



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

AVALIAÇÃO DO EMPREGO DE TÉCNICAS DE APRENDIZAGEM DE MÁQUINA EM
RESTAURAÇÃO DE IMAGENS SATELITAIS E AÉREAS

Daniel Fernandes Pereira

Relatório de Iniciação Científica
do programa PIBIC, orientada pelo
Dr. Haroldo Fraga de Campos Velho (INPE) e
Dra. Ana Paula Abrantes de Castro e Shiguemori (IFSP)

INPE
São José dos Campos
2022

RESUMO

Processamento Digital de imagem é um conjunto de técnicas computacionais voltadas para análise e tratamento de imagens adquiridas através de sensores. As técnicas computacionais tem como objetivo a manipulação de imagem, visando melhorar a qualidade da imagem ou a remoção de ruídos. O projeto tem como objetivo estudar o filtro neural multiescala para restauração de imagens, como continuidade e reprodução de uma tese de doutorado. Esse estudo visa a implementação do filtro neural multiescala em linguagem python, aplicados em imagens de satélite e aéreas.

Palavras-chave: Processamento Digital de imagem. Ruído. Filtro Neural Multiescala.

SUMÁRIO

1 INTRODUÇÃO	6
2.DESENVOLVIMENTO	8
Atividade 1 – Reunião com a orientadora do projeto	8
Atividade 2 – Capacitação do Bolsista	8
Atividade 3 – Desenvolvimento do código	8
3 RESULTADOS PARCIAIS	9
3.1 TESTES COM O USO DA BIBLIOTECA KERAS	9
3.2 IMPORTAÇÃO DOS DADOS DO PROJETO ORIGINAL	10
4 CONCLUSÕES	12
5 REFERÊNCIAS BIBLIOGRÁFICAS	13

1 INTRODUÇÃO

Nos últimos anos, o uso da inteligência artificial vem se mostrando uma ferramenta eficaz em várias aplicações. Uma das metodologias mais utilizadas da área de inteligência artificial são redes neurais de aprendizado profundo, capazes de modelar dados com uso de várias camadas de neurônios artificiais, com mapeamentos lineares e não lineares.

O projeto tem como foco principal estudar o filtro neural multiescala para restauração de imagens. Inicialmente pretende-se reproduzir resultados da tese de doutorado da profa. Ana Paula Abrantes de Castro e Shiguemori – orientadora desta pesquisa de IC. Esse estudo visa a implementação do filtro neural multiescala em linguagem Python, com aplicações em imagens de satélite e obtidas em aeronaves.

O relatório descreve os avanços feitos no projeto até o momento, explicando o processo de estudo do filtro neural multiescala e mostrando progressos na criação do código principal.

2.DESENVOLVIMENTO

A seguir são apresentadas as atividades realizadas até o momento:

1. Estudo do ambiente de desenvolvimento *Colab*;
2. Estudo da linguagem de programação *Python*;
3. Estudo dos conceitos de rede neural de aprendizado profundo;
4. Estudo das bibliotecas *Keras* e *TensorFlow* e suas aplicações no ambiente de desenvolvimento *Colab*;
5. Análise de modelos de código que exploram as bibliotecas de rede neural(*keras* e *TensorFlow*) com objetivo de manipular dados de imagens;
6. Uso da plataforma *MatLab online* para visualização e avaliação de dados de arquivos com extensão “.mat” .;
7. Importação dos dados utilizados na tese de doutorado da orientadora.

Atividade 1 – Reunião com a orientadora do projeto

As reuniões são realizadas pessoalmente, com a participação da orientadora. Além das reuniões, a orientadora mantém contato para acompanhamento da pesquisa através de e-mail e de aplicativo de mensagens virtuais.

