Universidade Paulista - UNIP

Lucas Ramos de Oliveira

PROCESSAMENTO DE IMAGEM E ANÁLISE DE DADOS APLICADOS À ASTRONOMIA

Limeira

2023

Universidade Paulista - UNIP

Lucas Ramos de Oliveira

PROCESSAMENTO DE IMAGEM E ANÁLISE DE DADOS APLICADOS À ASTRONOMIA

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em ciência da computação sob a orientação do professor Dr. Danilo Rodrigues Pereira, PhD.

Limeira

2023

Lucas Ramos de Oliveira

PROCESSAMENTO DE IMAGEM E ANÁLISE DE DADOS APLICADOS À ASTRONOMIA

Trabalho de conclusão de curso apresentado à banca examinadora da Faculdade UNIP, como requisito parcial à obtenção do Bacharelado em Ciência da Computação sob a orientação do professor Dr. Danilo Rodrigues Pereira, PhD

Aprovada em XX de XXXXX de 201X.

BANCA EXAMINADORA

_____

Prof. Dr. Nome completo

_____

Prof. Me. Nome completo

_____

Prof. Esp. Nome completo

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus por me permitir chegar até essa etapa do meu curso, a minha família por sempre acreditar e apoiar os meus sonhos e aos meus colegas de curso, pois o auxílio e apoio deles foi fundamental para meu desenvolvimento e crescimento como universitário.

“Se você quiser descobrir os segredos do Universo, pense em termos de energia, frequência e vibração”.

(Nikola Tesla)

RESUMO

Este trabalho aborda a temática do processamento de imagens no âmbito da astronomia, um campo que vem se destacando pela sua relevância no estudo do universo. Com a utilização de bibliotecas Python renomadas, como AstroPy e OpenCV(Open Source Computer Vision Library), serão apresentadas técnicas e metodologias que têm por finalidade aprimorar a qualidade das imagens astronômicas, tornando-as mais acessíveis e úteis para a pesquisa científica.

Dentre as principais abordagens, destacam-se a aplicação de ajustes de brilho, saturação, contraste e redução de ruído, com foco nas imagens no formato FITS, que são amplamente adotadas no âmbito da astronomia. Tais etapas de processamento desvendam detalhes previamente imperceptíveis, representando um avanço significativo para o campo da pesquisa científica.

O estudo se concentra em imagens capturadas pelo telescópio espacial James Webb da Nebulosa da Águia onde serão utilizados filtros específicos para a criação de imagens no esquema de cores RGB. Essas imagens processadas revelam informações anteriormente invisíveis, adicionando valor ao conjunto de dados astronômicos.

Além disso, o trabalho abordará a análise da similaridade entre as imagens processadas através da comparação de histogramas, empregando o método "cv2.HISTCMP_BHATTACHARYYA" do OpenCV.

Em resumo, o processamento de imagens desempenha um papel crucial na tradução de observações avançadas em imagens que proporcionam informações valiosas e compreensíveis. Este trabalho representa uma contribuição relevante para a exploração do universo, por meio do processamento de imagem, tornando-o mais acessível e esclarecedor.

Palavra-Chave: Processamento; Imagem; Dados

ABSTRACT

This work addresses the theme of image processing in the field of astronomy, a field that has been gaining prominence due to its relevance in the study of the universe. Using renowned Python libraries such as AstroPy and OpenCV (Open Source Computer Vision Library), techniques and methodologies will be presented with the aim of enhancing the quality of astronomical images, making them more accessible and useful for scientific research.

Among the main approaches, adjustments to brightness, saturation, contrast, and noise reduction will be highlighted, focusing on images in the FITS format, widely adopted in astronomy. These processing steps unveil details that were previously imperceptible, representing a significant advancement for scientific research.

The study focuses on images captured by the James Webb Space Telescope of the Eagle Nebula, where specific filters will be used to create images in the RGB color scheme. These processed images reveal previously invisible information, adding value to the astronomical dataset.

Additionally, the work will address the analysis of similarity between processed images through histogram comparison, employing the "cv2.HISTCMP_BHATTACHARYYA" method from OpenCV.

In summary, image processing plays a crucial role in translating advanced observations into images that provide valuable and understandable information. This work represents a relevant contribution to the exploration of the universe through image processing, making it more accessible and enlightening.

Keywords: Processing; Image; Data

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

FITS: Flexible Image Transport System - Sistema de Transporte de Imagens Flexível

JPL: Jet Propulsion Laboratory - Laboratório de Propulsão a Jato

JWST: James Webb Space Telescope - Telescópio Espacial James Webb

MAST: Mikulski Archive for Space Telescopes - Arquivo Mikulski para Telescópios Espaciais

MIRICam: Mid-Infrared Instrument - Instrumento de Infravermelho Médio

NASA: National Aeronautics and Space Administration - Administração Nacional da Aeronáutica e Espaço

NIRCam: Near Infrared Camera - Câmera de Infravermelho Próximo

OpenCV: Open Source Computer Vision Library - Biblioteca de Visão Computacional de Código Aberto

SUMÁRIO

1. INTRODUÇÃO:

A astronomia é uma área de pesquisa que vem crescendo bastante e tem despertado o interesse tanto por parte do público geral quanto por parte dos pesquisadores e cientistas. Esse crescimento acerca da astronomia se dá graças ao desenvolvimento de novas tecnologias cada vez mais avançadas e à curiosidade humana em relação ao universo[6].

Entende-se que a astronomia é uma ciência que busca compreender os objetos celestes, a sua evolução e propriedades. Nesse contexto, o processamento de imagem tem se tornado fundamental para análise das imagens e dos dados astronômicos obtidos.

Ao mesmo tempo que os aparelhos utilizados para estudar os astros vêm evoluindo, sendo alguns deles os telescópios, câmeras e lentes, muitos fatores podem influenciar no resultado dos dados obtidos. A iluminação, ruídos e angulação dos corpos celestes, por exemplo, são um dos desafios que podem ser enfrentados na tentativa de se obter uma bela astrofotografia.

Com o auxílio de ferramentas disponíveis é possível realizar o processamento das imagens a fim de se obter uma imagem de qualidade do qual poderá ser útil para que a comunidade científica possa ir mais a fundo nos estudos e entendimento do universo.

Para obtenção das imagens e dados de astrofotografias, é necessário o uso de equipamentos específicos para coleta de imagens, sendo um desses equipamentos os telescópios espaciais. Lançado em 24 de abril de 1990 o telescópio Hubble se mostrou bastante útil para comunidade científica, pois através dele foi possível obter imagens de objetos celestes jamais observados pelo homem, auxiliando no estudo acerca do nosso universo.

O telescópio Hubble é um telescópio que foi projetado para observações ópticas e ultravioletas. Porém a fim de ir mais longe, em Dezembro de 2021 [9] a Administração Nacional da Aeronáutica e Espaço (NASA) lançou um dos telescópios mais modernos da atualidade o James Webb Space Telescope(JWST) que é capaz de observar através do infravermelho podendo realizar observações de objetos à bilhões de anos luz de distância.

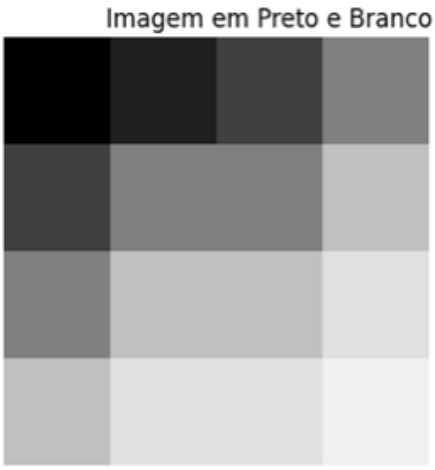
Neste trabalho será realizado o processamento de imagem com dados obtidos pelo JWST a fim de comparar os dados das imagens e conscientizar sobre a importância dos avanços tecnológicos para auxílio e contribuição da comunidade científica.

Imagens astronômicas assim como imagens digitais comuns são representadas como arrays de números do qual são chamados de matrizes ou tensores, dependendo da dimensão das imagens. Cada elemento da matriz vai representar um pixel da imagem do qual estará associado a um valor numérico que corresponde à intensidade de luz ou radiação naquela parte da imagem, ou seja, quanto maior for o valor do pixel, mais brilhante será o ponto correspondente da imagem.

Imagens preto e branco podem ser descritas por uma única matriz de números que definem o brilho do pixel(Figura 01). Já imagens coloridas é definida por 3 arrays de números que vai especificar a intensidade das cores usando os canais (vermelho, verde e azul) cada um representado por uma matriz separada. Cada pixel no canal RGB é representado por um valor numérico que indica a intensidade da cor correspondente e ao se juntarem vão formar a imagem colorida(Figura 02).

Figura: 01 Exemplo de uma imagem preto e branco representada por uma matriz 5x5.

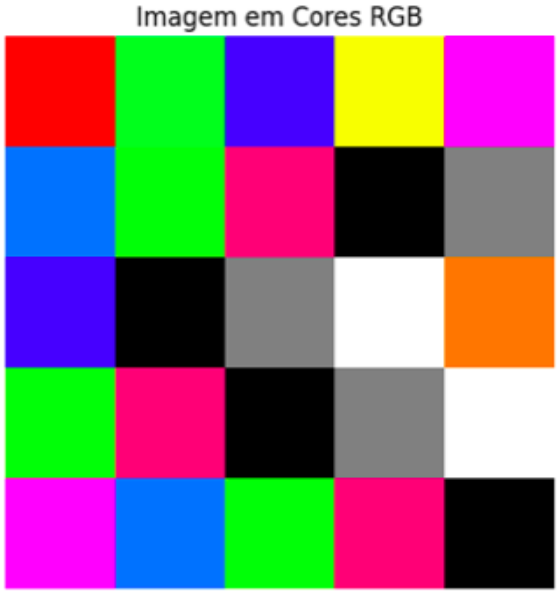
[8]:
```python
import numpy as np
import matplotlib.pyplot as plt
# Autor: Lucas Oliveira
# Matriz 5x5 representando uma imagem em preto e branco (valores de 0 a 255)
image_matrix = np.array([
    [0, 32, 64, 128, 255],
    [64, 128, 128, 192, 255],
    [128, 192, 192, 224, 255],
    [192, 224, 224, 240, 255],
    [255, 255, 255, 255, 255]
], dtype=np.uint8)

# Mostrar a matriz como uma imagem em preto e branco
plt.imshow(image_matrix, cmap='gray', vmin=0, vmax=255)
plt.title('Imagem em Preto e Branco')
plt.axis('off')  # Desativar eixos
plt.show()
```



Fonte: Elaborado pelo Autor

Figura 02: Exemplo de uma imagem colorida representada por uma matriz 5x5.

[9]:
```python
import numpy as np
import matplotlib.pyplot as plt
# Autor: Lucas Oliveira
# Matriz 5x5 representando uma imagem RGB (valores de 0 a 255)
image_matrix = np.array([
    [[255, 0, 0], [0, 255, 0], [0, 0, 255], [255, 255, 0], [255, 0, 255]],
    [[0, 128, 255], [128, 255, 0], [255, 0, 128], [0, 0, 0], [128, 128, 128]],
    [[0, 0, 255], [0, 0, 0], [128, 128, 128], [255, 255, 255], [255, 128, 0]],
    [[128, 255, 0], [255, 0, 128], [0, 0, 0], [128, 128, 128], [255, 255, 255]],
    [[255, 0, 255], [0, 128, 255], [128, 255, 0], [255, 0, 128], [0, 0, 0]]
])

# Mostrar a matriz como uma imagem RGB
plt.imshow(image_matrix)
plt.title('Imagem em Cores RGB')
plt.axis('off')  # Desativar eixos
plt.show()
```



Fonte: Elaborado pelo Autor

1.1 Processamento de Imagem:

O processamento de imagem é uma área da computação com foco na manipulação de imagens para melhorar a qualidade da imagem digital ou até mesmo obter informações relevantes contidas nas imagens. Utiliza-se métodos e técnicas usados para modificar ou analisar imagens digitais. Essas imagens podem ser obtidas por meio de várias fontes, como câmeras, telescópios ou outros dispositivos de imagem.

É uma disciplina bastante ampla e que desempenha um papel muito importante em diversas aplicações, podendo ser utilizado na visão computacional uma subárea da inteligência artificial que tem como objetivo capacitar máquinas a interpretar e compreender informações visuais a partir de imagens ou vídeos e também é bastante utilizado na medicina podendo ser utilizado para diagnósticos por imagem, reconhecimento e classificação de patologias, telemedicina e diversas outras áreas da medicina e ciência.

O processamento de imagem tem como principais objetivos aprimorar a qualidade das imagens, extrair informações, reconhecer padrões e até mesmo restaurar imagens. Para alcançar esses objetivos, é necessário uma série de etapas, essas etapas são essenciais para uma variedade de aplicações, incluindo a astronomia, onde o processamento de imagem desempenha um papel vital para aprimoramento e análise de imagens do espaço, permitindo uma compreensão mais profunda e detalhada do universo.

1.1.2 Processamento de Imagem na Astronomia:

A história do processamento de imagens é uma jornada de inovação, desde a invenção da fotografia até os primeiros computadores digitais. O avanço tecnológico, aliado aos programas espaciais norte-americano, abriu novas possibilidades nesse campo.

No ano de 1964, no Jet Propulsion Laboratory (JPL), técnicas computacionais revolucionárias começaram a ser aplicadas para aprimorar as imagens da lua que eram transmitidas por uma sonda Ranger. Essas técnicas

marcaram o início de uma nova era, corrigindo distorções na câmera de TV da sonda.

