

# Aprendizado Profundo para Classificação de Imagens

Explorando o uso conjunto de redes neurais para classificação e remoção de ruído em imagens de dígitos manuscritos

Autor: Gabriel Eduardo Lima

Orientador: Ricardo de la Rocha Ladeira, Me.

Coorientador: Eder Augusto Penharbel, Me.

IFC - Campus Blumenau

28 de novembro de 2023



**INSTITUTO FEDERAL**  
Catarinense  
Campus Blumenau

1. Introdução
2. Desenvolvimento
3. Conclusão
4. Referências

1. Introdução
  - 1.1. Apresentação do Tema
  - 1.2. Objetivos
  - 1.3. Metodologia
  - 1.4. Justificativas e Contribuições
  - 1.5. Trabalhos Correlatos
2. Desenvolvimento
3. Conclusão
4. Referências

# 1.1 Apresentação do Tema

Inteligência Artificial

*Machine Learning*

*Deep Learning*

Visão Computacional

Classificação de Imagens

Reconhecimento Óptico de Caracteres

Problema: Redes neurais e dados ruidosos [4].

Abordagem: Pré-processamento e *autoencoders* [12, 13].

Questionamento: Desenvolvimento conjunto dos modelos?

## 1.2 Objetivos

Considerando o problema de classificação de dígitos manuscritos em imagens ruidosas.

Objetivo geral: **explorar** o incremento na acurácia (taxa de acertos) de um classificador de imagens por meio do uso e da adaptação de uma rede *autoencoder* para remoção de ruído.

## 1.3 Metodologia

Classificação da pesquisa [14]:

- Quanto à natureza: primária;
- Quanto aos objetivos: exploratória;
- Quanto aos procedimentos: bibliográfica e experimental.

Pesquisa experimental:

- Experimentos I, II, III e IV.

# 1.3 Metodologia

Materiais e ferramentas:

- Redes neurais: *Multilayer Perceptron* MLP;
- Dados: *Modified National Institute of Standards and Technology* (MNIST) [2];
- Ruído: Ruído Gaussiano Estacionário Aditivo;
- Métrica: Acurácia;
- Linguagem: Python (3.11.5)<sup>1</sup>.



---

<sup>1</sup>Códigos disponíveis em: <http://bit.ly/lima001-bcc-tcc>.  
Acessível também pelo código QR.

## 1.4 Justificativas e Contribuição

### Justificativas:

- Relevância do tema;
- Adequação para pesquisa;
- Método científico.

### Contribuição:

- Exploração de ideias (teoria e prática) que não são amplamente estudadas na literatura.



## 1.5 Trabalhos Correlatos

Classificação de imagens de dígitos manuscritos [1, 2, 7].

Classificação de imagens na presença de dados ruidosos [8–11, 16].

Treino de redes neurais em conjunto para solucionar problemas relacionados [6, 15].

1. Introdução
2. Desenvolvimento
  - 2.1. Fundamentação Teórica
  - 2.2. Detalhamento dos Experimentos
  - 2.3. Resultados e Discussões
3. Conclusão
4. Referências

