,0]
Delta	[0,1,0]
Neg	[0,0,1]

Tabela 5. Alguns dos hiper-parâmetros usados no modelo.

Nome	Valor
<i>batch_size</i>	128
<i>epochs</i>	30
<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>

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* - veja a Figura 3 e a Tabela 5.

Tabela 6. Sumário do classificador.

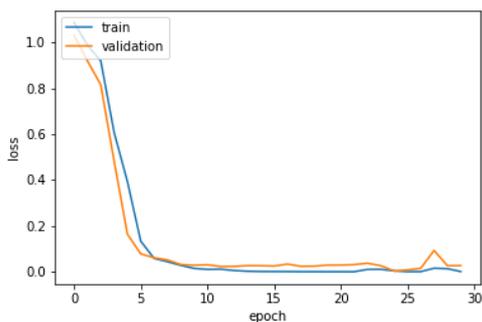
Camada	Saída	Filtro	Passo	Param #
Conv2D	(None, 50, 50, 64)	(5, 5)	1	1664
Conv2D	(None, 50, 50, 64)	(5, 5)	1	102464
MaxPooling	(None, 25, 25, 64)	-	(2, 2)	0
Conv2D	(None, 25, 25, 128)	(3, 3)	1	73856
MaxPooling	(None, 12, 12, 128)	-	(2, 2)	0
Dropout	(None, 12, 12, 128)	-	-	0
Flatten	(None, 18432)	-	-	0
Dense	(None, 128)	-	-	2359424
Dense	(None, 3)	-	-	387
Total de parâmetros:				2,537,795

4. Apresentação e Análise dos Resultados

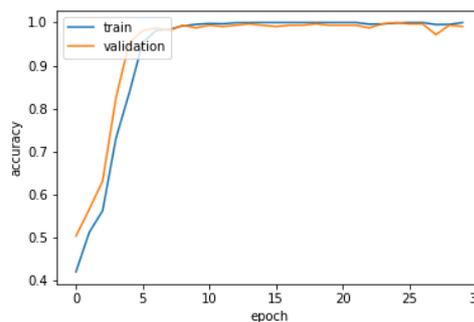
Apesar do número de imagens de singularidades na base de dados não ser elevado, após a ampliação totalizou 6393 amostras no geral (2138 Laços, 2131 Deltas e 2124 Negs), o modelo alcançou acurácia e medida-F de 100%, como mostram as Tabelas 7, 8 e 9-11 - esse valor de medida-F foi obtido por meio da média das medidas-F das classes Laço, Delta e Neg. Esse desempenho alcançado não garante que o modelo esteja pronto para ser usado como parte de um processo de detecção em um ambiente de produção, pois esses resultados de acurácia e medida-F podem variar dependendo das imagens de singularidades que estão no conjunto de teste, visto que o processo de particionamento dos conjuntos de imagens de treino, validação e teste é aleatório.

Tabela 7. Matriz de confusão.

	Laço	Delta	Neg
Classe verdadeira	Laço	107	0
	Delta	0	107
	Neg	0	0
		Classe Predita	



(a)



(b)

Figura 5. Perda ou *loss* (a) e acurácia ou *accuracy* (b) durante o treinamento do classificador sobre o conjunto de treinamento (*train*) e validação (*validation*).

Observando os resultados obtidos pelo classificador, pode parecer que a CNN está super-ajustada aos dados, porém, isso é pouco provável pois além do particionamento das imagens nos conjuntos de treino, validação e teste, foi adicionada a camada do tipo *dropout*, que, na etapa de treinamento, zera ou desativa aleatoriamente algumas conexões de saída da camada anterior. Além disso, o classificador mostrou no treinamento que conseguiu generalizar pois obteve desempenho semelhante, nos gráficos de perda (*loss*) na Figura 5(a) e de acurácia (*accuracy*) na Figura 5(b), durante as 30 épocas (*epoch*), em ambos os conjuntos de treinamento (linha azul) e validação (linha laranja).

Tabela 8. Acurácia do conjunto de teste de classificação de singularidades separada por classe.

Laço	Delta	Neg
1	1	1

Tabela 9. Medida-F, precisão e revocação do conjunto de teste de classificação de singularidades para a classe Laço.

Medida-F	Precisão	Revocação
1	1	1

Tabela 10. Medida-F, precisão e revocação do conjunto de teste de classificação de singularidades para a classe Delta.

Medida-F	Precisão	Revocação
1	1	1

Tabela 11. Medida-F, precisão e revocação do conjunto de teste de classificação de singularidades para a classe Neg.