Além de elevar a qualidade das imagens espaciais, essas técnicas serviram como base para métodos avançados de realce e restauração em programas espaciais subsequentes. As expedições Apollo, por exemplo, colheram enormes benefícios desses avanços, consolidando sua importância nas missões tripuladas. (**Marques Filho, 1999**)

Na astronomia o formato mais utilizado para análise de imagens são imagens com formato FITS (Flexible Image Transport System), esse formato é um formato usado principalmente para armazenar e processar dados astronômicos, como imagens e dados espectrais. Esse tipo de arquivo é bastante utilizado para atender às necessidades da comunidade astronômica, pois é capaz de armazenar uma variedade de tipos de dados, incluindo tabelas, imagens e metadados relacionados.

As imagens com formato FITS tem duas vantagens sobre os outros formatos de arquivos, sendo uma dessas vantagens que os dados podem ser armazenados como qualquer tipo de número real e outra que esses dados podem representar qualquer coisa, incluindo medidas físicas. Esses arquivos precisam ser abertos e processados através de softwares ou bibliotecas específicos para tratamento dos arquivos e podem ser convertidos para outros formatos como .jpg ou .png por exemplo.

2. Objetivo:

O objetivo deste trabalho é realizar o processamento de imagens astronômicas obtidas pelo telescópio espacial James Webb por meio de uma aplicação Python com auxílio de bibliotecas disponíveis e uma breve análise nos dados obtidos.

O objetivo primordial deste trabalho é contribuir para o avanço contínuo da pesquisa astronômica, disponibilizando técnicas de processamento de imagens que tornam as imagens do espaço mais acessíveis e informativas

facilitando a compreensão mais profunda do cosmos e servir como recurso valioso para estudantes e pesquisadores interessados em aprofundar seus estudos no campo da astronomia.

Esse trabalho terá como foco o processamento digital de imagens na área da ciência de observação de corpos celestes, ou seja, será demonstrado aqui técnicas que poderão ser utilizadas por estudantes e pesquisadores que se interessam pela área de observação dos cosmos do qual auxiliarão no estudo e aprendizagem da astronomia.

2.1 Objetivos Específicos:

- Uso de técnicas de processamento de imagem para o melhoramento de astrofotografias.
- Estudar técnicas de processamento de imagem e análise de dados utilizando métodos de filtragem, ajuste de brilho, saturação, contraste e redução de ruído.
- Desenvolvimento de código Python para que possa ser utilizado no processamento de imagens, com auxílio de bibliotecas como por exemplo, AstroPy, NumPy dentre outros.
- Colaborar com a ciência de estudo astronômico através dos resultados obtidos.

2.1.2. Justificativa

Apesar dos avanços tecnológicos na área da astronomia, não é sempre que se obtém astrofotografias com qualidade suficiente para estudo pela comunidade científica, pois muitos fatores podem interferir na obtenção dessas imagens como, por exemplo: ruídos de fundo, distorções ópticas, baixa luminosidade dentre outros, fazendo com que as imagens obtidas percam algumas informações importantes.

Para melhoria das imagens existem diversas ferramentas e bibliotecas que podem ser utilizadas para auxílio e correção das astrofotografias, fazendo com

que o resultado final obtido seja de uma imagem que poderá ser utilizada pela comunidade científica para estudo e divulgação.

Com base nas informações citadas, o desenvolvimento deste trabalho dispõe-se a reduzir e até mesmo solucionar os desafios encontrados nos dados obtidos por essas tecnologias, utilizando técnicas de processamento de imagens e análise de dados.

3. Metodologia

Será feito o desenvolvimento de uma aplicação Python para analisar e processar os arquivos FITS(Flexible Image Transport System) de imagens astronômicas. Essa aplicação vai ler os arquivos FITS, analisar os dados do arquivo e realizar o processamento dos arquivos com imagens de filtros e canais diferentes na escala RGB utilizando técnicas de melhoramento de saturação e brilho das imagens em canais de cores diferentes, após a análise e melhoria será feito a junção dos arquivos para que o resultado final seja de apenas uma imagem melhorada e colorida.

A primeira etapa consiste na aquisição da imagem, que pode ser realizada por meio de câmeras ou outros dispositivos de captura de imagem e que posteriormente serão enviadas para os observatórios, nesse trabalho o Arquivo Mikulski para Telescópios Espaciais (MAST).

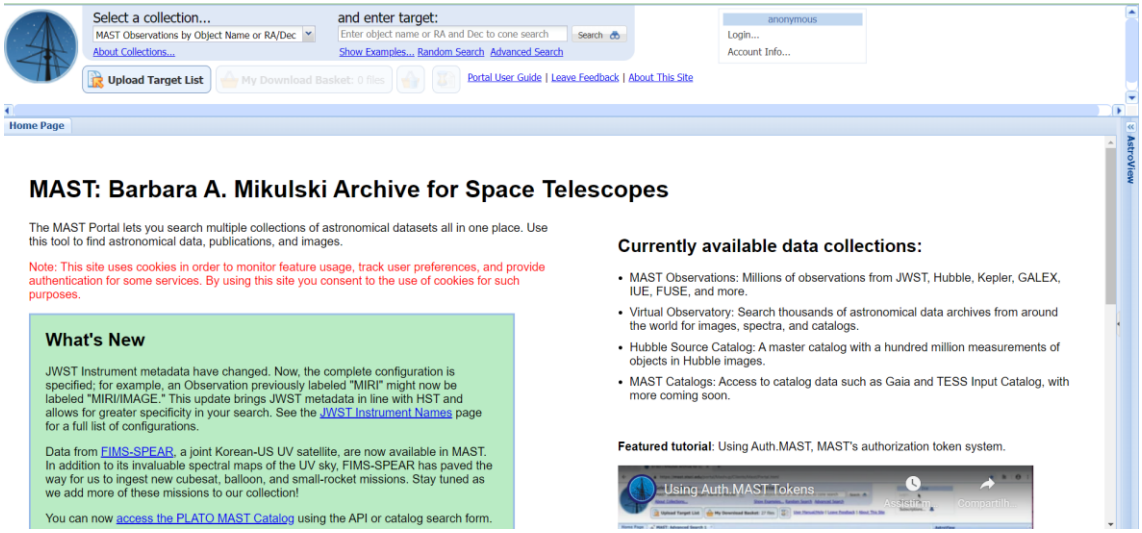
Em seguida, a imagem passa por um pré-processamento, onde são aplicadas técnicas para reduzir ruídos e corrigir imperfeições. Posteriormente, o aprimoramento da imagem visa melhorar sua qualidade, tornando os detalhes mais visíveis, muitas vezes usando técnicas como ajuste de contraste, saturação e equalização de histograma.

Na etapa de transformação de imagem, haverá processos como rotação e dimensionamento, para adaptar a imagem a fins específicos de análise ou apresentação. Finalmente, as imagens processadas são exibidas por meio da visualização, tornando os dados mais compreensíveis e úteis.

3.1. DESENVOLVIMENTO

Primeiramente é necessário realizar o download dos arquivos FITS que podem ser encontrados no MAST: Barbara A. Mikulski Archive for Space Telescopes(Figura 03) e é possível também realizar o download de alguns exemplos no NoirLab (Figura 04).

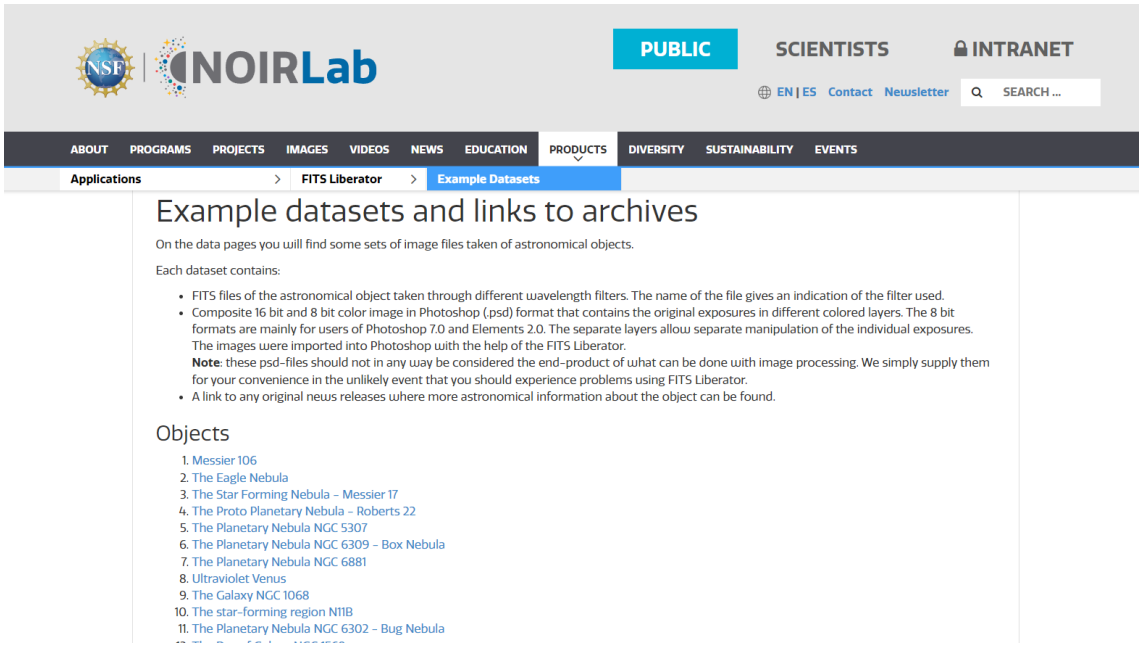
Figura 03: Portal do MAST.



Fonte: https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html

Acesso em: 02/10/2023.

Figura 04: Página do NoirLab onde é possível realizar o download de exemplos de arquivos FITS.



Fonte:

https://noirlab.edu/public/products/applications/fitsliberator/datasets

Acesso em: 02/10/2023.

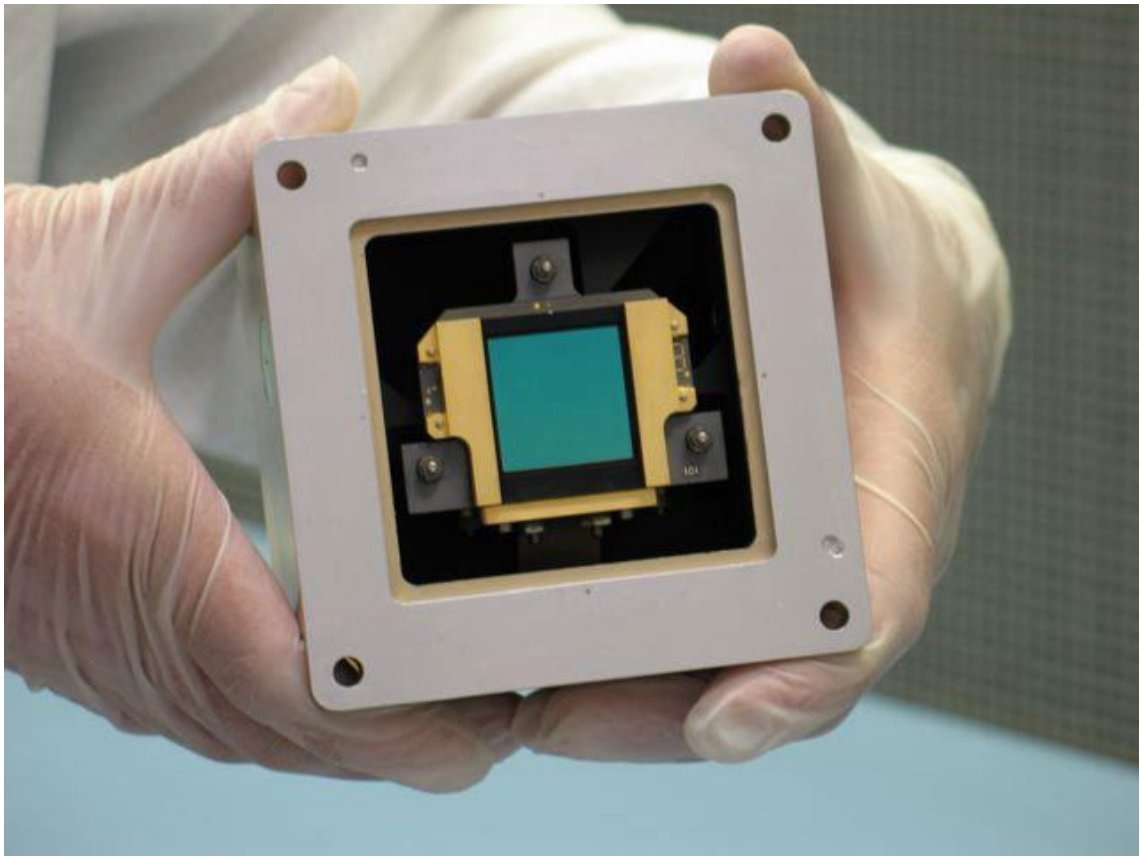
Para esse trabalho será feito a análise dos arquivos FITS da Nebulosa da Águia. Um aglomerado estelar aberto localizado na constelação de Serpente. O arquivo foi obtido diretamente no MAST e enviado pelo novo telescópio espacial JWST.

Serão utilizados arquivos FITS diferentes capturados pelo Mid-Infrared Instrument (MIRI) e o Near Infrared Camera (NIRCam) do Webb, cada um com um filtro de cor do telescópio espacial JWST.

Para a primeira imagem processada os filtros escolhidos foram os: F1500w representando a imagem em escala vermelha R, F1130w na escala de cor verde G e o F770w na escala Azul B. Enquanto que, para a segunda imagem os filtros escolhidos foram os: F470N representando a imagem em escala vermelha R, F090W na escala de cor verde G e F200W na escala azul B.