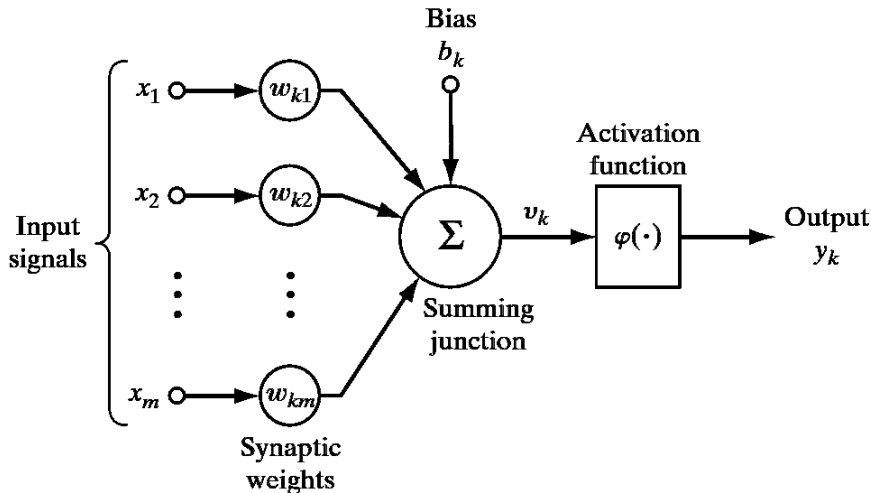
## 2.1.1 Redes Neurais Artificiais

Rede neural artificial: modelar a maneira que o cérebro humano executa uma tarefa. Desenvolvida em hardware ou software. Processo adaptativo (aprendizagem) [5].

Tecnicamente: dispositivo de processamento da dados. Composto por neurônios artificiais.

## 2.1.2 Modelo de Neurônio Artificial

Figura: Modelo teórico de neurônio artificial. Adaptado de Haykin [5, p. 11].



## 2.1.2 Modelo de Neurônio Artificial

$$\vec{\mathbf{x}} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \quad (1)$$

$$\mathbf{W} = [w_{k0} \quad w_{k1} \quad w_{k2} \quad \dots \quad w_{km}] \quad (2)$$

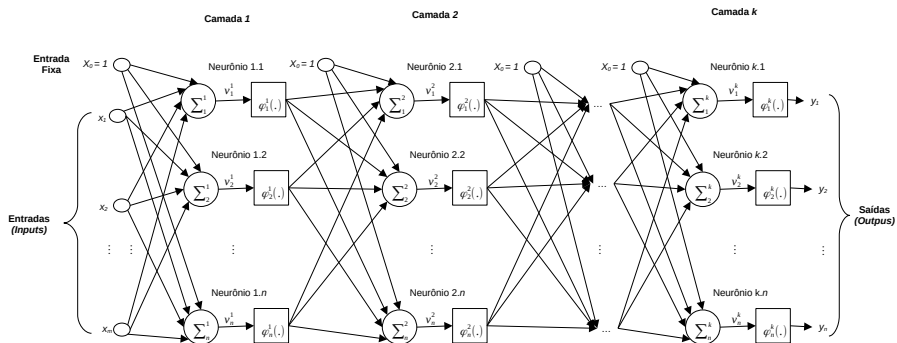
$$\vec{\mathbf{v}}_k = \mathbf{W} \vec{\mathbf{x}} \quad (3)$$

$$\vec{\mathbf{y}}_k = \varphi(\vec{\mathbf{v}}_k) \quad (4)$$

$$\varphi(\vec{\mathbf{v}}_k) = (\varphi(v_1)) \quad (5)$$

## 2.1.3 Arquiteturas de Redes Neurais Artificiais

Figura: Modelo teórico de rede neural *Multilayer Feedforward*. Elaboração própria.



## 2.1.4 Aprendizagem

Algoritmos de aprendizagem.

Treino e Generalização.

Validação e Inferência (Predição).

## 2.1.5 Multilayer Perceptron

Arquitetura *Multilayer Feedforward*.

Treinado pelo algoritmo *backpropagation*.

Otimização de função de erro pela descida do gradiente.

$$MSE = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n (y_j^i - \hat{y}_j^i)^2 \quad (6)$$

$$E[k] = \sum_{j=1}^n (y_j^k - \hat{y}_j^k)^2 \quad (7)$$

$$\hat{w}_{i,j}^m = w_{i,j}^m - \alpha \frac{\partial E}{\partial w_{i,j}^m} \quad (8)$$



