



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

O processamento digital de imagens é uma área de estudo que utiliza técnicas computacionais para análise e tratamento de imagens obtidas por meio de diversos tipos de câmeras: sistema óptico de lentes, sensores CCD (Charge Coupled Device), CMOS (Complementary Metal Oxide Semiconductor). Independentemente do dispositivo utilizado, as imagens obtidas sempre apresentam algum tipo de ruído. Neste projeto de iniciação científica, é proposto a implementação e avaliação de um filtro neural multiescala de aprendizado profundo para a restauração de imagens, um tema fundamental para várias áreas, incluindo pesquisa em astronomia e astrofísica. Ao contrário das abordagens que utilizam técnicas de aprendizado de máquina para restauração de imagens, o treinamento do algoritmo de multiescala é realizado a partir de uma única imagem (CASTRO et al., 2008). A plataforma Keras (KERAS, 2008) será utilizada para o desenvolvimento da rede neural profunda neste projeto. Uma avaliação inicial é realizada com a imagem “Lenna”, que é uma referência típica em avaliação de processamento de imagens. Em seguida, são realizados experimentos com imagens astronômicas. As métricas de avaliação de desempenho da restauração são as mesmas utilizadas por Castro et al. (2008).

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

SUMÁRIO

1 INTRODUÇÃO	1
2.DESENVOLVIMENTO	2
3 RESULTADOS	3
3.1 TESTES E RESULTADOS DO USO DA BIBLIOTECA KERAS	3
3.2 IMPORTAÇÃO DOS DADOS DO PROJETO ORIGINAL	4
3.3 REALIZAÇÃO DE TESTES COM OS DADOS IMPORTADOS	6
3.4 ATIVAÇÃO DA REDE NA IMAGEM LENNA DEGRADADA	7
3.5 APERFEIÇOAMENTO DO MODELO DE REDE NEURAL	10
3.6 COMPARAÇÃO DE MÉTRICAS	11
3.7 ATIVAÇÃO EM NOVAS IMAGENS	11
4 CONCLUSÕES	15
5 REFERÊNCIAS BIBLIOGRÁFICAS	16

LISTA DE FIGURAS

Figura 1: Dados sem ruído	3
Figura 2: Dados com ruído	3
Figura 3: Dados após o treino	3
Figura 4: Modelo de rede neural	3
Figura 5: Gráfico de perda de semelhança	3
Figura 6: Dados do arquivo .mat	5
Figura 7: Plotagem da entrada	6
Figura 8: Entrada com ruído e saída desejada	6
Figura 9: Ativação da rede treinada no círculo degradado	7
Figura 10: Entrada degradada e saída desejada (<i>Lenna</i>)	8
Figura 11: Ativação da rede treinada na imagem <i>Lenna</i>	8
Figura 12: Atualização do modelo de rede neural	10
Figura 13: Ativação na imagem <i>Galáxia</i>	12
Figura 14: Ativação na imagem <i>Saturno</i>	13

1 INTRODUÇÃO

Nos últimos anos, o uso da inteligência artificial vem se mostrando um meio muito eficaz na resolução de problemas. Uma das formas de se utilizar inteligência artificial é criando uma rede neural de aprendizado profundo, que é capaz de modelar abstrações de alto nível de dados usando um grafo profundo com várias camadas de processamento, compostas de várias transformações lineares e não lineares.

O projeto tem como foco principal estudar o filtro neural multiescala para restauração de imagens, continuando e reproduzindo a tese de Castro et al. (2008). Esse estudo visa a implementação do filtro neural multiescala em linguagem python, aplicado em imagens com ruído.