Instrumento de Infravermelho Médio (MIRI)

Figura 05: MIRI Instrument Detector

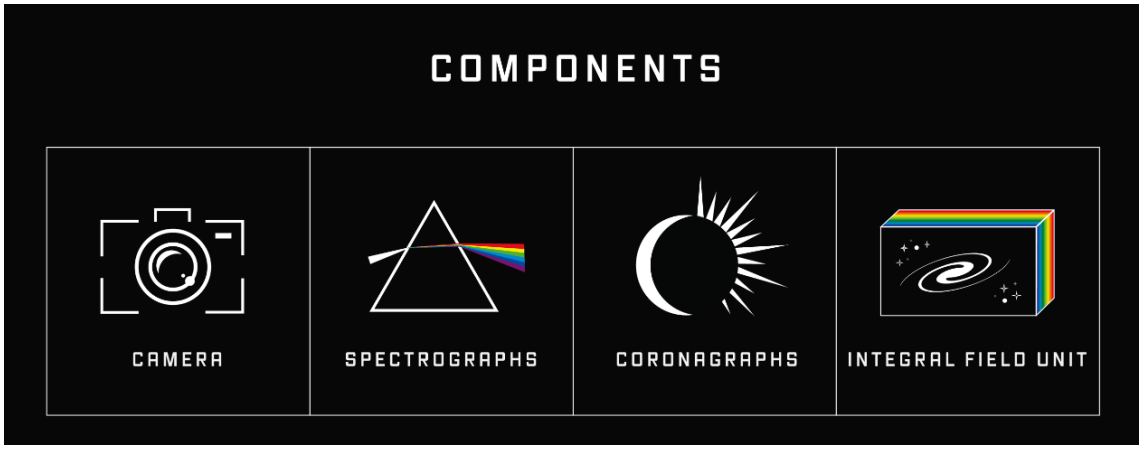


Fonte:
https://www.flickr.com/photos/nasawebbtelescope/4813321160/in/album-72157627124342395/ Acesso em: 05/102023.

De acordo com a documentação oficial do JWST[15] o Instrumento de Infravermelho Médio (MIRI) do telescópio fornece modos de observação de imagem e espectroscopia na faixa de aproximadamente 5 a 28 μm(Micrômetro), permitindo estudos dos astros mais jovens, análises de galáxias antigas com deslocamentos para o vermelho elevado e a análise de poeira quente e gás molecular.

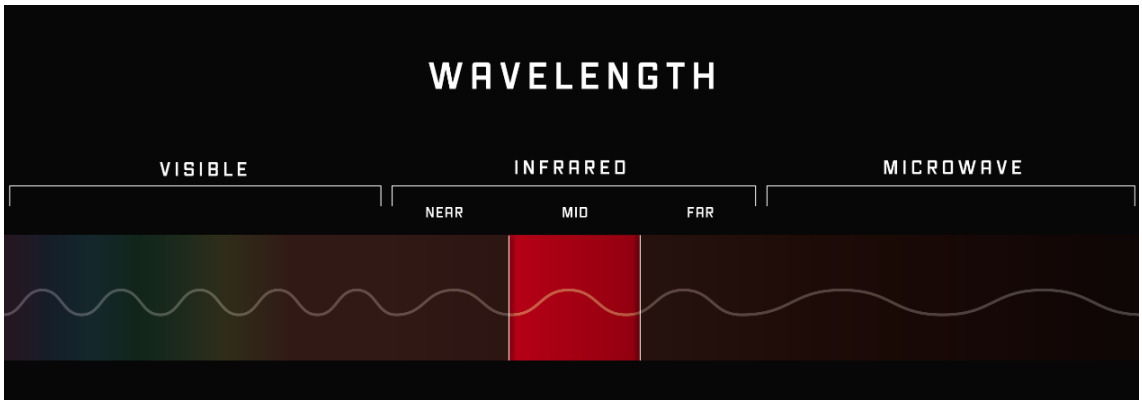
O MIRI fornece várias opções de observação, incluindo imagem e espectroscopia de baixa e média resolução com uma câmera e um espectrógrafo que é capaz de captar luz no infravermelho médio fornecendo novos detalhes físicos sobre os objetos distantes do qual observa.

Figura 06: Componentes do MIRICam



Fonte:
https://webbtelescope.org/contents/media/images/01FA0SZA5HPXKRKH8Y6PKB10V1?page=18&filterUUID=91dfa083-c258-4f9f-bef1-8f40c26f4c97
Acesso em: 16/10/2023.

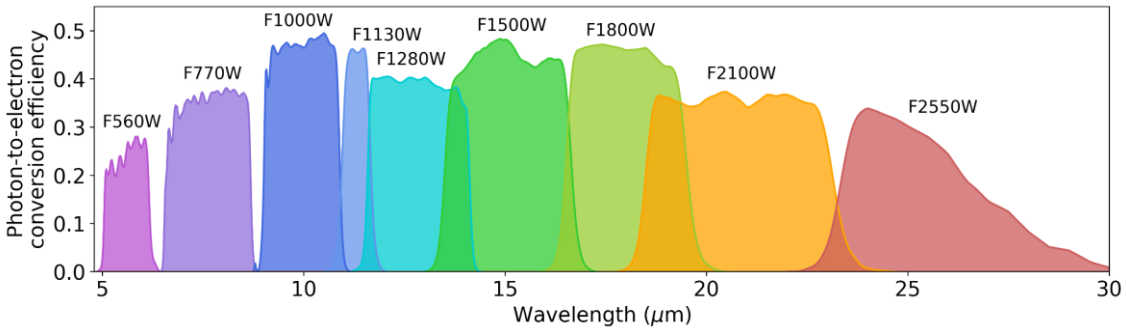
Figura 07: Comprimento de onda IR do MIRICam



Fonte:
https://webbtelescope.org/contents/media/images/01FA0SZA5HPXKRKH8Y6PKB10V1 Acesso em: 16/10/2023.

Filtros do MIRICam:

Figura 08: Curvas de filtro de imagem do MIRI



Fonte:https://jwst-docs.stsci.edu/jwst-mid-infrared-instrument

Acesso em: 16/10/2023.

Near Infrared Camera (NIRCam)

Figura 09: O instrumento NIRCam do JWST após passar nos testes da Lockheed e ser preparado para o envio até Goddard.

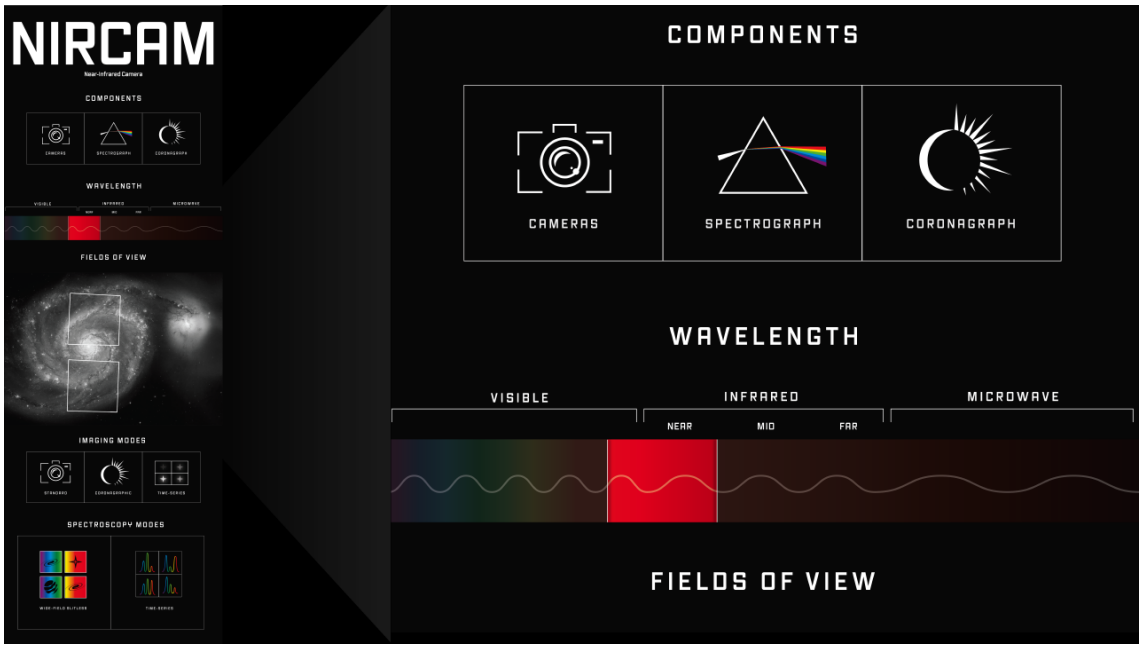
Fonte:
https://www.flickr.com/photos/nasawebbtelescope/4813329922/in/photolist-2oFYTBU-2nxLtWy-2o3hyLq-2oG4SnN-2nxQ73N-2oPHKYc-fmf3UY-2nxM1ES-2nBXgre-2nMZB56-2nxSb5C-ejcWcJ-2nyexzP-2nxMJP7-2nVgpCF-efMxLE-8kkzhW-ejcW93-2o5534c-fmd8GA-2nCVLVU-2nxQ95D-2nUQv7v-2nxMDrZ-2nEkyWM-2ofuXyE-2ny9zUP-2o3EzNY-2nyMWK8-2nyMY6K-2nyNUcc-2o3GqhP-2nyP1rS-2nycjX1-2nyzh6d-2nzDwak-2ny6mzs-2ny4WKC-2nz8WB6-2nRzoLC-2nxVKGc-2o3Da2c-2nLXNYV-2ny78mt-2nxXgTq-2nABkTG-2o7rEHe-2nRwKUy-2oTwh8W-2oTwhhi
Acesso em: 16/10/2023.

O NIRCam desenvolvido como parte do JWST é uma câmera que opera na faixa de 0,6 a 5 mícrons no JWST, projetado pela NASA como sucessor do Hubble. O instrumento NIRCam foi construído e desenvolvido pela Northrop

Grumman, em colaboração com várias instituições e equipes científicas, como parte do programa JWST e tem dois módulos de instrumento que apontam para campos de visão adjacentes, permitindo a observação de dois comprimentos de onda simultaneamente usando dicróicos, que é um tipo especial de espelho que reflete ou transmite luz de maneira seletiva com base na frequência (comprimento de onda) da luz.

No contexto do NIRCam no JWST, o uso de dicróicos permite observar simultaneamente em duas faixas de comprimento de onda distintas, tornando-o um instrumento versátil para a pesquisa astronômica. Ele oferece diversas capacidades, incluindo imagens, espectroscopia, monitoramento de estrelas e observação de exoplanetas. O objetivo é detectar as primeiras galáxias que emitiram luz após o Big Bang. O NIRCam tem cinco modos de observação e é um instrumento importante para a pesquisa em astronomia.

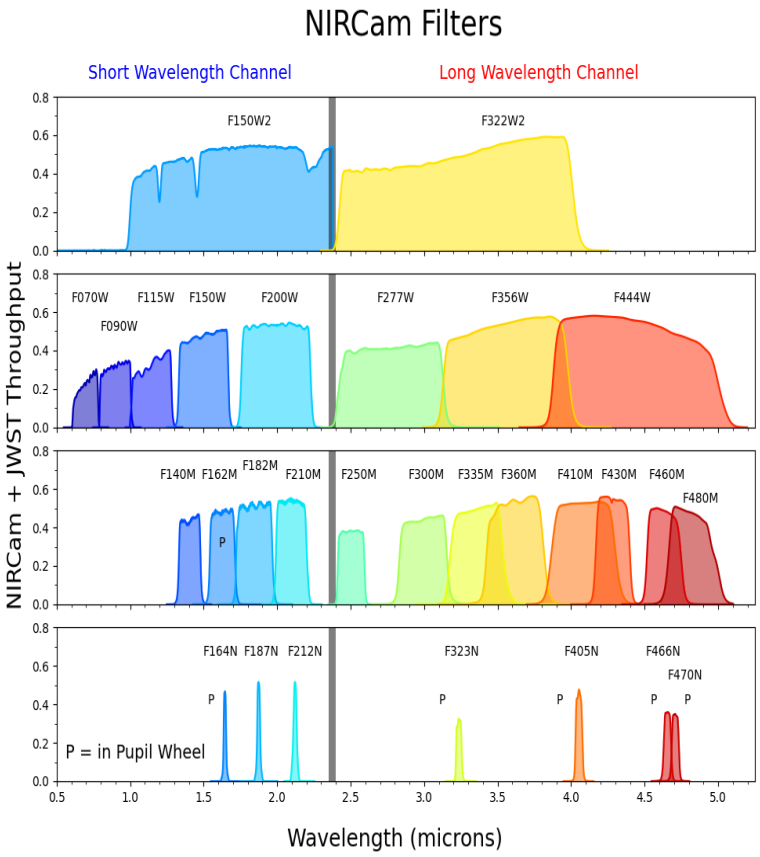
Figura 10: Instrumentos Científicos no Telescópio Espacial James Webb: Câmera de Infravermelho Próximo (NIRCam)



Fonte: https://webbtelescope.org/contents/media/images/01FA0SZSEW1TZ51BHG0EGW2EZP Acesso em: 16/10/2023.

Filtros do NIRCam:

Figura 11: Filtros do NRICam.



Fonte: https://jwst-docs.stsci.edu/jwst-near-infrared-camera Acesso em: 16/10/2023.

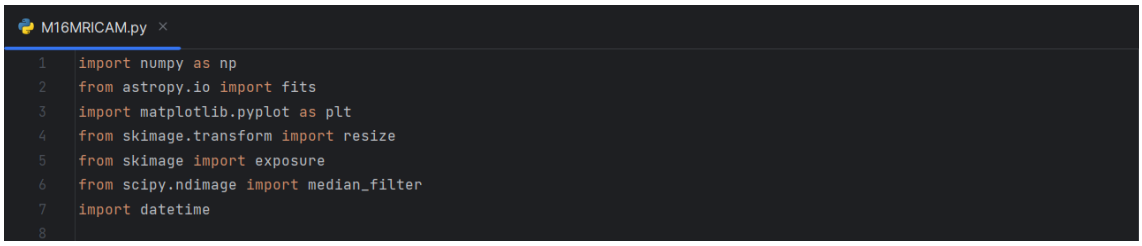
Interface Gráfica

Neste trabalho será feito o desenvolvimento de uma aplicação Python onde será possível realizar o processamento dos arquivos FITS obtidos. Para manipulação dos arquivos FITS será utilizado a ferramenta Python juntamente com auxílio de algumas bibliotecas, sendo essas: AstroPy, Numpy, matplotlib e skimage dentre outras bibliotecas.