## 2.1.6 Modelo de Ruído Estacionário Aditivo Gaussiano

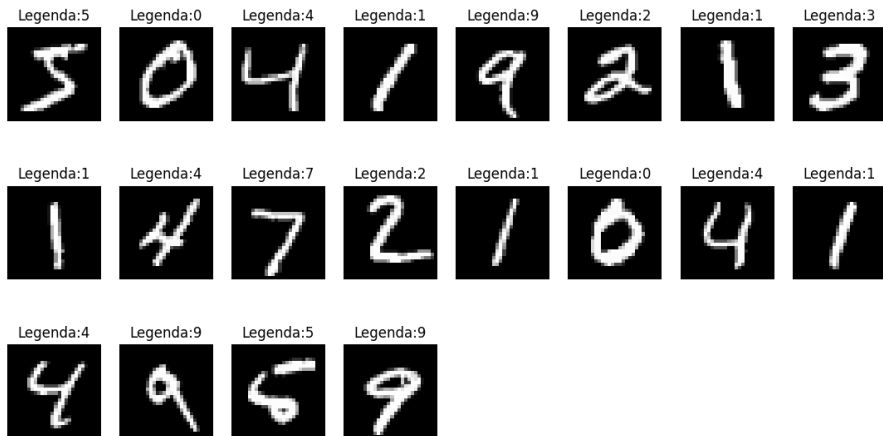
O ruído: não varia com o tempo; é somado ao dado original; é amostrado de uma distribuição de probabilidade normal.

**Figura:** Efeito visual da degradação de uma imagem. Elaboração própria.



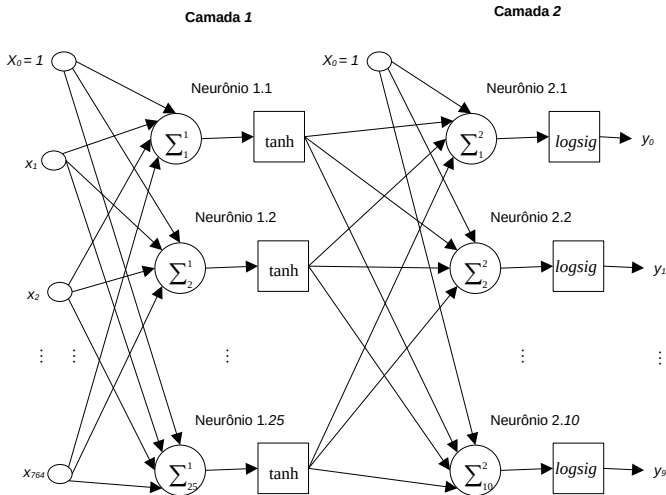
## 2.2.1 Experimento I: Classificação de Imagens

**Figura:** Exemplos de imagens pertencentes ao conjunto de dados MNIST e suas respectivas legendas. Elaboração própria.



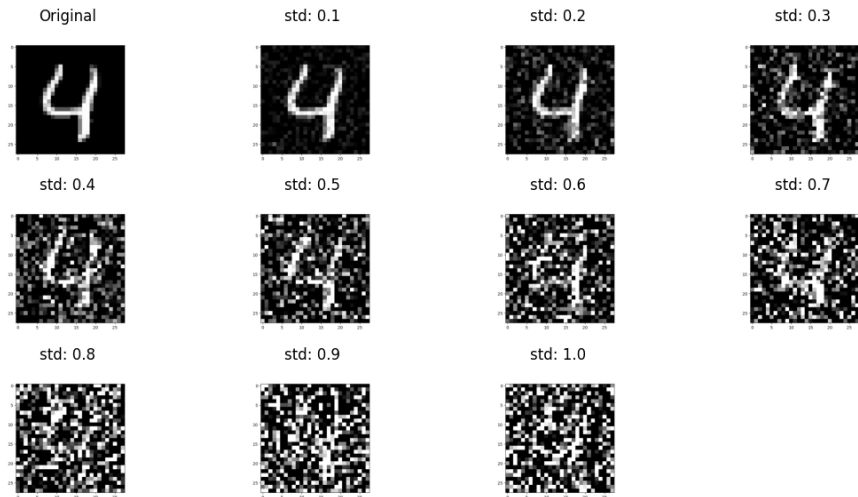
## 2.2.1 Experimento I: Classificação de Imagens

