16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

Remoção de Ruído em Imagens de Impressão Digital com Redes Neurais Convolucionais

LUIS H. SACCHI[1], NATHAN F. S. ABREU[2]

[1] Doutor em Engenharia Elétrica, Professor, IFSP, Campus Salto, sacchi@ifsp.edu.br.
[2] Graduando em Ciência da Computação, Voluntário, IFSP, Campus Salto, n.abreu@ifsp.edu.
Área de conhecimento (Tabela CNPq): 1.03.00.00-7 Ciência da Computação

RESUMO: Impressões digitais capturadas por sensores biométricos, por vezes sofrem interferência de ruído, decorrente de múltiplos fatores, como opacidade do cristal pelo excesso de uso, sujeira, desgaste natural, etc. Tendo em vista esse problema, o propósito deste projeto é fazer uso de Redes Neurais Convolucionais, especificamente a arquitetura de Autoencoders, para remover ruído de imagens de impressões digitais capturadas por sensores biométricos.

PALAVRAS-CHAVE: rede neural; autoencoders; visão computacional; digital; ruído.

Noise Removal in Fingerprint Images Using Convolutional Neural Networks

ABSTRACT: Fingerprint images captured by biometric sensors are sometimes affected by noise interference, resulting from multiple factors such as crystal opacity due to excessive use, dirt, natural wear, and others. In view of this problem, the purpose of this project is to employ Convolutional Neural Networks, specifically the Autoencoder architecture, to remove noise from fingerprint images captured by biometric sensors.

KEYWORDS: neural network; autoencoders; computer vision; fingerprint; noise.

INTRODUÇÃO

Softwares de identificação do tipo *Automated Fingerprint Identification System* (AFIS) precisam identificar com precisão as minúcias das digitais. Minúcias são os detalhes únicos de cada digital responsáveis por diferenciá-las. Entretanto, em muitos casos ocorre de fatores externos como acúmulo de sujeira e oleosidade da pele dos usuários, entre outros, acabam comprometendo a leitura de uma digital. Dessa forma faz-se necessário um bom pré-processamento da imagem para que o software possa identificar corretamente a digital, de forma que os detalhes sejam realçados. Souza Neto et al. (2022) conseguiram bons resultados com RNC Autoencoder para remover ruído de imagens de impressões digitais, porém não foi avaliado o ganho de desempenho ao submeter as imagens reconstruídas para realizar identificação em softwares do tipo AFIS, algo que será realizado neste projeto de pesquisa.

O processo de treinamento de um Denoising Autoencoder exige uma composição especial do *dataset* de treinamento. É preciso haver pares de imagens com ruído (sujas) e sem ruído (limpas) correspondentes. Uma imagem com ruído é alimentada na entrada da rede e a saída da rede é comparada com a imagem sem ruído correspondente. Funções de perda (*loss function*) especiais são usadas para mensurar quão a saída produzida difere da imagem sem ruído. O erro percebido na saída é usado para atualizar os pesos da rede neural com o algoritmo de *Backpropagation*. Esse treinamento feito com imagens sujas e limpas leva a rede a criar representações latentes que desprezam o ruído, permitindo assim, que a imagem limpa possa ser devidamente reconstruída na saída após o processo de treinamento.

MATERIAL E MÉTODOS

A rede neural foi desenvolvida na plataforma Google Colaboratory. A linguagem de programação utilizada foi o Python com a biblioteca de *deep learning* Tensorflow/Keras(GOOGLE, 2025a). O software SFinGe disponível em Capelli (2009) e *University of Bologna* (2025), foi utilizado para gerar impressões digitais artificialmente. Foi criado um *dataset* com 1500 digitais e que foi usado no treinamento da rede neural.

A metodologia da pesquisa foi estruturada da seguinte forma:

1. Conseguir *datasets* suficientemente grandes para garantir que o treinamento promova a generalização da RNC.
2. Criar ruído artificialmente nas imagens dos *datasets* disponíveis, de modo que simule ruído presente imagens capturadas com sensores reais.
3. Realizar os pré-processamentos nas imagens do *dataset* para ficarem com as características das imagens capturadas pelo sensor biométrico Digital Persona 4500, que é nosso sensor de referência.
4. Treinar redes neurais a fim de fazer ajustes na mesmo ou no *dataset*, para reduzir as funções de perda (*loss function*) e obter resultados adequados.