1. **Importação de bibliotecas (Figura 12):** O código consiste primeiro na importação das bibliotecas Python necessárias para execução do código sendo elas:

- Numpy é importado como np para suporte a operações numéricas.
- astropy.io.fits será importado para lidar com arquivos FITS.
- matplotlib.pyplot é importado para exibir a imagem resultante e os histogramas.
- skimage.transform.resize vai ser utilizado para redimensionar as imagens.
- skimage.exposure é importado para realizar ajustes no brilho e saturação da imagem.
- scipy.ndimage.median_filter é importado para aplicar um filtro de mediana nas imagens.
- datetime será utilizado para salvar os arquivos com base na data e hora atual da execução do código.

Figura 12: Importando as bibliotecas Python.



Fonte: Elaborado pelo Autor

2. **Definição de funções (Figura 13):** Nessa etapa será criado as funções para o pré-processamento dos arquivos.

- preprocess_data(data, brilho=1.0, saturação=1.0, denoise=False) é uma função que pré-processa os dados. Ela ajusta o brilho, reduz a saturação e pode aplicar uma redução de ruído opcional usando um filtro de mediana.
- denoise_image(data, tamanho_filtro=1) é uma função que aplica um filtro de mediana para reduzir o ruído na imagem.

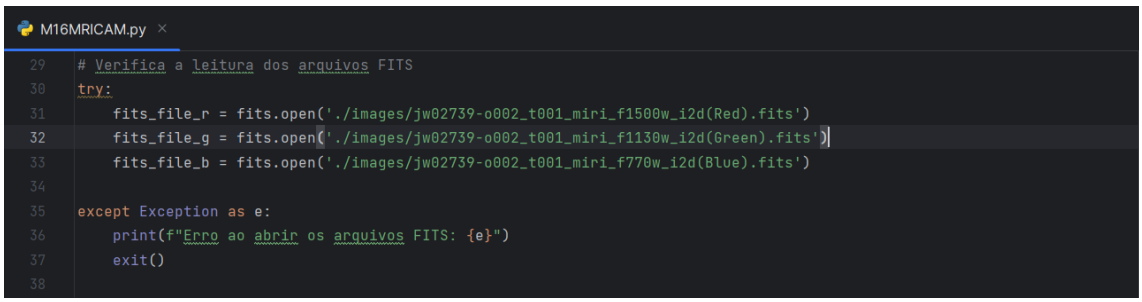
Figura 13: Função para o pré-processamento dos arquivos.

Fonte: Elaborado pelo Autor

3. **Abertura de arquivos FITS (Figura 14):**

- Ao ser executado o código tenta abrir os arquivos FITS correspondentes aos canais vermelho, verde e azul. Em caso de erro, uma mensagem de erro é exibida e a execução do programa é encerrada.

Figura 14: Leitura dos arquivos FITS obtidos.



Fonte: Elaborado pelo Autor

4. **Redimensionamento das imagens (Figura 15):**

- As imagens FITS vão ser redimensionadas para um tamanho comum de 1280x1280 pixels.

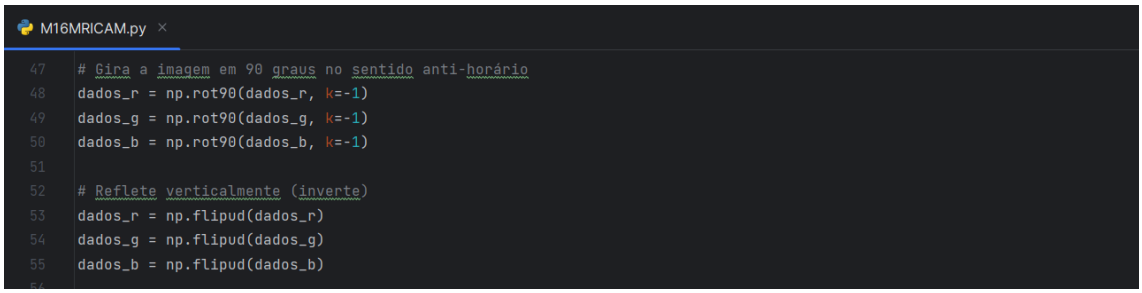
- Cada imagem é pré-processada usando a função preprocess_data com parâmetros de brilho, saturação e redução de ruído específicos.

Figura 15: Ajustando as imagens para um tamanho comum.

Fonte: Elaborado pelo Autor

5. **Rotação e inversão das imagens (Figura 16):** Para uma melhor visualização e análise será feito a rotação da imagem em 90 graus no sentido anti-horário e em seguida, cada imagem é refletida verticalmente (invertida).

Figura 16: Ajustando a visualização da imagem realizando a uma rotação em 90° no sentido anti-horário.



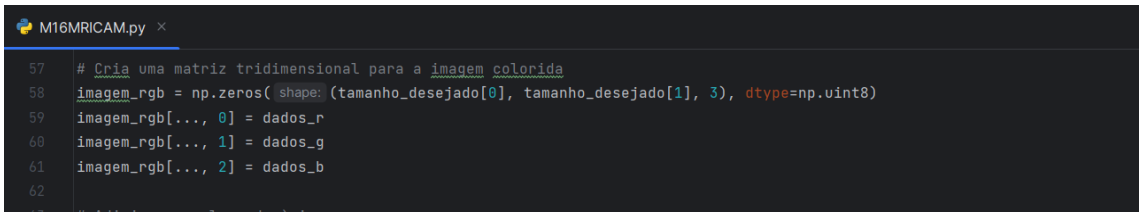
Fonte: Elaborado pelo Autor

6. **Criando a imagem colorida (Figura 17):**

- Uma matriz tridimensional imagem_rgb é criada para a imagem colorida, com três canais: vermelho, verde e azul.

- Os canais de cores são preenchidos com os dados processados das imagens redimensionadas.

Figura 17: Criando a imagem colorida com base nos arquivos FITS lidos no início do código.

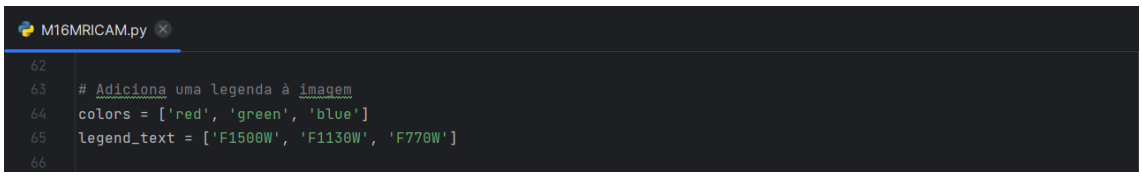


Fonte: Elaborado pelo Autor

7. **Adicionando legenda à imagem** (Figura 18)**:**

- Uma legenda é adicionada à imagem, indicando os canais e suas respectivas cores.

Figura 18: Adicionando uma legenda para a imagem processada.



Fonte: Elaborado pelo Autor

8. **Exibindo o histograma e a imagem processada (Figura 19):**

- São criados histogramas para cada canal (vermelho, verde e azul) e exibidos em janelas separadas.
- As imagens coloridas e os histogramas são exibidos.

Figura 19: Exibindo a imagem processada.

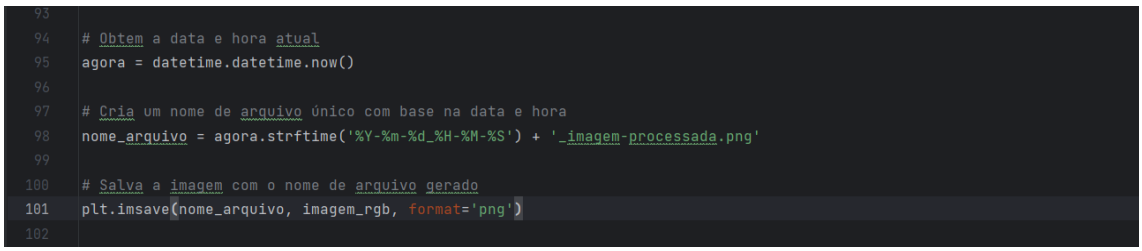
```
M16MRICAM.py

# Exibe a imagem colorida e os histogramas em janelas separadas
plt.figure(1)
plt.imshow(imagem_rgb)
plt.title('M16, Eagle Nebula JWST MRICam')
plt.axis('off')
plt.legend(handles=[plt.Line2D( xdata: [0],  ydata: [0], color=colors[i], label=text) for i, text in enumerate(legend_text)],loc='upper left')

plt.figure(2)
plt.hist(dados_r.ravel(), bins=256, range=(0, 255), color='red', alpha=0.5)
plt.title('Histograma Canal Vermelho')
plt.xlabel('Valor do Pixel')
plt.ylabel('Frequência')

plt.figure(3)
plt.hist(dados_g.ravel(), bins=256, range=(0, 255), color='green', alpha=0.5)
plt.title('Histograma Canal Verde')
plt.xlabel('Valor do Pixel')
plt.ylabel('Frequência')

plt.figure(4)
plt.hist(dados_b.ravel(), bins=256, range=(0, 255), color='blue', alpha=0.5)
plt.title('Histograma Canal Azul')
plt.xlabel('Valor do Pixel')
plt.ylabel('Frequência')

plt.show()
```

Fonte: Elaborado pelo Autor

9. **Salvando a imagem processada no computador (Figura 20):**

- É gerado um nome de arquivo único com base na data e hora atuais.
- A imagem colorida é salva em formato PNG com o nome de arquivo gerado.

Figura 20: Salvando o arquivo processado.



```
# Obtem a data e hora atual
agora = datetime.datetime.now()

# Cria um nome de arquivo único com base na data e hora
nome_arquivo = agora.strftime('%Y-%m-%d_%H-%M-%S') + '_imagem-processada.png'

# Salva a imagem com o nome de arquivo gerado
plt.imsave(nome_arquivo, imagem_rgb, format='png')
```

Fonte: Elaborado pelo Autor

10. **Fechando os arquivos FITS (Figura 21):**

- Os arquivos FITS são fechados após o processamento.

Figura 21: Fechando os arquivos FITS.

```
# Fecha os arquivos FITS
fits_file_r.close()
fits_file_g.close()
fits_file_b.close()
```

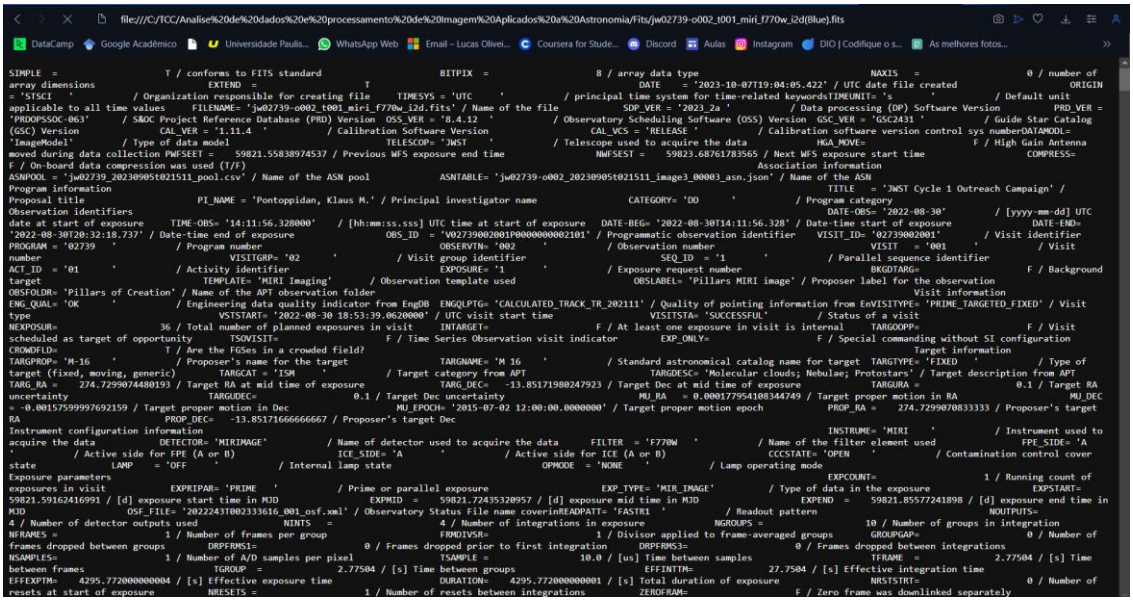
Fonte: Elaborado pelo Autor

Processando a Nebulosa da Águia com os filtros do MIRIcam:

O código foi desenvolvido para ler arquivos FITS, processar os dados astronômicos armazenados nos arquivos e ser capaz de extrair informações de canais específicos, como vermelho, verde e o azul, onde realiza uma série de operações para melhorar a qualidade e utilidade das imagens resultantes.