Medida-F	Precisão	Revocação
1	1	1

O método proposto neste trabalho apenas classifica as singularidades em 3 classes. Diferente dos métodos citados na Seção 2, o método apresentado neste trabalho apenas classifica as singularidades, não detecta, e por esse motivo não há resultados comparativos com outros métodos.

5. Conclusão

Este trabalho apresentou uma ferramenta de código aberto para criação de gabarito de singularidades para geração de base de dados e propôs um método de classificação de singularidades baseado em CNN que apresentou desempenho promissor, porém ele precisa ser testado sobre outras bases de dados. Para trabalhos futuros, é necessários obter mais bases de imagens com gabarito de singularidades e criar um modelo combinado com o classificador proposto neste trabalho para detectar singularidades em imagens de impressão digital.

Referências

- Awad, A. I. and Baba, K. (2012). Singular point detection for efficient fingerprint classification. *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, 2(1):1–7.
- Chen, H., Pang, L., Liang, J., Liu, E., and Tian, J. (2011). Fingerprint singular point detection based on multiple-scale orientation entropy. *IEEE Signal Processing Letters*, 18(11):679–682.
- Chollet, F. et al. (2015). Keras. <https://keras.io>.
- Dorizzi, B., Cappelli, R., Ferrara, M., Maio, D., Maltoni, D., Houmani, N., Garcia-Salicetti, S., and Mayoue, A. (2009). Fingerprint and on-line signature verification competitions at icb 2009. In *International Conference on Biometrics*, pages 725–732. Springer.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. Disponível em: <http://www.deeplearningbook.org>, Acesso em: 16 de set. 2019.
- Gupta, P. and Gupta, P. (2015). A robust singular point detection algorithm. *Applied Soft Computing*, 29:411–423.
- Huang, C.-Y., Liu, L.-m., and Hung, D. D. (2007). Fingerprint analysis and singular point detection. *Pattern Recognition Letters*, 28(15):1937–1945.

- Jain, A. K., Prabhakar, S., Hong, L., and Pankanti, S. (2000). Filterbank-based fingerprint matching. *IEEE transactions on Image Processing*, 9(5):846–859.
- Jin, C. and Kim, H. (2010). Pixel-level singular point detection from multi-scale gaussian filtered orientation field. *Pattern Recognition*, 43(11):3879–3890.
- Kawagoe, M. and Tojo, A. (1984). Fingerprint pattern classification. *Pattern recognition*, 17(3):295–303.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Maltoni, D., Maio, D., Jain, A., and Prabhakar, S. (2009). *Handbook of fingerprint recognition*. Springer Science & Business Media.
- Nguyen, D.-L., Cao, K., and Jain, A. K. (2018). Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge. In *2018 International Conference on Biometrics (ICB)*, pages 9–16. IEEE.
- Nilsson, K. and Bigun, J. (2003). Localization of corresponding points in fingerprints by complex filtering. *Pattern Recognition Letters*, 24(13):2135–2144.
- Oliphant, T. E. (2006). *A guide to NumPy*, volume 1. Trelgol Publishing USA.
- 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., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Qi, J. and Liu, S. (2014). A robust approach for singular point extraction based on complex polynomial model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 78–83.
- Qin, J., Han, C., Bai, C., and Guo, T. (2017). Multi-scaling detection of singular points based on fully convolutional networks in fingerprint images. In *Chinese Conference on Biometric Recognition*, pages 221–230. Springer.
- Ramo, P., Tico, M., Onnia, V., and Saarinen, J. (2001). Optimized singular point detection algorithm for fingerprint images. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 242–245. IEEE.
- Ross, A. A., Nandakumar, K., and Jain, A. (2006). *Handbook of multibiometrics*, volume 6. Springer Science & Business Media.
- Silva, P. (2019). Fingerprint label marker. Disponível em: <<https://github.com/PaulRps/Fingerprint-Label-Marker>>. Acesso em: 16 de jun. 2019.
- Silva, P., Vidal, L., Silva, A., Janduy, J., and Dantas, D. (2020). Classificação de singularidades em imagens de impressão digital baseada em redes neurais convolucionais. *Comunicações em Informática*, 4(1):3–6.
- Wu, F., Zhu, J., and Guo, X. (2019). Fingerprint pattern identification and classification approach based on convolutional neural networks. *Neural Computing and Applications*, pages 1–10.
- Zhou, J., Gu, J., and Zhang, D. (2007). Singular points analysis in fingerprints based on topological structure and orientation field. *Advances in Biometrics*, pages 261–270.

Zhu, E., Guo, X., and Yin, J. (2016). Walking to singular points of fingerprints. *Pattern Recognition*, 56:116–128.