RESULTADOS E DISCUSSÃO

A rede neural foi construída com a seguinte disposição de camadas: Convolucional, MaxPooling, Convolucional, MaxPooling, Convolucional, MaxPooling, Convolucional, MaxPooling, Convolucional, UpSampling, Convolucional, UpSampling, Convolucional, UpSampling, Convolucional, UpSampling, Convolucional. A rede foi submetida a treinamentos com dois *datasets* de digitais sujas e limpas a fim de aprender a reconstruir uma digital retirando o ruído antes presente nela.

FIGURA 1. Fragmento de código mostrando a disposição das camadas da rede neural utilizada.

```
input_data = tf.keras.layers.Input(shape=(290, 384, 1))

# Encoder
encoder = tf.keras.layers.Conv2D(64, (5,5), activation='relu', padding='same')(input_data)
encoder = tf.keras.layers.MaxPooling2D((2,2), padding='same')(encoder)

encoder = tf.keras.layers.Conv2D(128, (3,3), activation='relu', padding='same')(encoder)
encoder = tf.keras.layers.MaxPooling2D((2,2), padding='same')(encoder)

encoder = tf.keras.layers.Conv2D(256, (3,3), activation='relu', padding='same')(encoder)
encoder = tf.keras.layers.MaxPooling2D((2,2), padding='same')(encoder)

encoder = tf.keras.layers.Conv2D(512, (3,3), activation='relu', padding='same')(encoder)
encoder = tf.keras.layers.MaxPooling2D((2,2), padding='same')(encoder)

# Decoder
decoder = tf.keras.layers.Conv2DTranspose(512, (3,3), activation='relu', padding='same')(encoder)
decoder = tf.keras.layers.UpSampling2D((2,2))(decoder)

decoder = tf.keras.layers.Conv2DTranspose(256, (3,3), activation='relu', padding='same')(decoder)
decoder = tf.keras.layers.UpSampling2D((2,2))(decoder)

decoder = tf.keras.layers.Conv2DTranspose(128, (3,3), activation='relu', padding='same')(decoder)
decoder = tf.keras.layers.UpSampling2D((2,2))(decoder)

decoder = tf.keras.layers.Conv2DTranspose(64, (5,5), activation='relu', padding='same')(decoder)
decoder = tf.keras.layers.UpSampling2D((2,2))(decoder)

decoded = tf.keras.layers.Conv2DTranspose(1, (5,5), activation='relu', padding='same')(decoder)
```

FIGURA 2. Representação gráfica das camadas da rede neural.

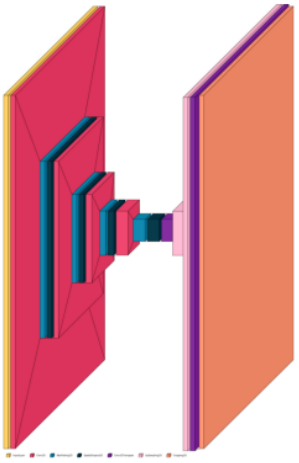
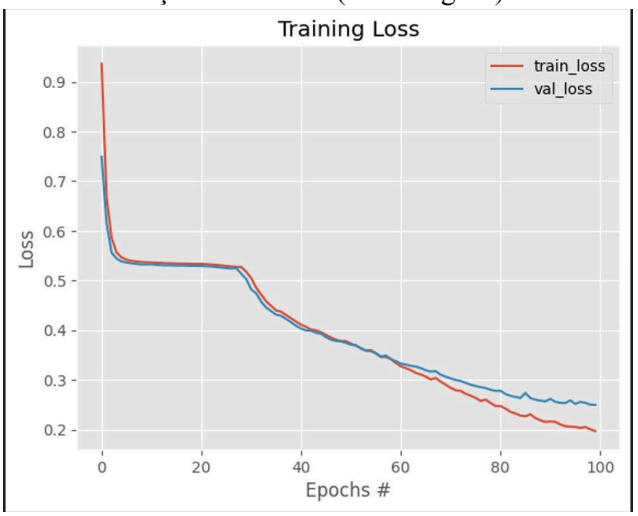