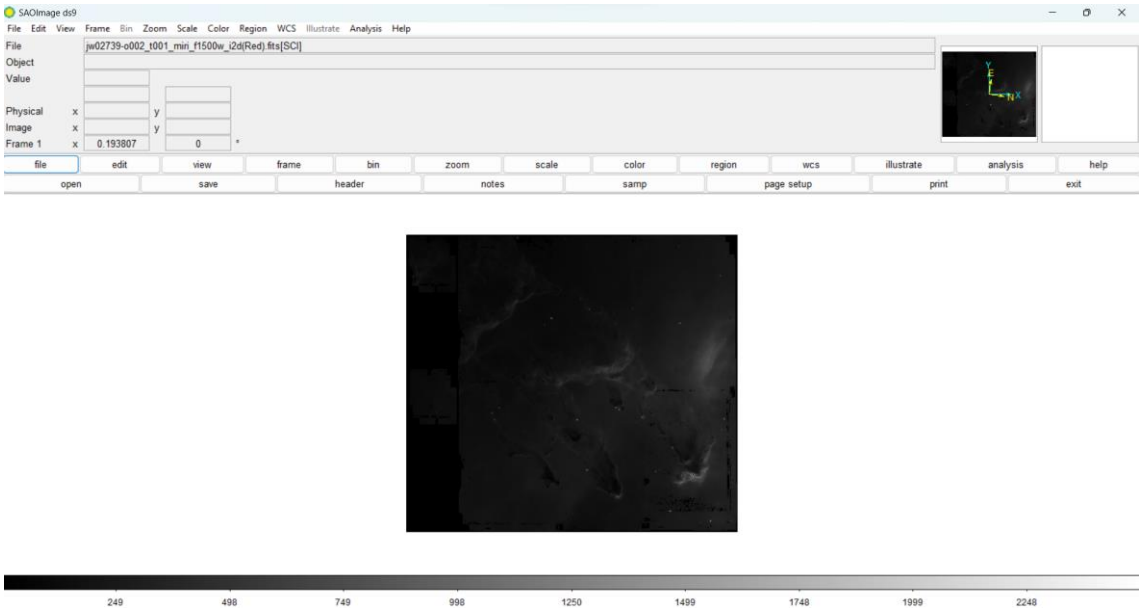
Para melhor demonstração da visualização das imagens,foi utilizado o Software SAOImage DS9 (Figuras 23, 24, 26, 27, 29 e 30), especializado em lidar com formatos FITS, uma vez que esses arquivos demandam softwares específicos ou bibliotecas para processamento. Caso o arquivo não seja aberto por uma aplicação dedicada, apenas um conjunto de dados e informações armazenadas no formato FITS será observado, como ilustrado na figura 22 abaixo.

Figura 22: Arquivo FITS aberto no navegador Opera GX.



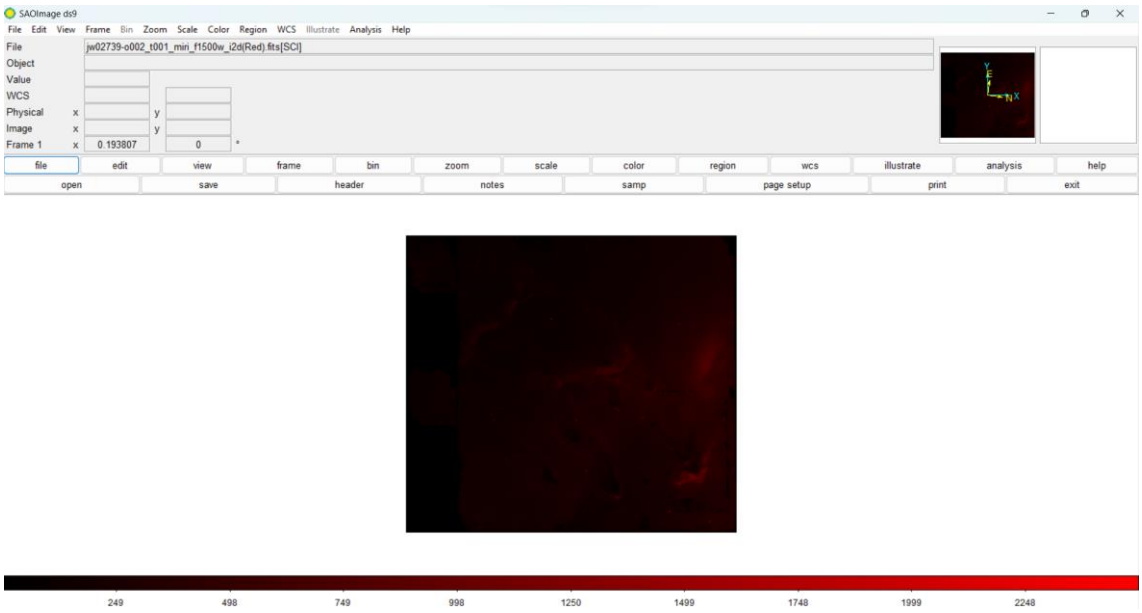
Fonte: Elaborado pelo Autor

Figura 23: Visualização do arquivo com filtro f1500w ao ser aberto pelo Software SAOImage DS9



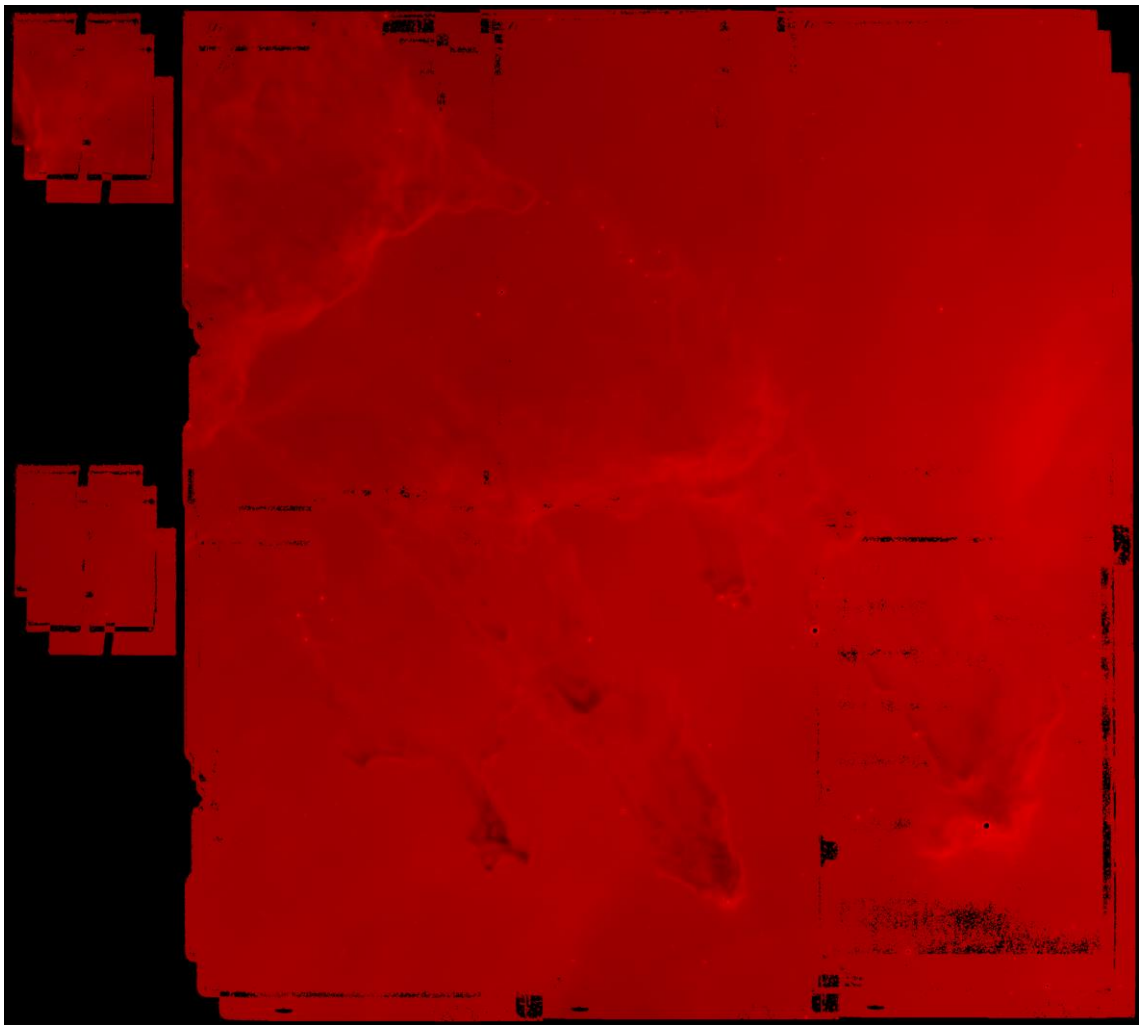
Fonte: Elaborado pelo Autor

Figura 24: Visualização do arquivo com filtro f1500w ao ser aberto pelo Software SAOImage DS9 aplicando cor vermelha.



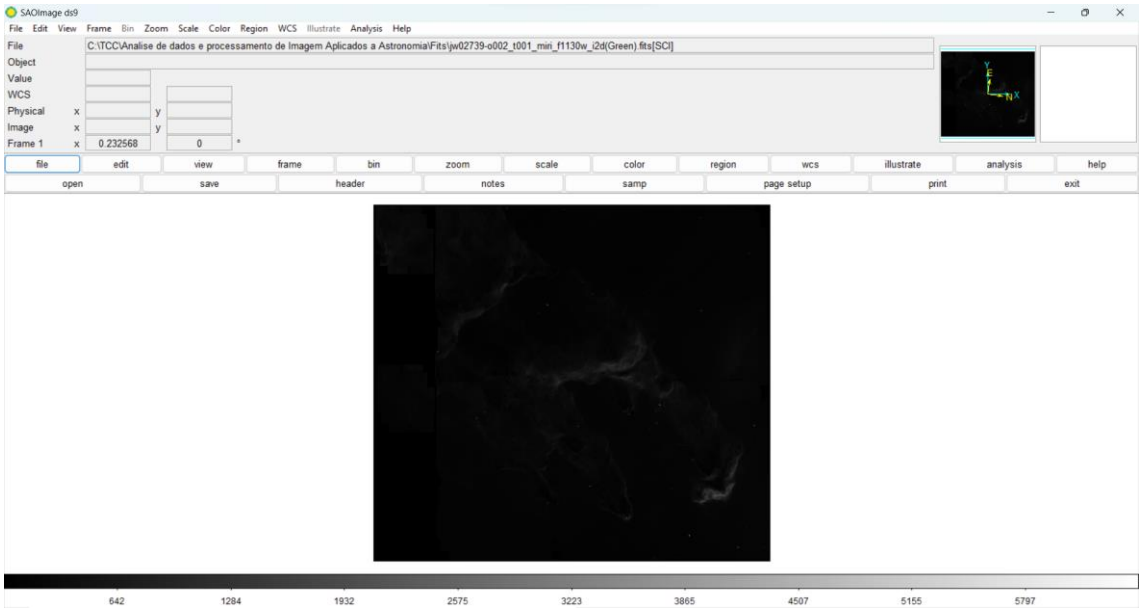
Fonte: Elaborado pelo Autor

Figura 25: Arquivo FITSf1500w_i2d com filtro de cor vermelho.



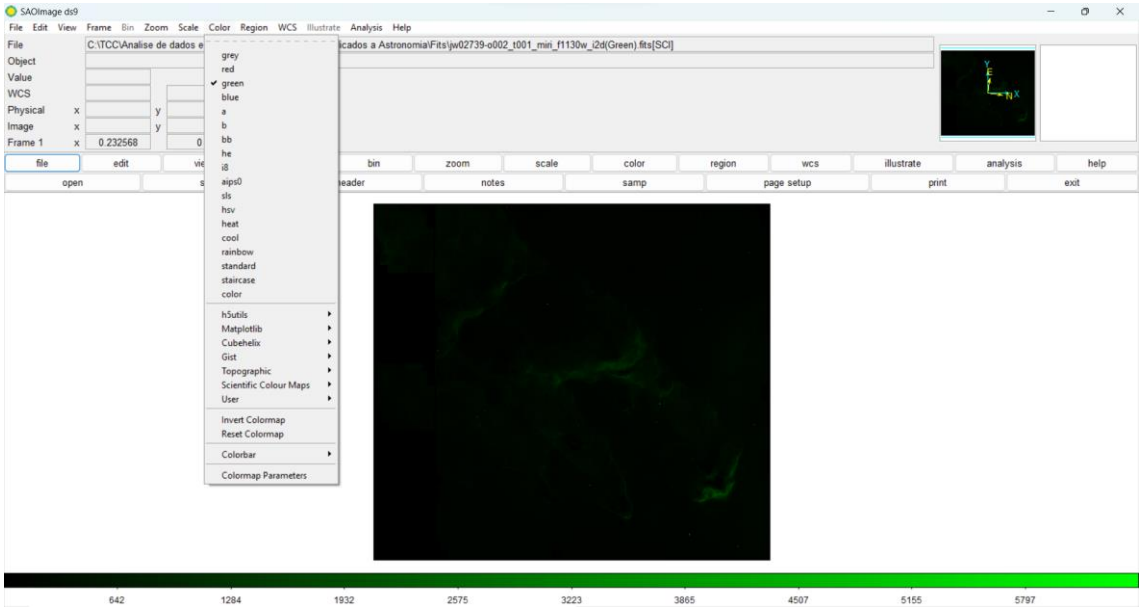
Fonte: Elaborado pelo Autor

Figura 26: Visualização do arquivo com filtro f1130w ao ser aberto pelo Software SAOImage DS9