O relatório tem como alvo expor 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:

Estudo do ambiente de desenvolvimento *Colab*

Estudo da linguagem de programação *Python*

Estudo dos conceitos de Rede neural e Aprendizado profundo

Estudo das bibliotecas *Keras* e *TensorFlow* e suas aplicações no ambiente de desenvolvimento *Colab*

Análise de modelos de código que exploram as bibliotecas de rede neural(*keras* e *TensorFlow*) com objetivo de manipular dados de imagens

Uso da plataforma *MatLab online* para visualização e confirmação de dados de arquivos .mat

Importação dos dados utilizados na tese de doutorado da orientadora

Realização de testes, treinando a rede com os dados importados

Ativação da rede na imagem Lenna degradada

Aperfeiçoamento do modelo da rede neural

Realização das métricas / Comparação

Ativação em novas imagens degradadas

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

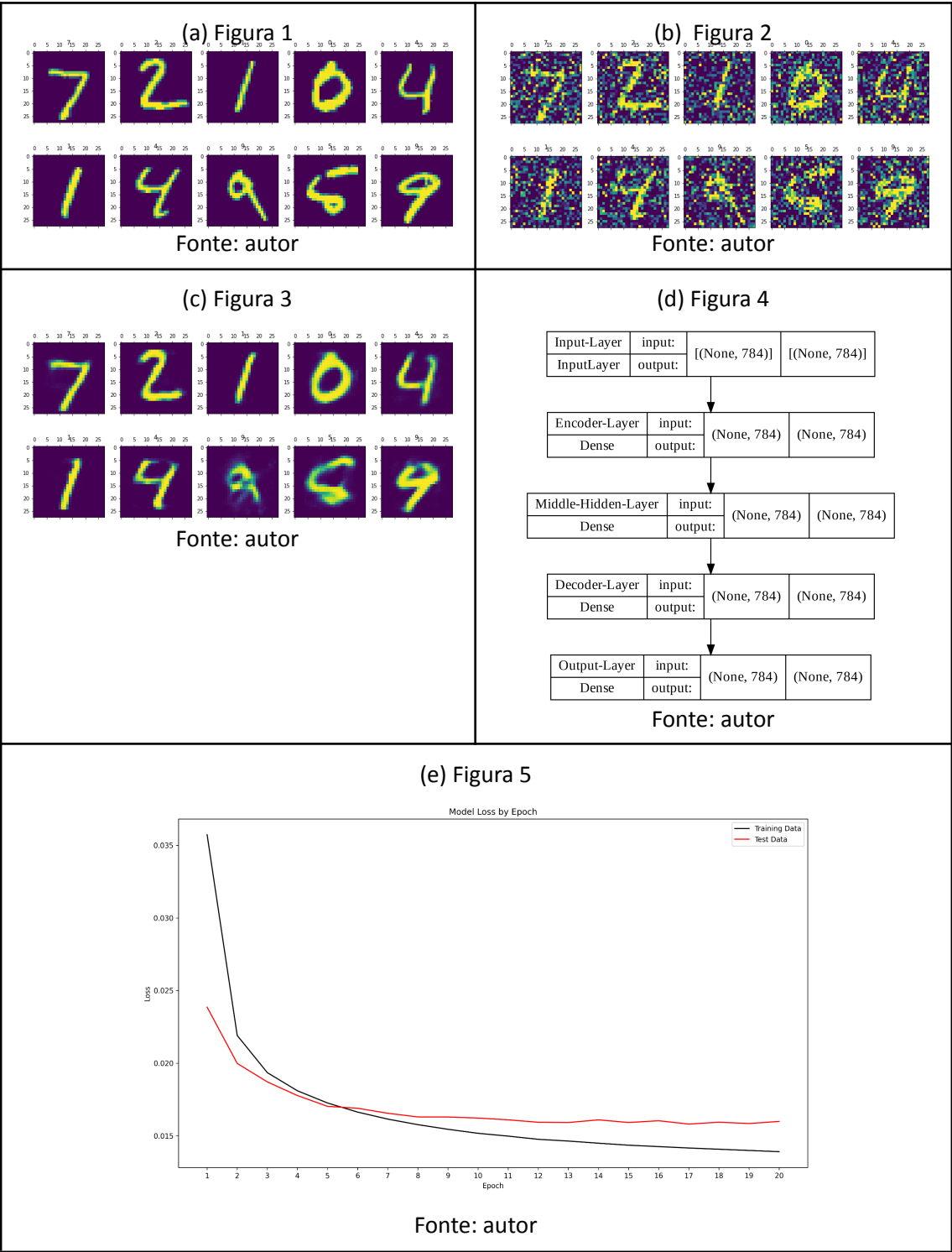
Durante a realização do projeto foram assistidas videoaulas sobre Redes neurais, o ambiente *Colab*, a linguagem *Python* e suas bibliotecas *Keras* e *Tensorflow*

Atividade 3 – Desenvolvimento do código

Foram feitos testes e desenvolvimentos nas etapas do programa principal.