Figura: Arquitetura do modelo de classificação desenvolvido, baseada em [3]



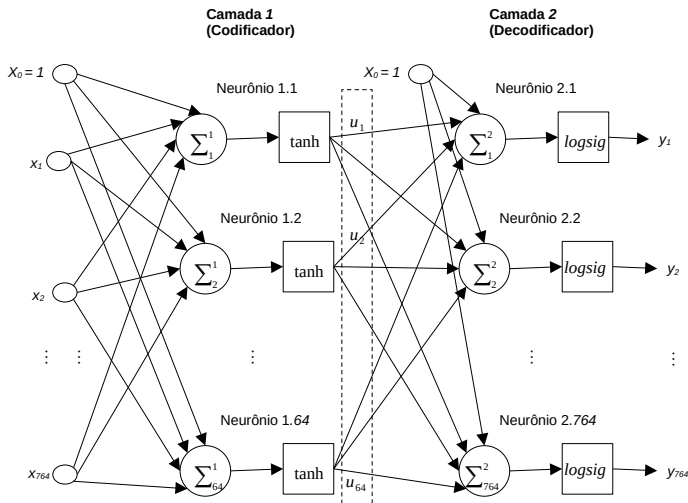
## 2.2.2 Experimento II: Ruído

**Figura:** Imagem (dígito 4) do conjunto de validação para diferentes níveis de ruído. Elaboração própria.



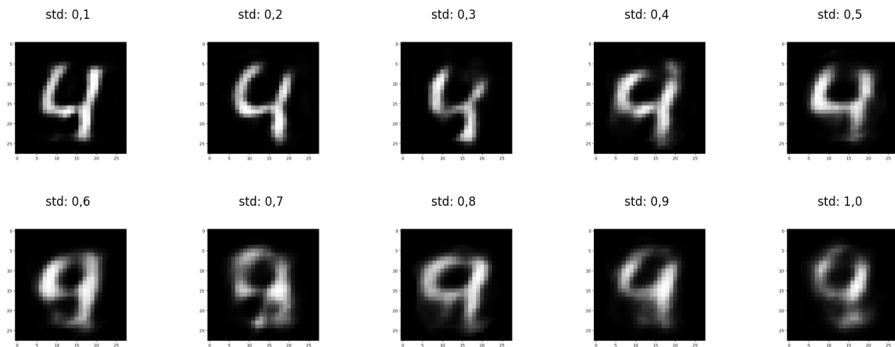
## 2.2.3 Experimento III: Pré-processamento

**Figura:** Arquitetura autoencoder MLP usada implementada pelos modelos de remoção de ruído.



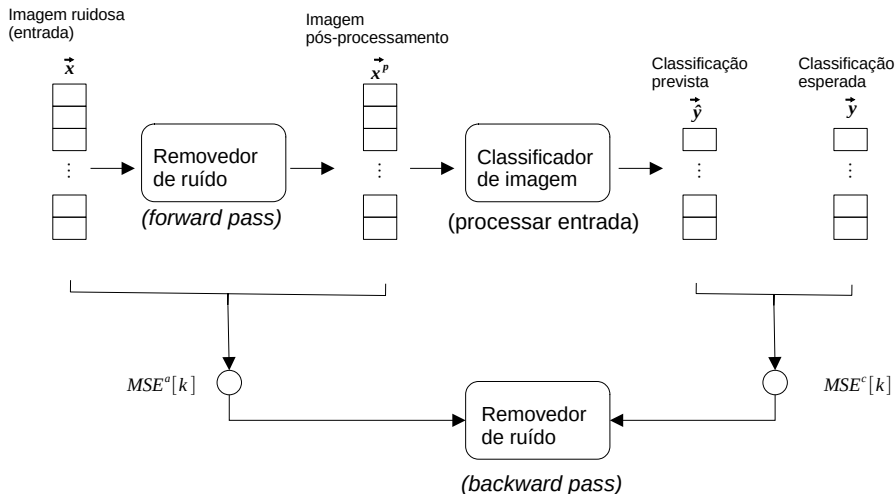
## 2.2.3 Experimento III: Pré-processamento

