

# Pré-processamento de imagens de baixa resolução utilizando Deep Learning baseado em um Autoencoder

Victor L. Costa, Eduardo H. Teixeira, Samuel Mafra and Felipe A. P. de Figueiredo

**Resumo**—O uso de *Machine Learning* na área de visão computacional vem crescendo entre o meio acadêmico e industrial e com isso, diversas técnicas de pré-processamento de *Machine Learning* podem ser usadas para uma ou mais aplicações. Devido a essa demanda, o número de imagens dentro de um *dataset* para realizar o treinamento de modelos de *Machine Learning* precisa ser grande e com uma boa qualidade. Assim, o pré-processamento de imagens pode se tornar crucial para melhorar e aperfeiçoar o desempenho de modelos de *Machine Learning*. Este artigo propõe o desenvolvimento de um *autoencoder* para melhorar a qualidade de imagens de baixa resolução a fim de comparar e analisar qual a melhoria que o pré-processamento pode trazer para um sistema de detecção de objetos.

**Palavras-Chave**—Pré-processamento, *Machine Learning*, Visão Computacional, Detecção de objetos.

**Abstract**—The use of *Machine Learning* in the area of computer vision has been growing among the academic and industrial environments and with that, several *Machine Learning* pre-processing techniques can be used for one or more applications. Due to this demand, the number of images within a *dataset* to train *Machine Learning* models needs to be large and of good quality. Thus, image pre-processing can become crucial to improve and improve the performance of *Machine Learning* models. This article proposes the development of an *autoencoder* to improve the quality of low resolution images in order to compare and analyze the improvement that pre-processing can bring to an object detection system.

**Keywords**—Pre-processing, *Machine Learning*, Computer Vision, Object Detection.

## I. INTRODUÇÃO

O uso de pré-processamento de dados é de suma importância para pesquisadores e engenheiros nas áreas de *Machine Learning* (ML) e Visão Computacional (VC), pois muitos dos dados recolhidos possuem algum tipo de ruído, sejam informações repetidas, erradas ou com dados corrompidos [1]. Isso ocorre em bancos de dados de imagens que possuem imagens de baixa qualidade, imagens com diferentes níveis de tremores ou desfoque e outros ruídos que atrapalham um sistema durante a etapa de detecção/classificação de objetos.

Victor L. Costa (victor.laragnoit@geb.inatel.br), Eduardo H. Teixeira (eduardot@gea.inatel.br), Samuel B. Mafra (samuelbmafra@inatel.br) e Felipe A. P. de Figueiredo (felipe.figueiredo@inatel.br) Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, Departamento de Engenharia de Telecomunicações. This work is partially supported by RNP, with resources from MCTIC, Grant No. No 01245.010604/2020-14, under the Brazil 6G project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações - Inatel), Brazil, the National Council for Scientific and Technological Development-CNPq (403827/2021-3), FAPESP (2021/06946-0)

## II. ARQUITETURA DO SISTEMA

A arquitetura de um *autoencoder* é composta por um *encoder* e um *decoder* que operam em conjunto, formando uma única rede. O *encoder* recebe em sua entrada o banco de dados, e durante o treinamento servirá para extrair o máximo de informação possível dessas imagens, formando um mapa de características [2]. Após essa extração de informações, é preciso realizar a decodificação para analisar o mapa de características criado pelo *encoder* e, dessa forma, reconstruir a imagem a partir do *decoder*, formando assim o *autoencoder* [3].

## III. DESENVOLVIMENTO DO MODELO PROPOSTO

Após entender o funcionamento de um *autoencoder*, é possível adaptar sua arquitetura para ao invés de apenas reconstruir a imagem de entrada, aprimorá-la e retorná-la sem os ruídos que o banco de dados possuía. Para isso, ao invés de usar a entrada e saída do *autoencoder* para calcular a função da perda (i.e., *loss*), pode-se calcular a perda através de uma imagem com alta resolução e a imagem reconstruída pelo modelo.

### A. Construção do Autoencoder

Para a construção de uma rede com uma boa performance, acurácia e velocidade de processamento foi utilizada como arquitetura a U-Net, que por sua vez é uma arquitetura desenvolvida para segmentação de imagens, e foi criada para aplicações médicas [4].

Realizando algumas modificações na arquitetura U-net, como o número de camadas, banco de dados e diferentes funções de perda como, *MSE* e *MAE*, é possível obter-se um *autoencoder* cuja função é o tratamento de imagens com baixa resolução. A rede possui diversos parâmetros que podem ou não ter seus valores modificados durante o treinamento para melhorar o desempenho da rede, ou seja, pesos treináveis e não treináveis. Desse modo, a rede apresentou 75 milhões de pesos totais, onde todos são treináveis.

### B. Treinamento do modelo

Durante o treinamento, foi utilizado uma amostra de imagens retiradas do *MARVEL dataset* [5]. Ao todo 20 mil imagens foram utilizadas, dentre elas 10 mil imagens para a o treinamento da rede e outras 10 mil imagens para testes de acuraria. Todas as imagens foram agrupadas em pares, ou seja, a imagem com alta resolução era agrupada com uma cópia da mesma imagem, mas com baixa resolução, pois o modelo depende do par para ajustar os pesos da rede e assim calcular a sua perda.