3 RESULTADOS PARCIAIS

3.1 TESTES COM O USO DA BIBLIOTECA KERAS



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 desses números. Embora os seres humanos possam inferir o significado das imagens, o padrão da imagem original é distinto da imagem mais regular da representação desses 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 de Castro et al. (2008) 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.

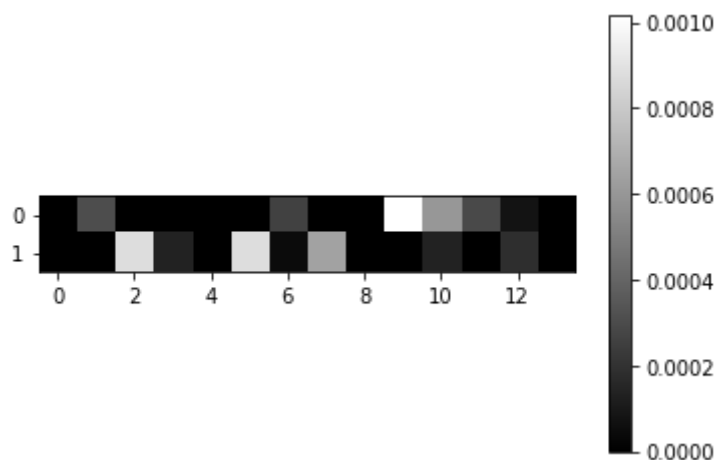
Figura 6

```
{'_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],
            [0.77707797],
            [0.8655332 ],
            ...,
            [0.76892605],
            [0.68251132],
            [0.67850312]]),
 'jg': array([[10194]], dtype=uint16),
 'numclusters': array([[33040]], dtype=uint16)}
```

Fonte: autor

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

Figura 7



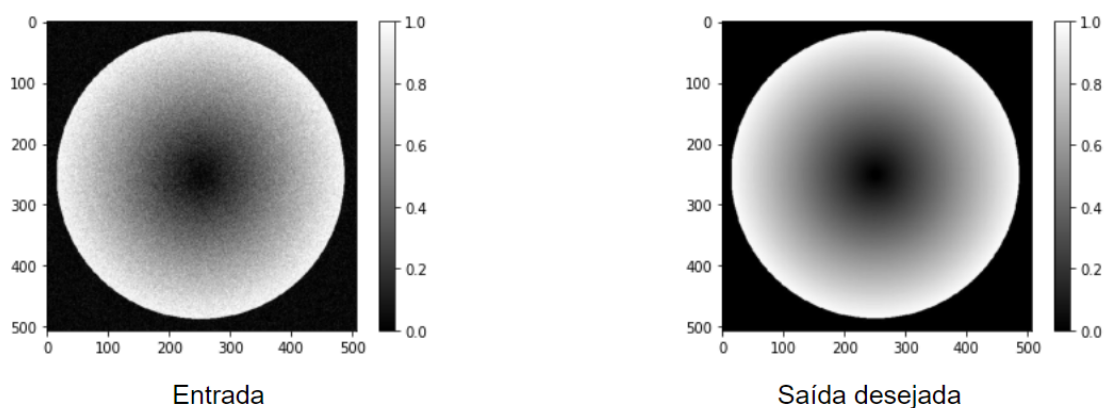
Fonte: autor

Plotagem de alguns dados de entrada presentes no arquivo *.mat*

3.3 REALIZAÇÃO DE TESTES COM OS DADOS IMPORTADOS

Com um modelo base montado, foram designados os mesmos números de camadas e neurônios, funções de ativação e taxa de aprendizagem que estavam na tese de doutorado de Castro et al. (2008)