**Figura:** Saída dos removedores de ruído (10 épocas) para entrada ruidosa referente ao dígito 4. Elaboração própria.



## 2.2.4 Experimento IV: Adaptação

**Figura:** Esquemático da adaptação explorada para treinar um modelo de remoção de ruído. Elaboração própria.



## 2.2.4 Experimento IV: Adaptação

Termo de erro (última camada):

$$eN_i = (-y_i + \hat{y}_i) \cdot \text{logsig}'(v_i) \quad (9)$$

Termo de erro adaptado (última camada):

$$e\hat{N}_i = (\phi(-y_i + \hat{y}_i) + \omega E^c[k]) \cdot \text{logsig}'(v_i) \quad (10)$$

Configurações  $\phi$  e  $\omega$  exploradas:

$$\left( \begin{array}{l} (+1.0; -1.0), (+0,8; -0,2), (+0,5; -0,5), \\ (+1.0; -0,5), (+1.0; +1.0), (+0,5; 0,5), \\ (+1.0; +0,5), (+0,8; +0,2), (+1.0; \pm 0,5) \end{array} \right) \quad (11)$$



## 2.3.1 Análise dos resultados do experimento I

**Tabela:** Acurácia (treino e validação) de classificação em relação às épocas de treino. Elaboração própria.

| Época | Acurácia — treino (%) | Acurácia — validação (%) |
|-------|-----------------------|--------------------------|
| 1     | 85,63                 | 91,57                    |
| 2     | 92,03                 | 92,40                    |
| 3     | 92,74                 | 92,46                    |
| 4     | 93,20                 | 92,95                    |
| 5     | 93,56                 | 92,97                    |
| 6     | 93,84                 | 93,32                    |
| 7     | 94,06                 | 93,50                    |
| 8     | 94,20                 | 93,46                    |
| 9     | 94,40                 | 93,67                    |
| 10    | 94,57                 | 93,61                    |

## 2.3.2 Análise dos resultados do experimento II

**Tabela:** Acurácia média de classificação para conjuntos de validação com diferentes níveis de ruído. Elaboração própria.

| Nível de ruído (std) | Acurácia média de classificação (%) |
|----------------------|-------------------------------------|
| 0,1                  | 88,18                               |
| 0,2                  | 81,66                               |
| 0,3                  | 71,96                               |
| 0,4                  | 61,38                               |
| 0,5                  | 52,15                               |
| 0,6                  | 44,58                               |
| 0,7                  | 38,31                               |
| 0,8                  | 33,72                               |
| 0,9                  | 29,85                               |
| 1,0                  | 27,02                               |

## 2.3.3 Análise dos resultados do experimento III

**Tabela:** Acurácia média de classificação para conjuntos de validação pré-processados pelos respectivos modelos de remoção de ruído treinados em diferentes épocas. Elaboração própria.

| Nível de ruído (std) | Acurácia média de classificação (%) - remove-dor de ruído 1 época | Acurácia média de classificação (%) - remove-dor de ruído 5 épocas | Acurácia média de classificação (%) - remove-dor de ruídos 10 épocas |
|----------------------|---|--|--|
| 0,1                  | 76,92   | 81,56  | 86,73  |
| 0,2                  | 73,71   | 82,51  | 84,76  |
| 0,3                  | 73,35   | 76,49  | 82,38  |
| 0,4                  | 61,69   | 73,09  | 77,47  |
| 0,5                  | 56,57   | 62,58  | 68,24  |
| 0,6                  | 55,89   | 56,94  | 63,71  |
| 0,7                  | 48,24   | 58,75  | 58,08  |
| 0,8                  | 52,85   | 56,53  | 54,58  |
| 0,9                  | 44,37   | 53,11  | 53,17  |
| 1,0                  | 39,70   | 43,75  | 50,85  |