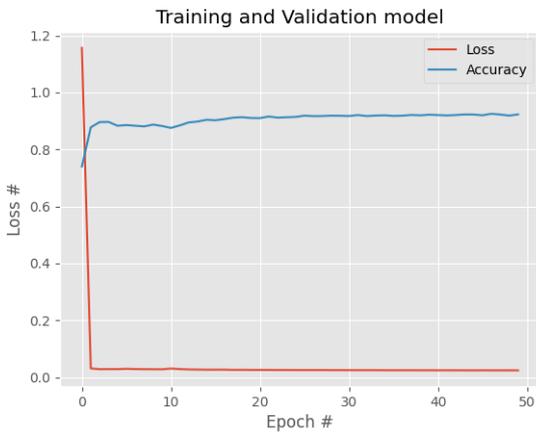


Fig. 1. Gráfico representando *loss* e *accuracy* durante o treinamento em *epochs*.



Fig. 2. Comparação entre imagem de entrada e imagem reconstruída pela rede

Durante o treinamento deve-se informar a quantidade de épocas utilizadas para o treinamento do modelo, esse parâmetro é de suma importância, visto que caso o modelo seja treinado com poucas épocas, ele pode não alcançar sua acurácia ótima, e caso treine por muitas épocas, pode ocorrer *overffiting*, ou seja, o modelo memoriza os resultados esperados do conjunto de treinamento, porém durante sua validação, ele apresenta uma alta quantidade de erros, indicando *overffiting*. Para o treinamento, foram utilizadas 50 épocas, onde ao final do mesmo, é gerado um gráfico apresentando o desempenho da rede, comparando a acuraria do modelo ao gerar imagens de alta resolução e sua perda durante todo o processo de treinamento, como mostrado na Figura 1.

#### IV. EXPERIMENTOS E ANALISE DOS RESULTADOS

Para realizar a validação da rede apresentada, foram realizados alguns experimentos de validação com o intuito de verificar a melhoria na resolução de imagens nunca vistas pela rede. Além desse experimento, a rede foi usada para verificar se ocorre algum impacto no desempenho de redes usadas para a detecção de embarcações.

Como exemplo do funcionamento do *autoencoder*, usa-se uma amostra a esquerda mostrada na Figura 2 como entrada e observa-se a saída da rede na amostra a direita, comparando os resultados. Conforme pode ser visto, a imagem é reconstruída com uma maior resolução.

Mesmo apresentando um resultado satisfatório na reconstrução, a rede foi usada para verificar qual o impacto gerado

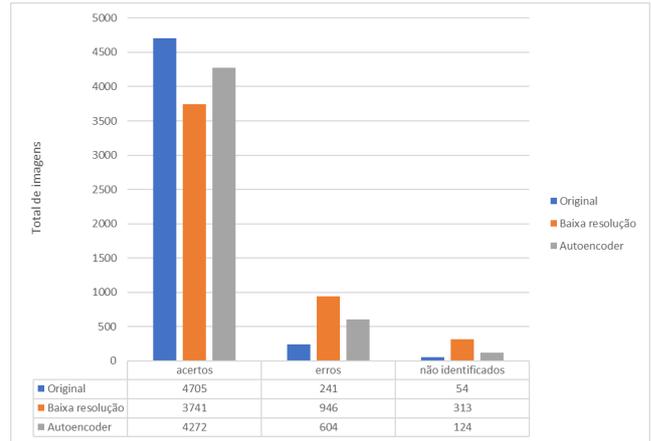


Fig. 3. Tabela representando os resultados da rede *autoencoder* aplicada a detecção de embarcações

na detecção de embarcações. Desse modo, um modelo de detecção foi usado em três testes diferentes, podendo ser exemplificado na 3. O primeiro teste, a rede é aplicada sobre as imagens originais. No segundo, são analisadas as imagens com resolução diminuída pela técnica *bicubic interpolation* e passadas pela rede de detecção. E a terceira, onde as imagens de baixa resolução são recuperadas por meio do modelo de super resolução e testadas na rede de detecção.

#### V. CONCLUSÃO

Com o desenvolvimento de novas técnicas para o pré-processamento de dados no campo de visão computacional, a rede desenvolvida neste trabalho agrega valor para situações onde não há um ambiente que favoreça com a qualidade da imagem final, seja devido à resolução da câmera utilizada ou mesmo com o cenário que contribuiu para a queda da resolução. Com um pré-processamento de imagens sendo utilizado em grandes bancos de dados, pode-se aprimorar os resultados de um sistema como detectores de objetos, assim como ajudar na visualização de imagens com baixa resolução.

#### REFERÊNCIAS

- [1] Eugenio Vocaturo, Ester Zumpano e Pierangelo Veltri. “Image pre-processing in computer vision systems for melanoma detection”. Em: dez. de 2018, pp. 2117–2124.
- [2] Qinxue Meng, Daniel R. Catchpole, David B. Skillicorn e Paul J. Kennedy. “Relational Autoencoder for Feature Extraction”. Em: *CoRR* abs/1802.03145 (2018). arXiv: 1802.03145.
- [3] Dor Bank, Noam Koenigstein e Raja Giryes. “Autoencoders”. Em: *CoRR* abs/2003.05991 (2020). arXiv: 2003.05991.
- [4] Olaf Ronneberger, Philipp Fischer e Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. Em: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597.
- [5] Erhan Gundogdu, Berkan Solmaz, Veysel Yücesoy e Aykut Koç. “MARVEL: A Large-Scale Image Dataset for Maritime Vessels”. Em: *Computer Vision – ACCV 2016*. Springer International Publishing, 2017, pp. 165–180. DOI: 10.1007/978-3-319-54193-8\_11.