Figura 8

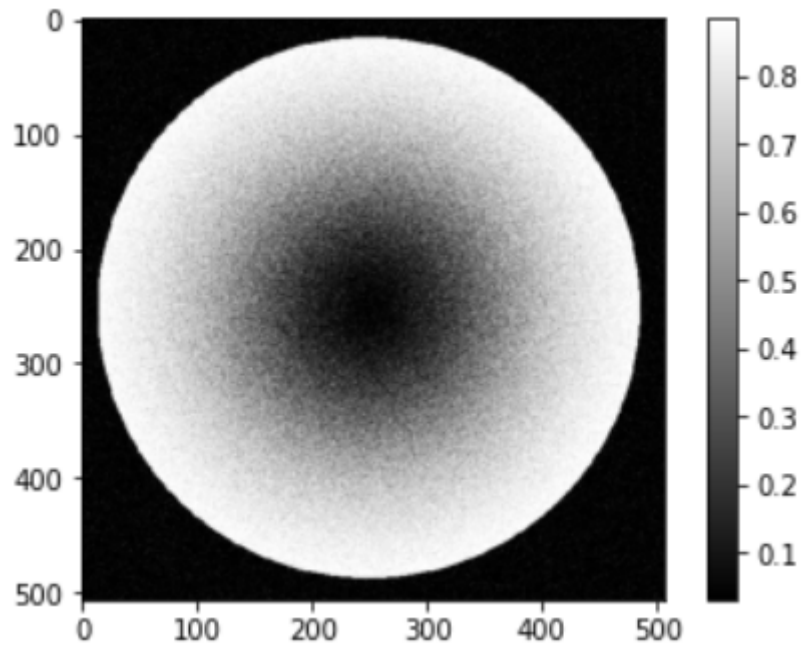


Fonte: autor

Na figura 8 acima, é apresentada a plotagem dos dados importados anteriormente que serão utilizados no treino da rede neural

Figura 9

RESULTADOS



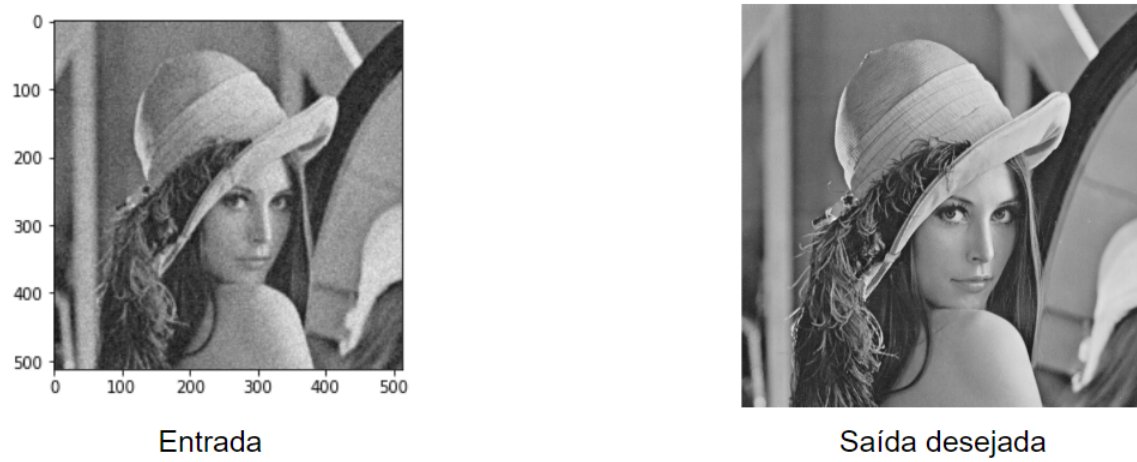
Fonte: autor

A figura 9 acima é um de muitos resultados de testes realizados com a rede neural

3.4 ATIVAÇÃO DA REDE NA IMAGEM LENNA DEGRADADA

Feitos os testes com o círculo degradado, o próximo passo foi realizar a ativação da rede treinada em uma imagem degradada diferente