## 2.3.4 Análise dos resultados do experimento IV

**Tabela:** Acurácia média de classificação para conjuntos de validação (com diferentes níveis de ruído) pré-processados pelos respectivos modelos de remoção de ruído treinados com configurações  $(\phi, \omega)$ . Elaboração própria.

| Configuração $(\phi, \omega)$     | Acurácia média de classificação (%) por nível de ruído (std) |       |       |       |       |       |       |       |       |       |
|-----------------------------------|--|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                                   | 0,1  | 0,2   | 0,3   | 0,4   | 0,5   | 0,6   | 0,7   | 0,8   | 0,9   | 1,0   |
| $(\phi = +1,0, \omega = -1,0)$    | 75,80  | 73,96 | 72,72 | 68,29 | 68,22 | 66,31 | 59,35 | 52,64 | 49,42 | 41,46 |
| $(\phi = +0,8, \omega = -0,2)$    | 80,38  | 79,62 | 72,01 | 65,84 | 68,09 | 61,36 | 55,32 | 52,40 | 45,95 | 42,91 |
| $(\phi = +0,5, \omega = -0,5)$    | 70,05  | 74,12 | 71,20 | 69,86 | 69,85 | 68,78 | 62,67 | 54,77 | 49,79 | 42,14 |
| $(\phi = +1,0, \omega = -0,5)$    | 82,66  | 80,82 | 80,29 | 75,75 | 70,28 | 67,75 | 62,88 | 60,69 | 52,65 | 48,26 |
| $(\phi = +1,0, \omega = +1,0)$    | 66,17  | 63,33 | 52,09 | 26,90 | 42,85 | 15,07 | 15,03 | 28,58 | 19,56 | 15,91 |
| $(\phi = +0,5, \omega = +0,5)$    | 57,30  | 60,60 | 34,15 | 42,93 | 15,64 | 9,74  | 9,72  | 24,91 | 9,72  | 9,74  |
| $(\phi = +1,0, \omega = +0,5)$    | 69,82  | 70,55 | 55,74 | 51,29 | 47,94 | 33,60 | 30,57 | 29,64 | 31,84 | 28,24 |
| $(\phi = +0,8, \omega = +0,2)$    | 72,49  | 72,57 | 56,75 | 54,69 | 54,26 | 46,16 | 38,73 | 39,24 | 34,13 | 34,13 |
| $(\phi = +1,0, \omega = \pm 0,5)$ | 73,64  | 74,81 | 60,12 | 60,66 | 57,10 | 51,35 | 50,44 | 42,90 | 43,33 | 37,73 |

1. Introdução
2. Desenvolvimento
3. Conclusão
4. Referências

### 3. Conclusão

Respondendo ao questionamento: Sim, é possível!

A adaptação do algoritmo backpropagation demonstra indícios de ser computacionalmente mais eficiente que o original.

### 3. Conclusão

Trabalhos futuros:

- Reprodução e ampliação dos experimentos conduzidos para testificar e formalizar os indícios observados;
- Estudar cenários similares ao apresentado na pesquisa.

Com o desenvolvimento dos indícios, os conceitos explorados podem ser usados para desenvolver soluções de *Deep Learning* eficientes voltadas para aplicações reais.