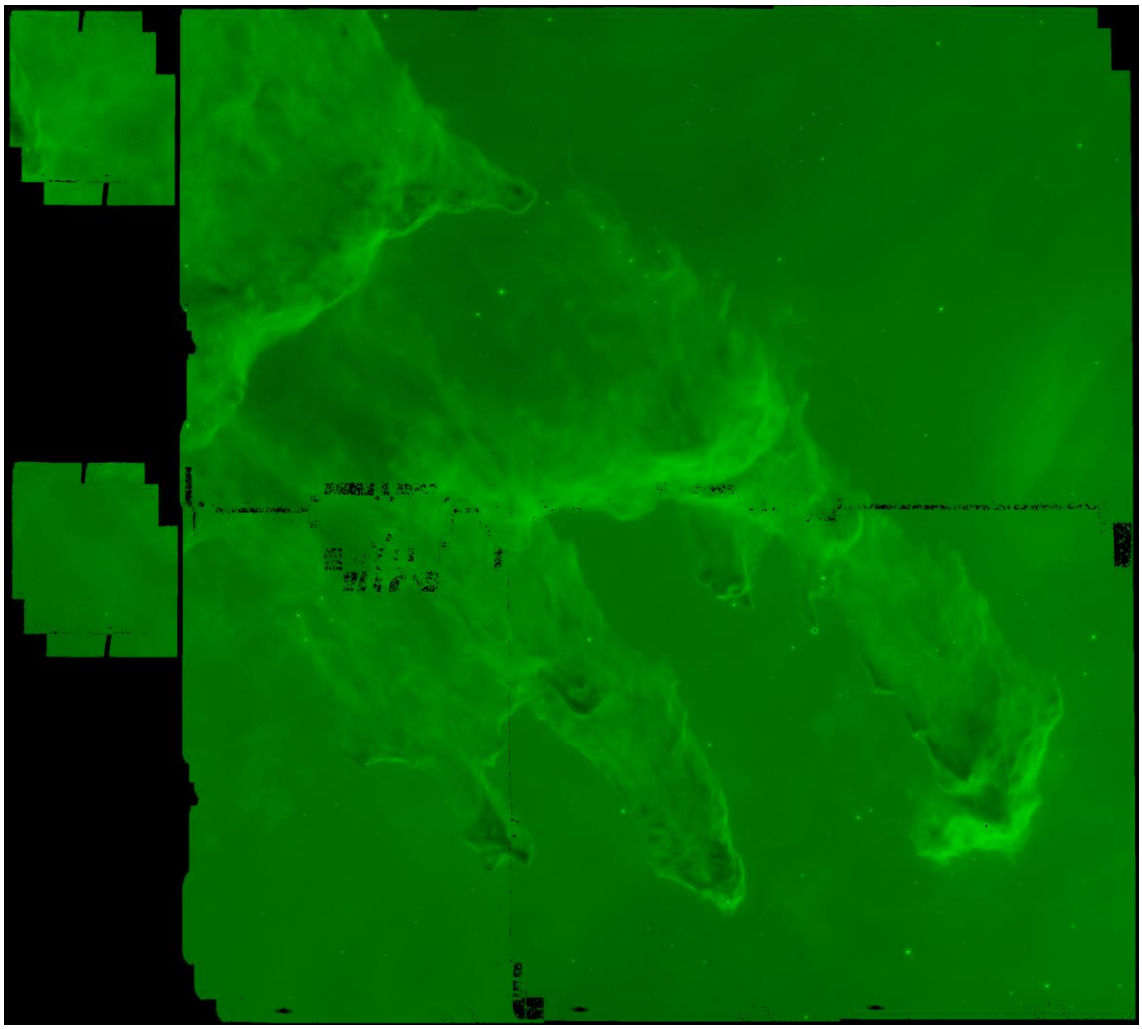
Fonte: Elaborado pelo Autor

Figura 27: Visualização do arquivo com filtro f1130w ao ser aberto pelo Software SAOImage DS9 aplicando cor verde.



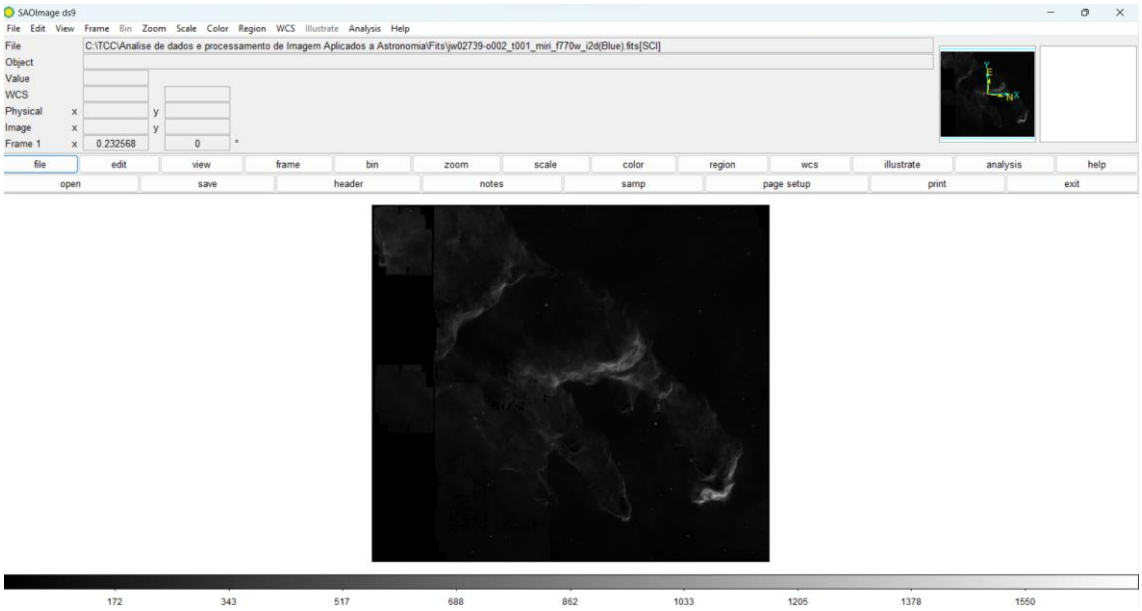
Fonte: Elaborado pelo Autor

Figura 28: Arquivo FITS f1130w_i2d com filtro de cor verde.



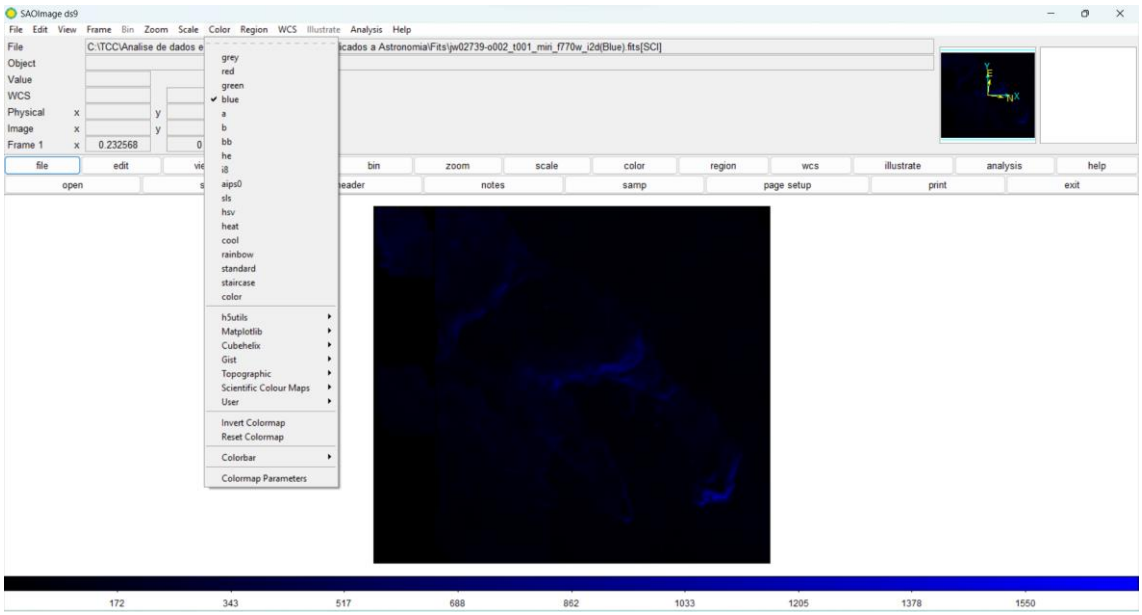
Fonte: Elaborado pelo Autor

Figura 29: Visualização do arquivo com filtro f770w ao ser aberto pelo Software SAOImage DS9



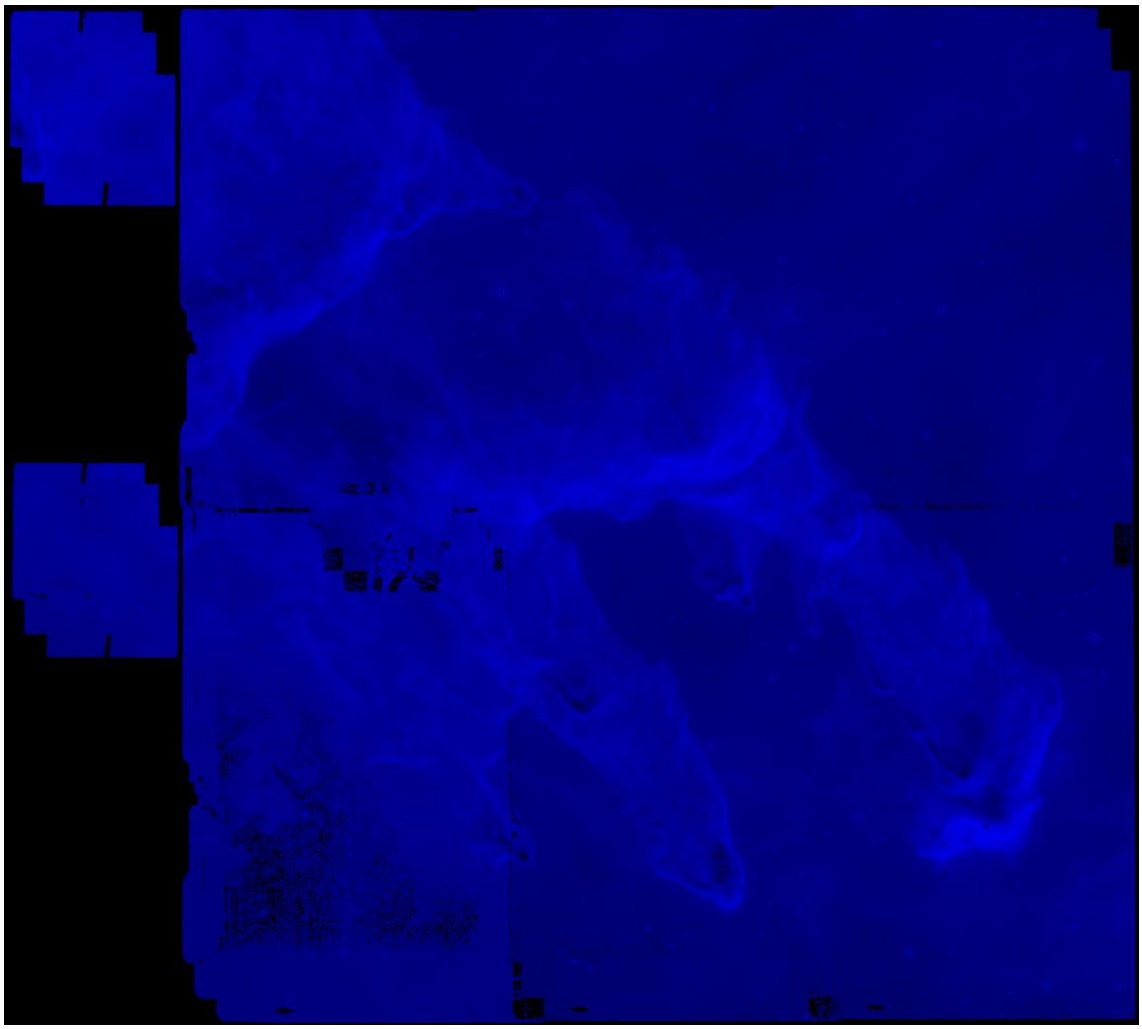
Fonte: Elaborado pelo Autor

Figura 30: Visualização do arquivo com filtro f770w ao ser aberto pelo Software SAOImage DS9 aplicando cor azul.



Fonte: Elaborado pelo Autor

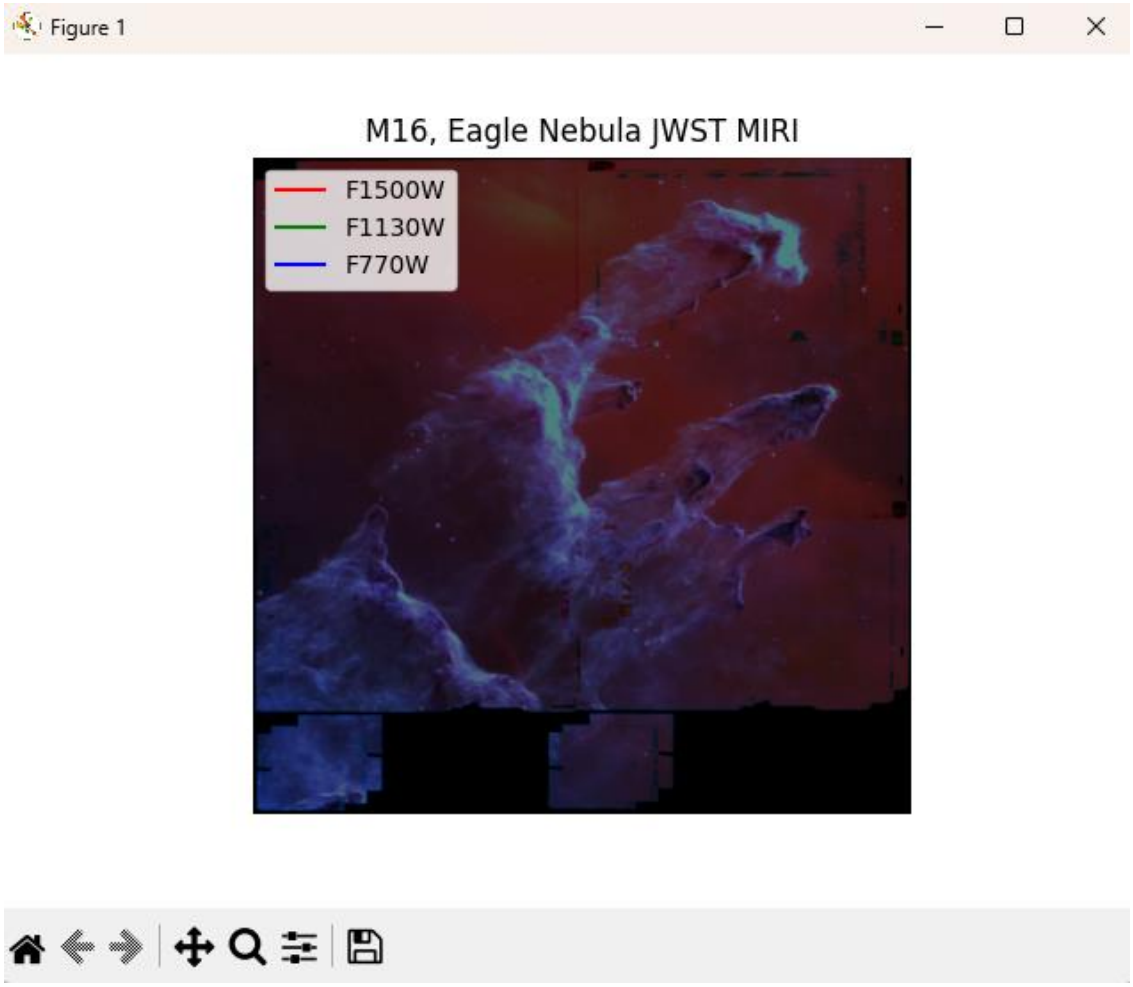
Figura 31: Arquivo FITS f770w_i2d com filtro de cor azul.