Figura 10

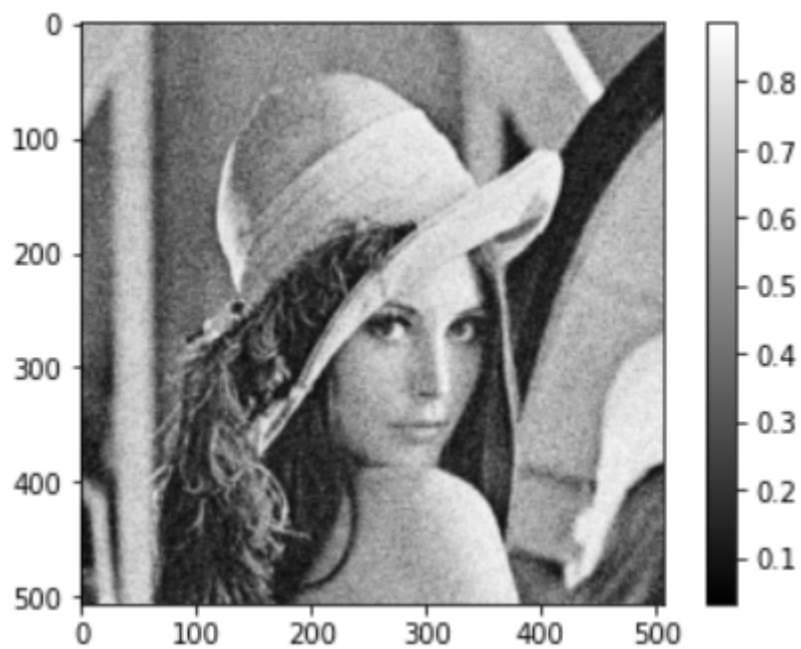


Fonte: autor

A figura 10 apresenta a plotagem da imagem degradada *Lenna* que foi reconstruída na figura 11 por meio da ativação da rede

Figura 11

RESULTADOS

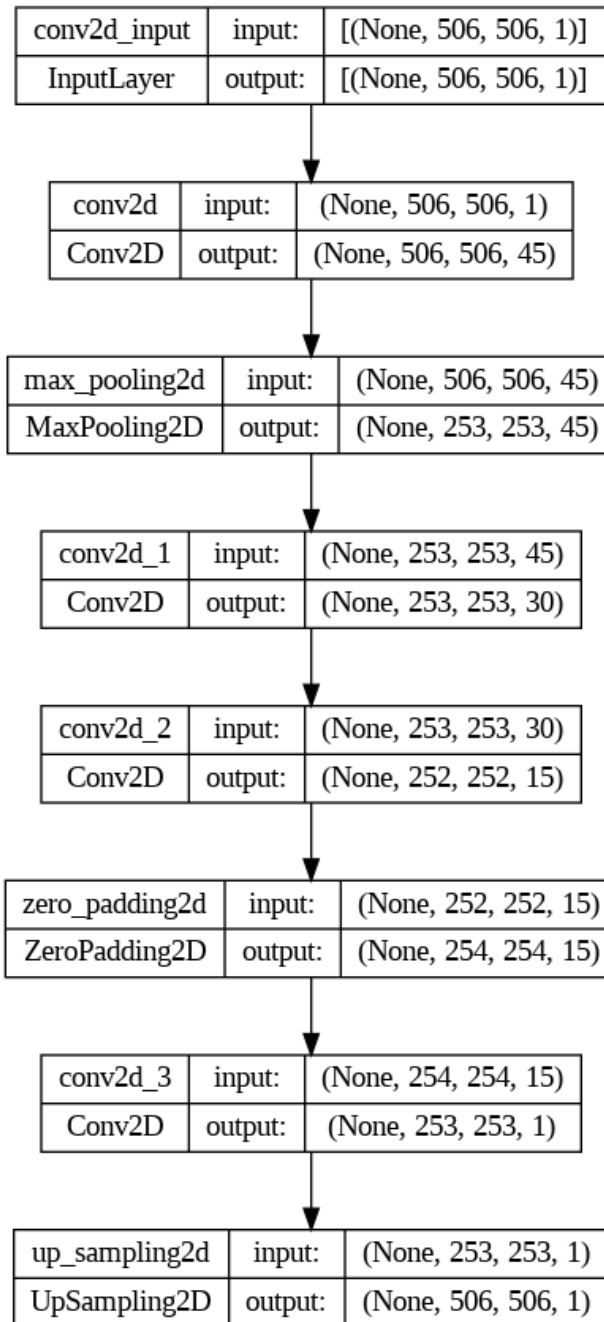


Fonte: autor

3.5 APERFEIÇOAMENTO DO MODELO DA REDE NEURAL

Uma das propostas iniciais deste projeto consistiu no aprimoramento do modelo de rede neural previamente desenvolvido. A partir da análise dos resultados e da identificação de possíveis limitações do modelo original, decidiu-se buscar melhorias que permitissem otimizar o desempenho e a eficácia do sistema.