# Referências I

- [1] Alejandro Baldominos, Yago Saez e Pedro Isasi. "A survey of handwritten character recognition with MNIST and EMNIST". Em: *Applied sciences* 9.15 (ago. de 2019), p. 16. DOI: 10.3390/app9153169. URL: <http://dx.doi.org/10.3390/app9153169>.
- [2] L. Bottou et al. "Comparison of classifier methods. a case study in handwritten digit recognition". Em: *IAPR*. Vol. 3. IEEE, 1994, pp. 77–82.
- [3] Magnus Ekman. *Learning deep learning. Theory and practice of neural networks, computer vision, natural language processing, and transformers using TensorFlow*. en. 1ª ed. Boston, MA: Addison Wesley, 2021, p. 752.
- [4] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep Learning*. London, England: MIT Press, 2016, p. 775.
- [5] Simon O. Haykin. *Neural Networks and Learning Machines*. 3ª ed. Upper Saddle River, NJ: Pearson, 2008, p. 936.
- [6] Ya Li et al. "Classification and representation joint learning via deep networks". Em: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. California: International Joint Conferences on Artificial Intelligence Organization, 2017.
- [7] Cheng-Lin Liu et al. "Handwritten digit recognition. benchmarking of state-of-the-art techniques". Em: *Pattern recognition* 36.10 (2003), pp. 2271–2285. DOI: 10.1016/s0031-3203(03)00085-2. URL: [http://dx.doi.org/10.1016/s0031-3203\(03\)00085-2](http://dx.doi.org/10.1016/s0031-3203(03)00085-2).
- [8] Mohammad Momeny et al. "A noise robust convolutional neural network for image classification". Em: *Results in engineering* 10 (2021), p. 100225. DOI: 10.1016/j.rineng.2021.100225. URL: <http://dx.doi.org/10.1016/j.rineng.2021.100225>.
- [9] Tiago S. Nazaré et al. "Deep convolutional neural networks and noisy images". Em: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Vol. 10657. Springer International Publishing, fev. de 2018, pp. 416–424.
- [10] Sudipta Singha Roy, Mahtab Ahmed e Muhammad Aminul Haque Akhand. "Classification of massive noisy image using auto-encoders and convolutional neural network". Em: *International Conference on Information Technology (ICIT)*. IEEE, out. de 2017, pp. 971–979.



# Referências II

- [11] Sudipta Singha Roy, Mahtab Ahmed e Muhammad Aminul Haque Akhand. "Noisy image classification using hybrid deep learning methods". Em: *Journal of Information and Communication Technology* 17.2 (abr. de 2018). DOI: 10.32890/jict2018.17.2.8253. URL: <http://dx.doi.org/10.32890/jict2018.17.2.8253>.
- [12] Pascal Vincent et al. "Extracting and Composing Robust Features with Denoising Autoencoders". Em: *International Conference on Machine Learning*. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 1096–1103. DOI: 10.1145/1390156.1390294. URL: <https://doi.org/10.1145/1390156.1390294>.
- [13] Pascal Vincent et al. "Stacked Denoising Autoencoders. Learning Useful Representations in a Deep Network with a Local Denoising Criterion". Em: *J. Mach. Learn. Res.* 11 (dez. de 2010), pp. 3371–3408.
- [14] Raul Sidnei Wazlawick. *Metodologia de Pesquisa para Ciência da Computação*. pt. 3ª ed. Rio de Janeiro, RJ: GEN LTC, 2020, p. 152.
- [15] Qiang Xu et al. "Multi-Task Joint learning model for segmenting and classifying tongue images using a deep neural network". Em: *Journal of biomedical and health informatics* 24.9 (2020), pp. 2481–2489. DOI: 10.1109/JBHI.2020.2986376. URL: <http://dx.doi.org/10.1109/JBHI.2020.2986376>. IEEE.
- [16] Yiren Zhou, Sibong Song e Ngai-Man Cheung. "On classification of distorted images with deep convolutional neural networks". Em: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017.

# Aprendizado Profundo para Classificação de Imagens

Explorando o uso conjunto de redes neurais para classificação e remoção de ruído em imagens de dígitos manuscritos

Autor: Gabriel Eduardo Lima

Orientador: Ricardo de la Rocha Ladeira, Me.

Coorientador: Eder Augusto Penharbel, Me.

IFC - Campus Blumenau

28 de novembro de 2023



**INSTITUTO FEDERAL**  
Catarinense  
Campus Blumenau