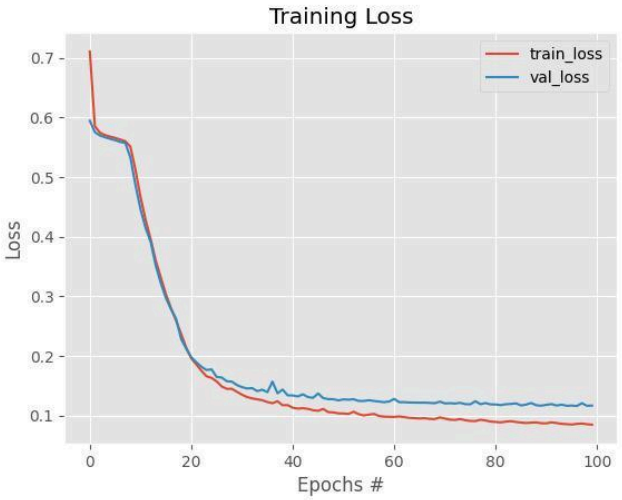
FIGURA 3. Evolução da *loss function* ao longo das épocas de treinamento para dados de treinamento e de validação - *dataset* 1 (408 imagens).



A princípio utilizando um *dataset* de 408 imagens para treinamento. O gráfico demonstra a queda da *loss function* ao decorrer das épocas de treinamento, tanto com as imagens do *dataset* de treinamento (90%) quanto com de validação (10%), imagens nunca vistas pelo modelo.

Foi possível chegar a uma taxa de erro de 0,2 para as imagens de treinamento e 0,25 para as imagens de validação. Houve alguma dificuldade da rede em generalizar, temos algum efeito danoso de overfitting presente e que pode ser verificado no gráfico com o afastamento exagerado entre *loss function* em dados de treinamento e de validação. O tamanho pequeno do dataset também impediu a loss function a descer a níveis mais próximos de zero.

FIGURA 4. Evolução da *loss function* ao longo das épocas de treinamento para dados de treinamento e de validação - *dataset* 2 (1500 imagens).

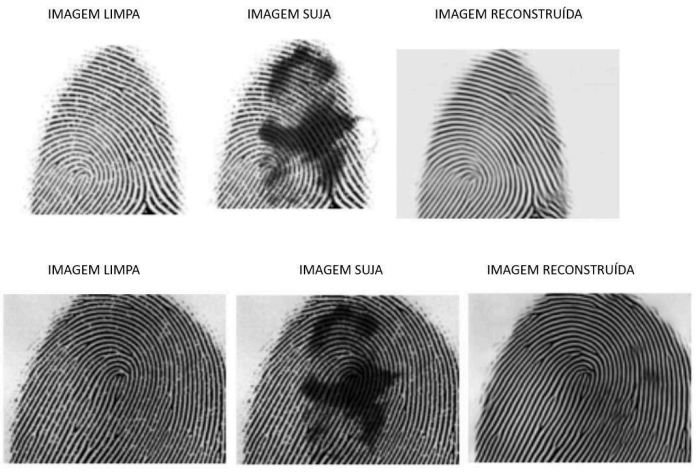


Posteriormente outro treinamento feito foi com um *dataset* com 1500 digitais, de forma que a rede tivesse mais material para aprender e assim chegar a resultados melhores.

Neste gráfico é possível notar que tanto o erro para o *dataset* em dados de treino quanto de validação reduziram comparados ao teste anterior, ficando abaixo de 0,25 para o conjunto de validação e abaixo de 0.1 para o conjunto de treinamento. Além disso é possível notar que com este *dataset* a *loss function* teve uma queda brusca ainda no intervalo das épocas entre 0 e 20 enquanto no treino anterior essa queda demorou muito mais ciclos de treinamento para ocorrer.

Sendo assim é possível notar a diferença que um *dataset* com mais conteúdo faz no treinamento do modelo.

FIGURA 5. Comparação entre as imagens originais, sujas e reconstruídas do *dataset* de validação.



Após o treinamento com o *dataset* 2 de 1500 digitais foi possível chegar a resultados bastante promissores, como mostra a figura anterior. O modelo consegue reconstruir digitais limpas de uma forma satisfatória.