Figura 12



Fonte: autor

A figura 12 mostra uma plotagem do novo modelo. O modelo é uma arquitetura de rede neural convolucional (CNN) composta por várias camadas para processamento de imagens. A entrada do modelo são imagens de tamanho 506x506 com um único canal (imagem em escala de cinza).

A primeira camada é uma camada de convolução com 45 filtros de tamanho 2x2. A função de ativação utilizada é a ReLU (Unidade Linear Retificada), que ajuda a introduzir não-linearidade na rede.

Em seguida, temos uma camada de MaxPooling2D, que reduz a resolução espacial da saída da camada de convolução através do uso de uma janela de pooling de tamanho 2x2.

A próxima camada é outra camada de convolução com 30 filtros de tamanho 2x2, também utilizando a função de ativação ReLU e com preenchimento (padding) 'same', que mantém a mesma dimensão espacial da entrada.

Após isso, temos outra camada de convolução com 15 filtros de tamanho 2x2 e função de ativação ReLU.

A camada ZeroPadding2D é usada para adicionar uma margem de preenchimento de 1 pixel em cada lado da imagem.

A seguir, temos mais uma camada de convolução com 1 filtro de tamanho 2x2 e função de ativação sigmoid, que é utilizada para obter a saída final da rede.

Por fim, temos uma camada de UpSampling2D, que aumenta a resolução espacial da saída da camada de convolução anterior através de um fator de escala de 2x2.

3.6 Comparação de métricas

Após a aplicação dos modelos de restauração de imagens, foram realizadas comparações detalhadas das métricas de qualidade de imagem para avaliar o desempenho e a eficácia de cada abordagem. As métricas utilizadas nas comparações foram: média, variância, desvio padrão, a raiz do erro médio quadrático (RMSE) e a relação sinal-ruído (SNR).

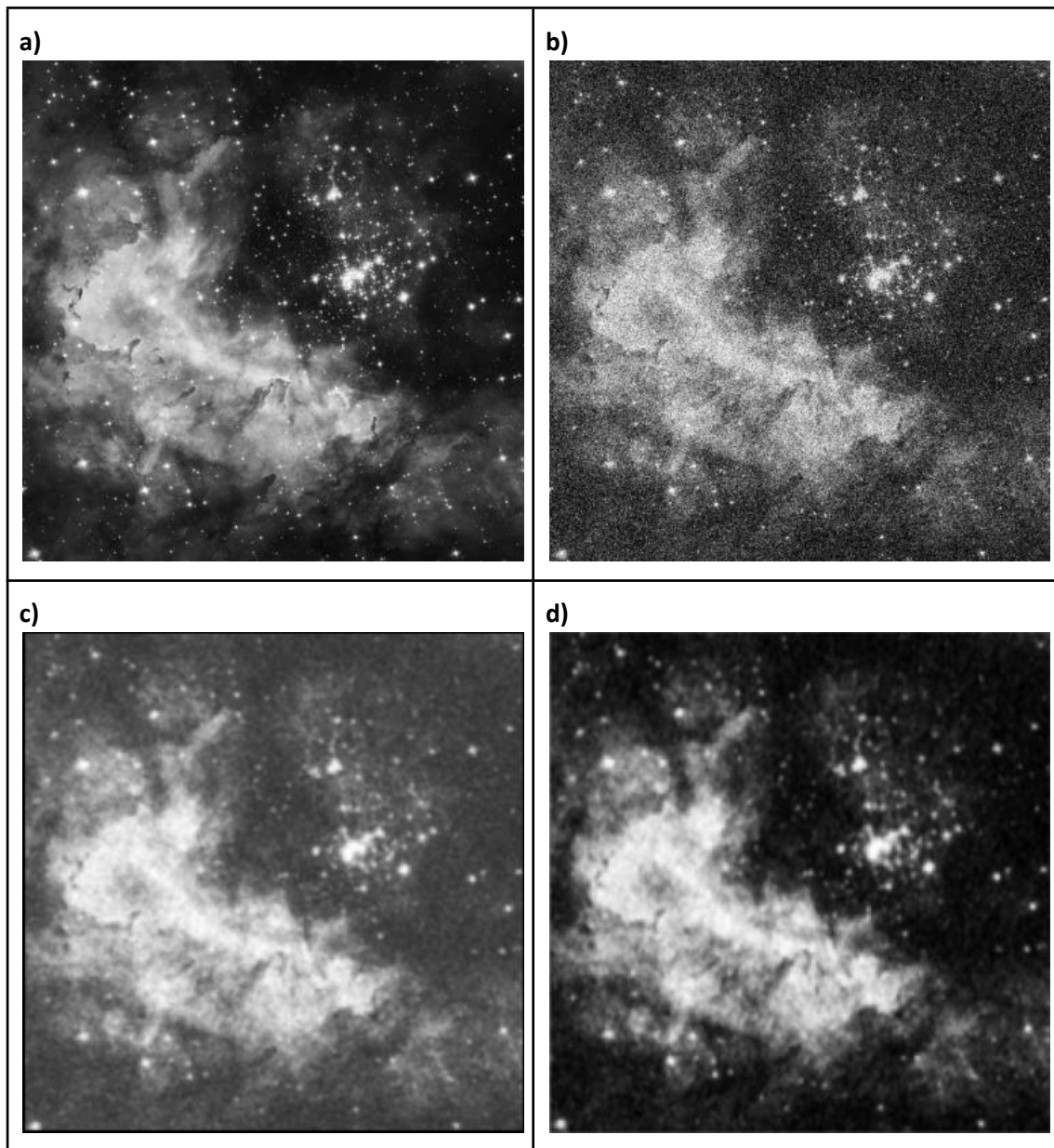
Porém, foi identificado que o conjunto de dados que estava sendo utilizado se tratava do conjunto bruto(256.036), diferente do reduzido(11.318 e 2.886) impossibilitando que as comparações se tornassem válidas com as métricas de Castro et al. (2008).