Fonte: Elaborado pelo Autor

Após a execução do código desenvolvido e junção dos arquivos FITS o resultado final obtido é uma imagem da Nebulosa da Águia(Figura 32) com informações mais visíveis onde já é possível observar pequenos pontos de algumas estrelas em formação.

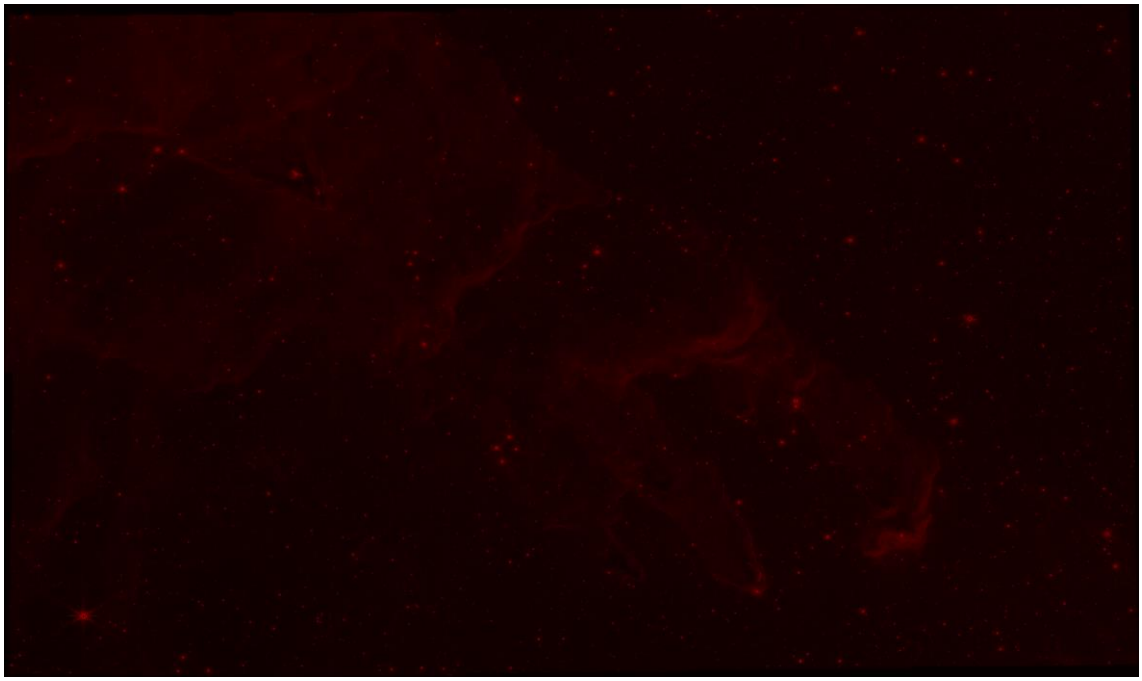
Figura 32: Visualização da Nebulosa M16 com Filtros: f1500w, f1130w e f770w MIRI JWST após execução do código desenvolvido.



Fonte: Elaborado pelo Autor

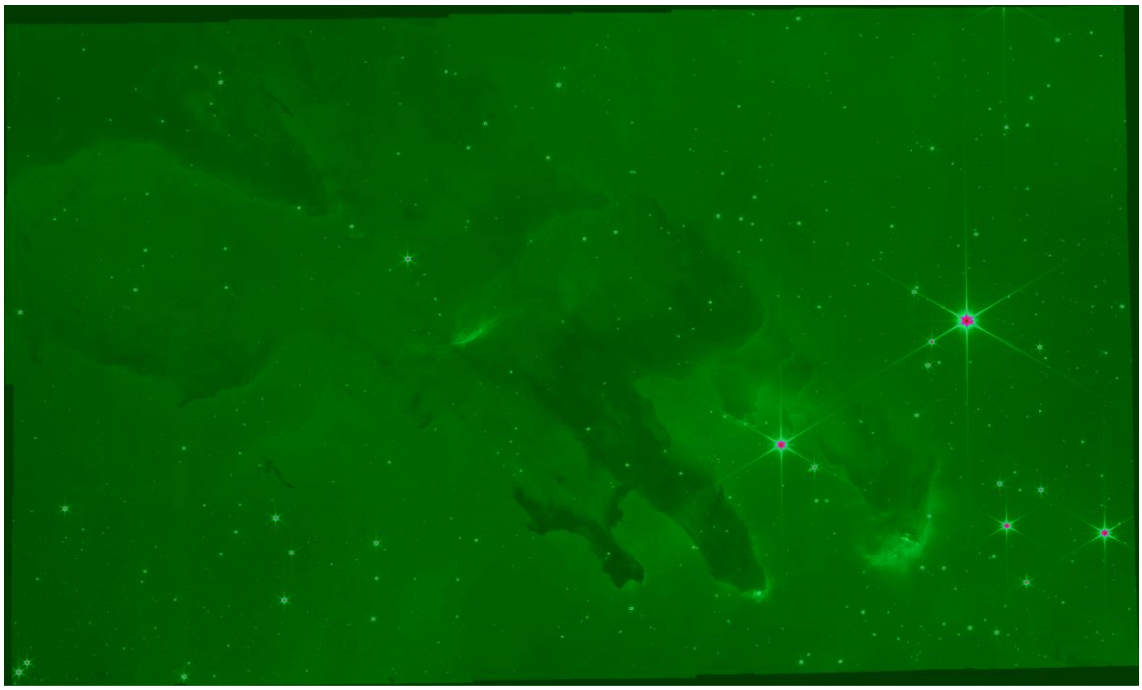
Processando a Nebulosa da Águia com os filtros do NIRcam:

Figura 33: Arquivo FITS f470n_i2d com filtro de cor vermelho.



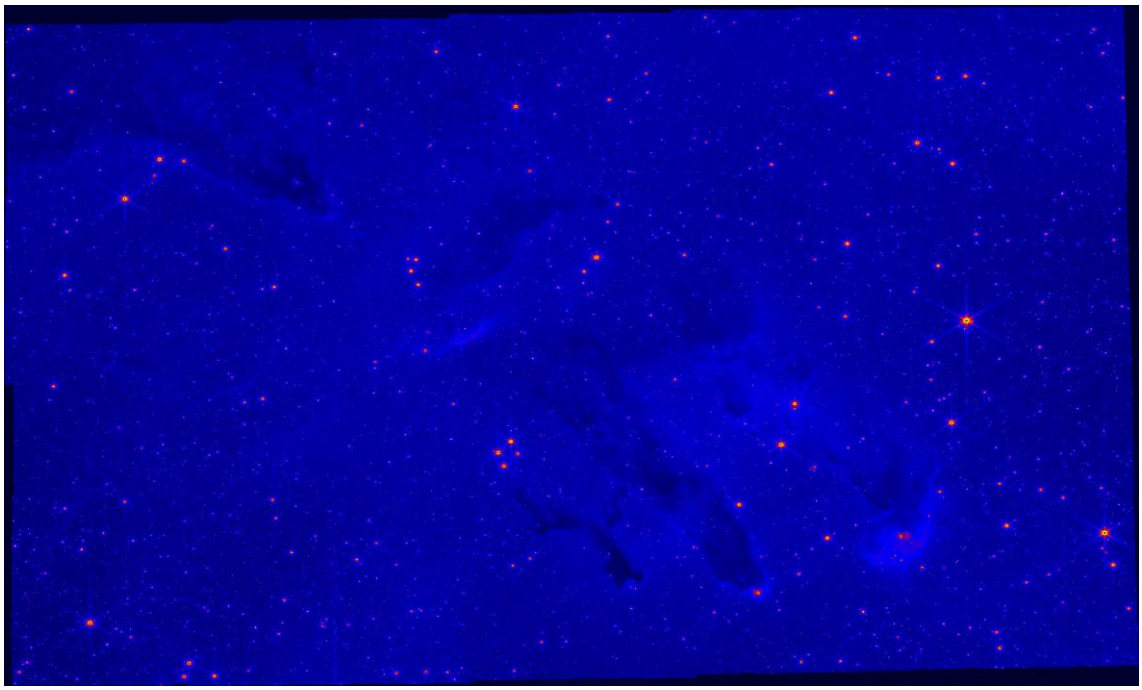
Fonte: Elaborado pelo Autor

Figura 34: Arquivo FITS f200w_i2d com filtro de cor verde.



Fonte: Elaborado pelo Autor

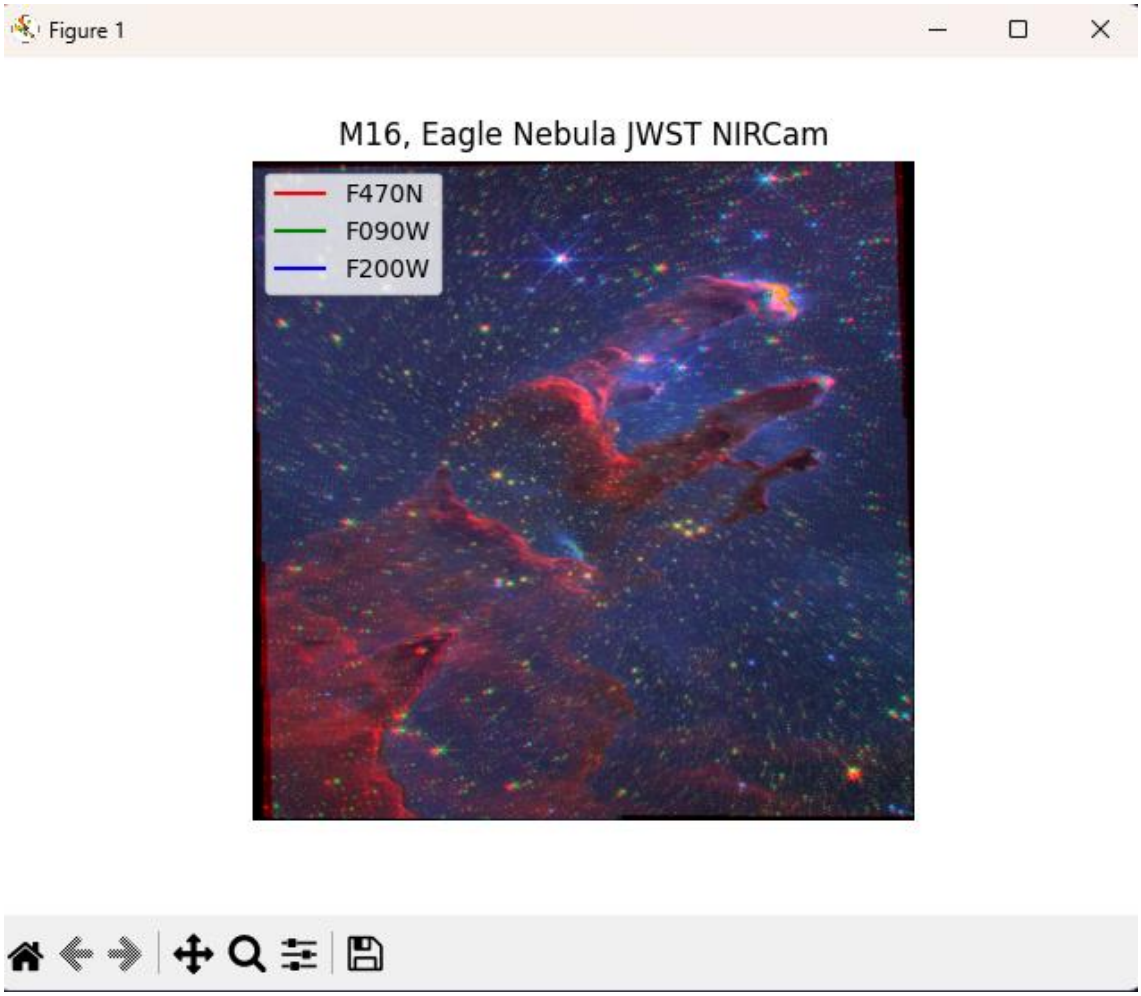
Figura 35: Arquivo FITS f090w_i2d com filtro de cor azul.



Fonte: Elaborado pelo Autor

O código utilizado no processamento da imagem do MIRICam é semelhante ao código que também será utilizado para o processamento dos arquivos FITS com filtros da NIRCam onde será feito apenas ajustes no código para se obter o resultado da figura 36 abaixo com mais informações visíveis.