Para trabalhos futuros o uso de software AFIS se faz necessário para uma avaliação de resultados mais acurada, será possível determinar se o modelo consegue reconstruir imagens capturadas por um sensor biométrico. A utilização de exemplares de digitais capturadas por sensores reais é mais um teste a se fazer, sendo assim possível comprovar se a rede neural consegue ser generalizada para casos além das imagens artificialmente geradas.

A implementação do modelo em hardware de TV BOX's descaracterizadas também é um objetivo futuro, pois dessa forma será possível utilizar o modelo atuando localmente em um sistema de verificação de digitais de baixo custo, como no projeto IFSPresente, IFSPresente(2025).

CONCLUSÕES

Dado os resultados obtidos é possível constatar que uma rede neural do tipo Autoencoder é capaz de retirar ruídos de digitais e reconstruí-las de forma coerente à original. O resultado visual é notório. Entretanto, para trabalhos futuros ainda se faz necessário a realização de ajustes na arquitetura da rede utilizada a fim de otimizar os resultados da melhor forma e obter o melhor modelo possível. Testes de validação com softwares como AFIS se fazem necessários, além do teste práticos com capturas de sensores reais e sistemas embarcados, com o foco na aplicação em hardware de TV BOX's descaracterizadas.

CONTRIBUIÇÕES DOS AUTORES

L.H.S. contribuiu com a criação do modelo e experimentos N.F.S.A. procedeu com a criação e avaliação dos datasets, redação do trabalho.

Todos os autores contribuíram com a revisão do trabalho e aprovaram a versão submetida.

AGRADECIMENTOS

Agradecimentos a toda equipe do projeto TV-BOX e IFSPresente do IFSP Salto.

REFERÊNCIAS

BALDI, P.; HORNIK, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, v. 2, n. 1, p. 53-58, 1989. DOI: 10.1016/0893-6080(89)90014-2.

CAPPELLI, R. SFinGe. In: LI, S. Z.; JAIN, A. (eds.). *Encyclopedia of Biometrics*. Boston, MA: Springer, 2009. Disponível em: https://doi.org/10.1007/978-0-387-73003-5_8. Acesso em: 31 mar. 2025.

FOLHA DE S.PAULO. TVs box piratas viram computadores e auxiliam aulas em escolas públicas. Disponível em:https://www1.folha.uol.com.br/educacao/2025/03/tvs-box-piratas-viram-computadores-e-auxiliam-aulas-em-escolas-publicas.shtml. Acesso em: 2 abr. 2025.

GOOGLE. Google Colaboratory. Disponível em: https://colab.research.google.com/. Acesso em: 31 mar. 2025.

GOOGLE. TensorFlow Lite. Disponível em: https://www.tensorflow.org/lite/guide?hl=pt-br. Acesso em: 2 abr. 2025.

HID GLOBAL. Leitor biométrico 4500 Fingerprint Reader. Disponível em: https://www.hidglobal.com/pt/products/4500-fingerprint-reader. Acesso em: 31 mar. 2025.

ICL NOTÍCIAS. TVs box piratas viram computadores e auxiliam aulas em escolas públicas. Disponível em:https://iclnoticias.com.br/tvs-box-piratas-computadores-escolas-publicas/.Acesso em: 2 abr. 2025.

INSTITUTO FEDERAL DE SÃO PAULO - IFSP. Servidores e estudantes do IFSP apresentam resultados de projetos com a Receita Federal. Disponível em: https://ifsp.edu.br/component/content/article/17-ultimas-noticias/4848-servidores-e-estudantes-do-ifsp-apresentam-resultados-de-projetos-com-a-receita-federal. Acesso em: 2 abr. 2025.

SOUZA NETO, Sandoval Veríssimo de. *et al.* Destaque de imagens de impressão digital utilizando autoencoders profundos totalmente convolucionais. *Brazilian Journal of Development*, Curitiba, v. 8, n. 5, p. 40027-40042, maio 2022. DOI: 10.34117/bjdv8n5-474.

UNIVERSITY OF BOLOGNA. *SFinGe – Synthetic Fingerprint Generator*. Disponível em: https://biolab.csr.unibo.it/DatabaseSoftware.asp?organize=Software&pathAttiv=&pathSubj=&selObj=&select=2. Acesso em: 15 ago. 2025.

VAŽAN, Robert. SourceAFIS. Disponível em: https://sourceafis.machinezoo.com/. Acesso em: 31 mar. 2025.