Atividade 2 – Capacitação do Bolsista

Durante a realização do projeto foram assistidas videoaulas sobre Redes neurais, o ambiente *Colab*, a linguagem *Python* e suas bibliotecas *Keras* e *Tensorflow*

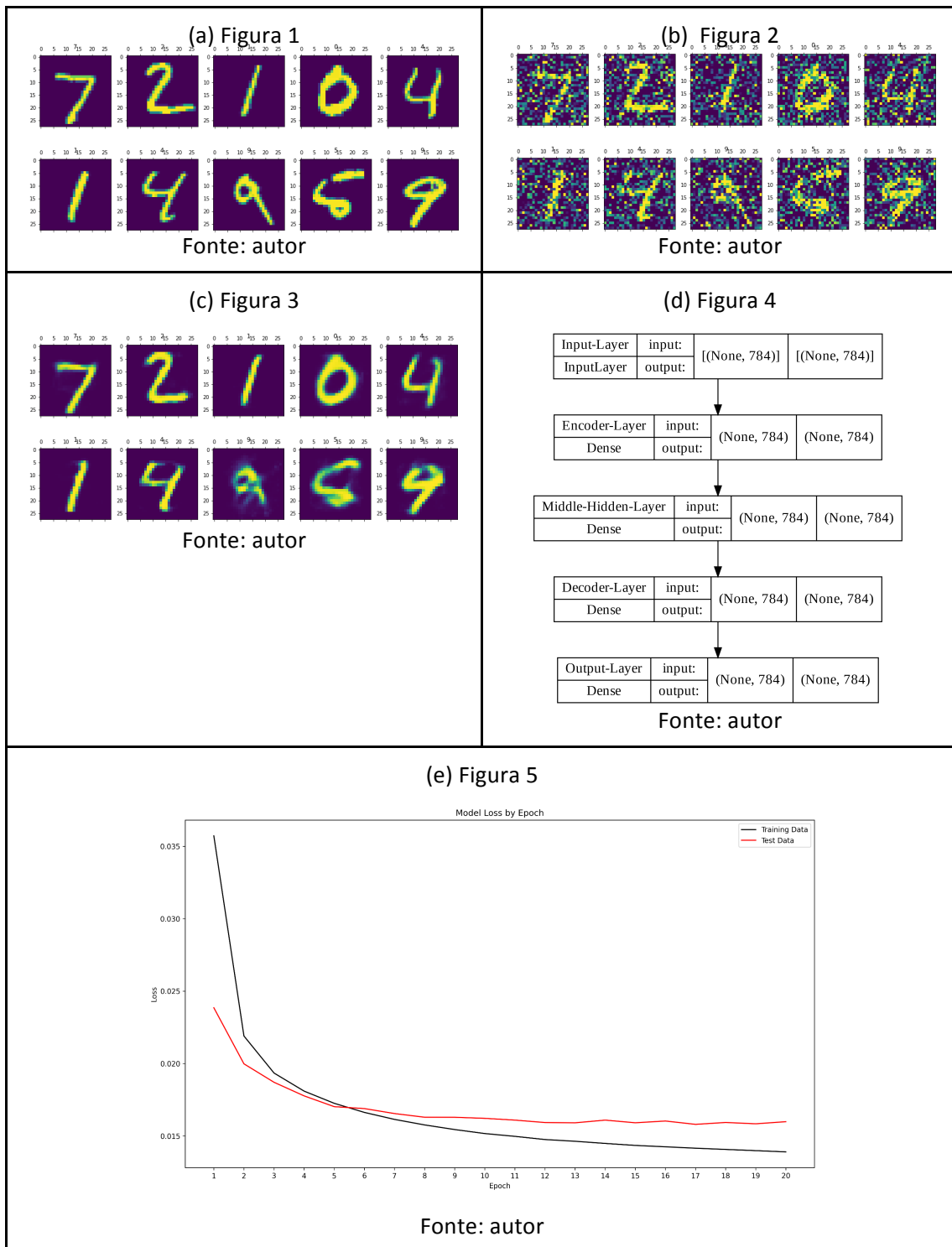
Atividade 3 – Desenvolvimento do código em linguagem Python

Foram feitos desenvolvimentos e testes e nas etapas do programa principal.