3.7 Ativação em novas imagens

Neste capítulo, exploramos a aplicação dos dois modelos de aprendizagem profunda, um modelo aproximado ao anterior e uma versão aprimorada, para restaurar imagens de uma galáxia e do planeta Saturno.

Figura 13

Galáxia



Fonte: autor

a) Imagem sem degradação.

b) Imagem degradada.

c) Imagem restaurada (aproximação).

d) Imagem restaurada (Novo modelo)

A tabela acima apresenta as versões da imagem “*Galáxia*”, que apresentam as seguintes métricas:

Imagem	μ_f	ν_f	σ_f	$\sqrt{E}$	SNR
Original	79.2991	2775.1797	52.6799	—	—
Degradada	84.7657	3424.0309	58.5152	—	—
Modelo aproximado	76.5332	2364.2234	48.6232	3.2895	23.2883
Novo modelo	78.91047	2525.4016	50.2533	3.3145	23.1516

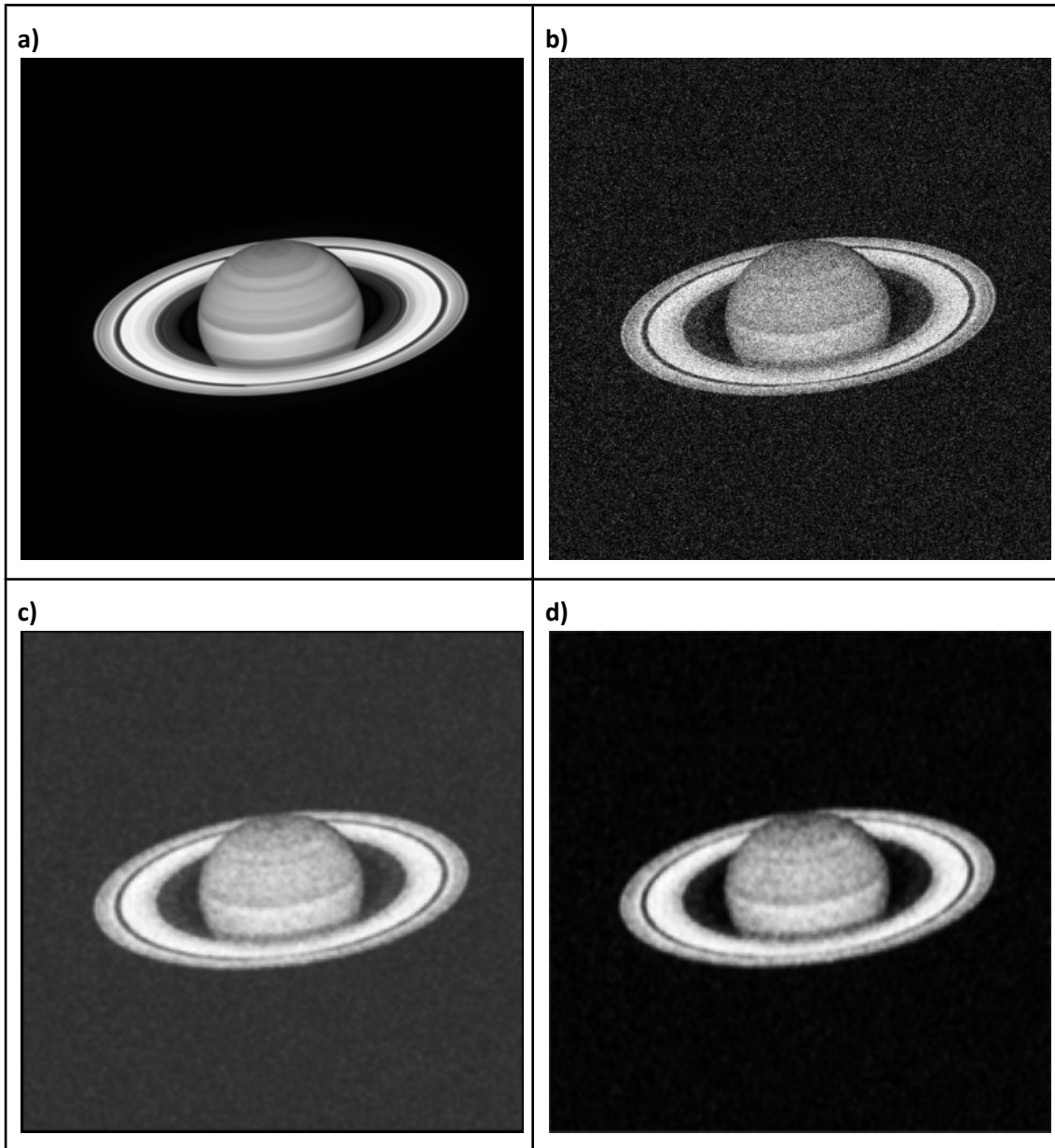
Onde:

- μ_f - Média
- ν_f - Variância
- σ_f - Desvio padrão
- $\sqrt{E}$ - Raiz quadrada do erro médio quadrático
- SNR - Relação sinal-ruído

Seguindo o segmento acima, foram feitos testes com a imagem “*Saturno*”:

Figura 14

Saturno



Fonte: autor

a) Imagem sem degradação.

b) Imagem degradada.

c) Imagem restaurada (aproximação).

d) Imagem restaurada (Novo modelo).

Imagem	μf	νf	σf	$\sqrt{E}$	SNR
Original	26.1807	3618.8721	60.1570	—	—
Degradada	38.4833	3247.3011	56.9850	—	—
Modelo aproximado	44.3932	2546.1362	50.4592	4.5998	20.9840
Novo modelo	45.6588	2676.925	51.7390	4.7674	20.1014

4 CONCLUSÕES

Até o momento, tivemos êxito em estudar e desenvolver várias etapas do projeto de forma fluída, culminando na submissão de um [artigo](#) ao SIBGRAPI. No entanto, foram encontradas dificuldades ao identificar dados antigos que integravam a matriz do projeto, o que nos levou a recorrer a um conjunto de dados alternativo. Essa alteração impacta a precisão da reprodução do modelo de rede neural base, restringindo, assim, alguns dos objetivos iniciais de comparação.

5 REFERÊNCIAS BIBLIOGRÁFICAS

DEEPLARNING. <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-H6Lw1vyND>>

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>