3 RESULTADOS PARCIAIS

3.1 TESTES COM O USO DA BIBLIOTECA KERAS

<https://biblio.direito.ufmg.br/?p=5082> (VER ESSE LINK PARA SABER COMO CITAR IMAGENS)



Nas Figuras 1-5 acima, foi realizado um teste com a imagem mostrada na Fig. 1, que representa a imagem original. A Figura 2 mostra a mesma imagem, mas contaminada com ruído branco gaussiano. Uma rede neural de aprendizado profundo foi aplicada para restaurar a imagem degradada.

A imagem restaurada com aplicação da rede de aprendizado profundo é mostrada na Figura 3. Nota-se que os números 9 e 5 não foram restaurados de forma adequada. Entretanto, a imagem original referente a estes números são as que apresentam uma menor similaridade com a forma de referência mais comum destes números. Embora os seres humanos possam inferir o significado das imagens, o padrão da imagem original é distinto da imagem mais regular da representação destes números. Desta forma, mesmo removendo parcialmente a degradação, a reconstrução perde um pouco do significado.

A Figura 4 mostra os passos da metodologia:

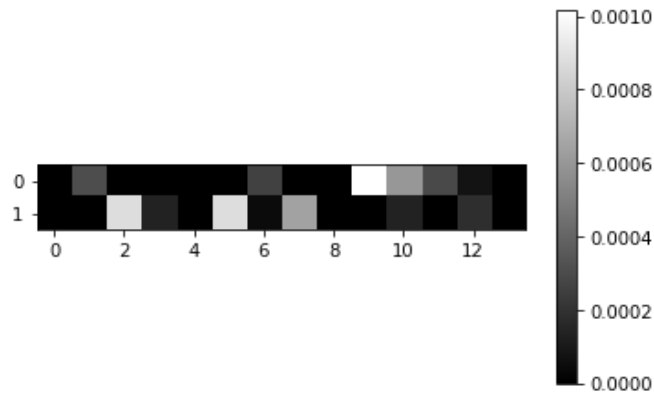
- a) Dados importados do conjunto de dados da biblioteca *Keras* sem a adição de ruído
- b) Dados importados do conjunto de dados da biblioteca *Keras* com a adição de ruído
- c) Resultado após aplicação do modelo de rede neural
- d) Modelo de rede neural utilizado
- e) Gráfico de perda de semelhança por época

3.2 IMPORTAÇÃO DOS DADOS DO PROJETO ORIGINAL

O objetivo da tese de doutorado da profa. Castro e Shiguemori é configurar/treinar uma rede neural para restauração de dados a partir de uma única imagem com vários tons de cinza. Assim, tem-se uma imagem de referência (sem contaminação de ruído) e outra imagem com ruídos.

As imagens são segmentadas, onde cada segmento representa uma amostra da imagem ruidosa e de referência. Os segmentos são um conjunto de pixels da imagem, que podem conter diferentes números de pixels, denotadas com diferentes níveis – ou escalas – da imagem segmentada.

A Figura 6 abaixo mostra um exemplo de imagem segmentada. Aqui, somente imagens em tons de cinza serão processadas. A Figura 7 é uma amostra de uma tela de trabalho com execução do código Python via CoLab-Google com acesso à plataforma *Keras* de implementação de rede de aprendizado profundo.



Fonte: autor

Figura 6: Gráfico de alguns dados de entrada presentes no arquivo com extensão “.mat”.

```

{'_header_': b'MATLAB 5.0 MAT-file, Platform: PCWIN, Created on: Fri May 11 17:31:54 2007',
 '_version_': '1.0',
 'globals_': [],
 'ans': array([[ 1, 28]], dtype=uint8),
 'w': array([[1.98169237e-03, 5.69506649e-02, 1.82906940e-02, ...,
 8.70217169e-02, 3.07622215e-02, 7.13329178e-15],
 [2.47852833e-02, 1.04869449e-01, 4.08739505e-02, ...,
 1.47122097e-01, 8.93006070e-02, 3.04110522e-28],
 [5.27577424e-04, 4.71032041e-03, 3.29181530e-03, ...,
 4.60380620e-03, 1.32264978e-01, 3.16274940e-26],
 ...,
 [0.00000000e+00, 9.01960784e-02, 6.27450980e-02, ...,
 0.00000000e+00, 2.78431373e-01, 0.00000000e+00],
 [2.94117647e-01, 0.00000000e+00, 9.41176471e-02, ...,
 0.00000000e+00, 2.74509804e-01, 0.00000000e+00],
 [8.62745098e-02, 3.52941176e-02, 6.27450980e-02, ...,
 8.62745098e-02, 0.00000000e+00, 0.00000000e+00]]),
 'ro': array([[0.85]]),
 'n': array([[28]], dtype=uint8),
 'nw': array([[33040]], dtype=uint16),
 'x': array([[0.          , 0.07843137, 0.          , ..., 0.          , 0.17254902,
 0.10588235],
 [0.07843137, 0.          , 0.          , ..., 0.17254902, 0.10588235,
 0.          ],
 [0.          , 0.          , 0.          , ..., 0.10588235, 0.          ,
 0.03137255],
 ...,
 [0.06666667, 0.04313725, 0.09803922, ..., 0.          , 0.03921569,
 0.          ],
 [0.09803922, 0.15686275, 0.08235294, ..., 0.          , 0.          ,
 0.16862745],
 [0.          , 0.          , 0.          , ..., 0.          , 0.          ,
 0.          ]]),
 'Nepmax': array([[1]], dtype=uint8),
 'epoca': array([[1]], dtype=uint8),
 'flag': array([[1]], dtype=uint8),
 'eta': array([[0.5]]),
 'i': array([[256036]], dtype=int32),
 'flag1': array([[1]], dtype=uint8),
 'maior': array([[0.90396885]]),
 'v': array([[0.79729059],
 [0.7707797],
 [0.8655332 ],
 ...,
 [0.76892605],
 [0.68251132],
 [0.67850312]]),
 'jg': array([[10194]], dtype=uint16),
 'numclusters': array([[33040]], dtype=uint16)}

```

Fonte: autor

Figura 6: Os dados foram importados como arquivo “.mat”, criando uma variável do tipo dict.

4 CONCLUSÕES

Até o momento, foi possível estudar e desenvolver algumas etapas do projeto de forma fluída, gerando resultados concretos. Porém, houve algumas dificuldades ao identificar certos dados que faziam parte do projeto matriz por serem muito antigos, resultando no atraso do processo de criação do código principal, que necessita desses dados para satisfazer as propostas da pesquisa.

5 REFERÊNCIAS BIBLIOGRÁFICAS

DEEP LEARNING: <https://www.deeplearningbook.com.br/o-que-sao-redes-neurais-artificiais-profundas/>

COLAB, Colaboratory, 2019. Disponível em: <https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=P-H6Lw1vyNNd>

KERAS. Disponível em: <https://keras.io/> (Acesso em: 03 agosto de 2022).

SCIKIT- LEARN. Disponível em: <https://scikit-learn.org/stable/> (Acesso em: 03 agosto de 2022).

TENSORFLOW Core. Disponível em: <https://www.tensorflow.org/tutorials> (Acesso em: 03 agosto de 2022).

MATLAB ONLINE. Disponível em: <https://www.mathworks.com/products/matlab-online.html> (Acesso em: 03 agosto de 2022).

ABRANTES DE CASTRO, Ana Paula. Restauração de imagens com operadores modelados por redes neurais artificiais. 2009. Tese. Programa de Pós-Graduação do INPE em Computação Aplicada. São José dos Campos. 2009

https://bdtd.ibict.br/vufind/Record/INPE_9f0f80c8028c9466d2abdda8f0ccf3f2

<http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18@80/2009/02.16.18.02/doc/publicacao.pdf>