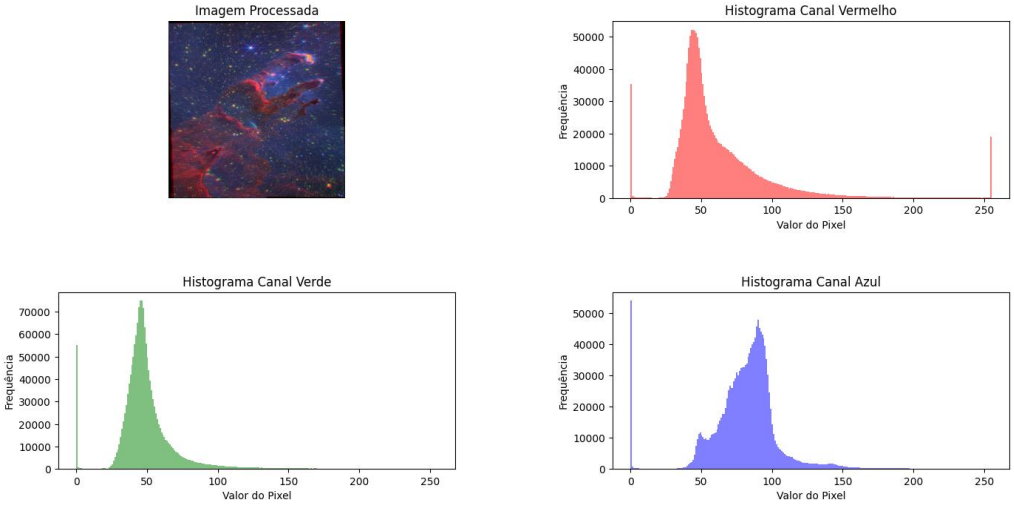
Figura 36: Visualização da Nebulosa M16 NIRCam com Filtros: F470N, F090W e F200W após execução do código Python desenvolvido.



Fonte: Elaborado pelo Autor

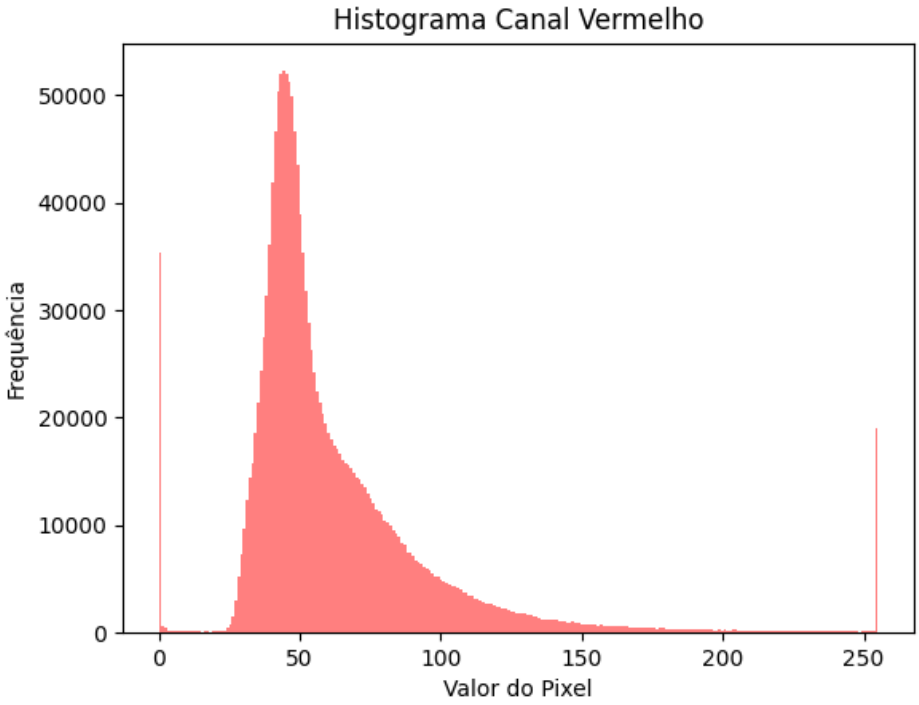
Tanto o código utilizado para processar as imagens do NIRCam quanto o utilizado para processamento das imagens do MIRICam contam com a exibição de histogramas onde será possível verificar informações das imagens nos canais de cores RGB (Figuras 37, 38, 39 e 40).

Figura 37: Interface apresentada após a execução do código.



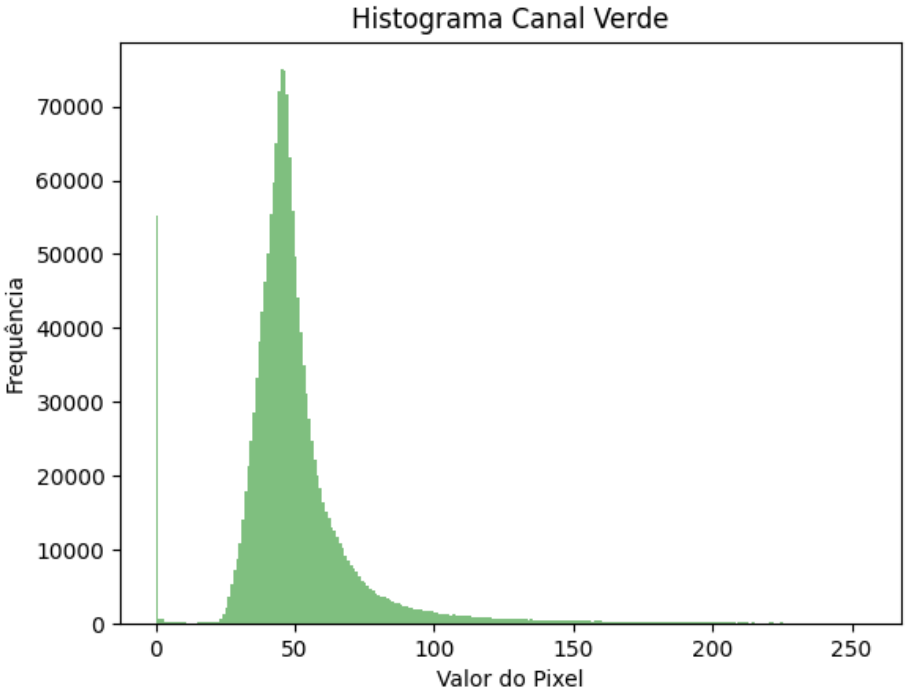
Fonte: Elaborado pelo Autor

Figura 38: Histograma do arquivo FITS f470n_i2d.



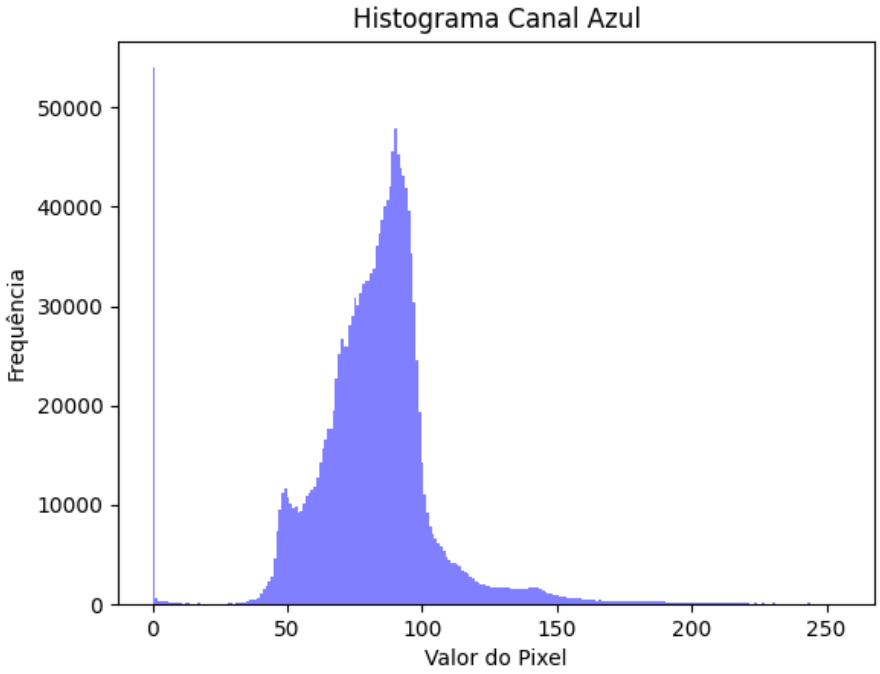
Fonte: Elaborado pelo Autor

Figura 39: Histograma do arquivo FITS f200w_i2d.



Fonte: Elaborado pelo Autor

Figura 40: Histograma do arquivo f090w_i2d.



Fonte: Elaborado pelo Autor

3.1.2. Comparando os histogramas:

Após realizar o processamento das imagens, será utilizado uma função do OpenCV para analisar a similaridade entre dois histogramas nos canais de cores RGB, sendo esses histogramas o da imagem processada utilizando o NIRCam(Figura 36) e o histograma da imagem processada com MIRICam(Figura 32).

O método **“cv2.HISTCMP_BHATTACHARYYA”**, presente no OpenCV, é uma ótima ferramenta para análise de imagens astronômicas. Essa função pode ser usada para avaliar a semelhança entre histogramas de intensidade de pixels em imagens do espaço.

Um valor baixo indica concordância, especialmente próximo a zero, sugerindo alta similitude, enquanto valores mais altos indicam maior diferença

ou dissimilaridade entre as distribuições. Isso é crucial na análise de imagens, inclusive na astronomia, onde a identificação de objetos celestes, eventos ou fenômenos requer a avaliação da similaridade entre distribuições de intensidade de pixel.

Esse método pode ser aplicado na identificação de objetos celestes, acompanhamento de eventos astronômicos, estudo da evolução estelar, classificação de galáxias e correção de imagens astronômicas. O **“cv2.HISTCMP_BHATTACHARYYA”** é uma ferramenta fundamental na pesquisa e no desenvolvimento de sistemas de análise de imagens astronômicas, contribuindo para a compreensão do cosmos.

Figura 41: Distância Bhattacharyya

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

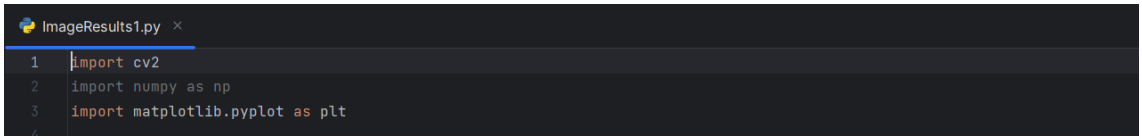
(Fonte: Documentação OpenCV)

Explicando a aplicação desenvolvida para análise e comparação dos histogramas:

1. Para a análise dos dados, primeiro é necessário realizar a importação de algumas bibliotecas(Figura 42) do qual será utilizada na aplicação,sendo essas:

- OpenCV (cv2) para processamento de imagens
- NumPy (numpy) para manipulação de arrays
- e o Matplotlib (matplotlib.pyplot) para criar gráficos e visualizações.

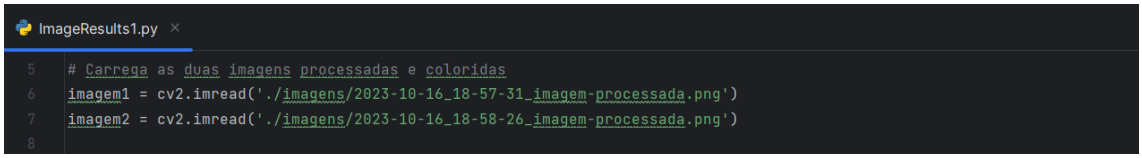
Figura 42: Importando as bibliotecas para análise dos histogramas.



Fonte: Elaborado pelo Autor

2. . Após importar as bibliotecas necessárias, é feita a leitura das duas imagens já processadas e coloridas (Figura 43).

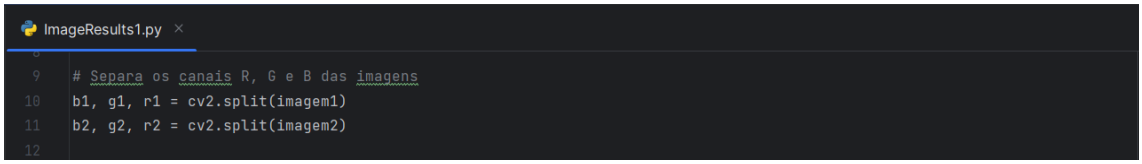
Figura 43: Realizando a leitura das imagens processadas.



Fonte: Elaborado pelo Autor

3. Após o carregamento as imagens são divididas em seus canais de cores individuais (vermelho, verde e azul) e armazenadas nas variáveis **b1, g1, r1** para a primeira imagem e **b2, g2, r2** para a segunda imagem (Figura 44).

Figura 44: Separando os canais de cores de cada imagem e armazenando em uma variável.

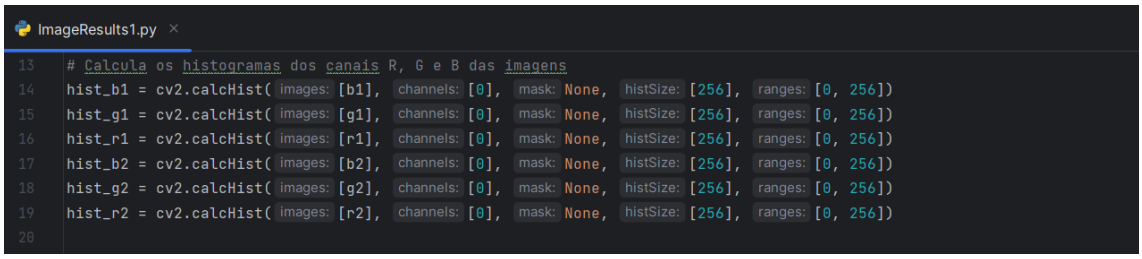


Fonte: Elaborado pelo Autor

4. Os histogramas dos canais R, G e B são calculados para ambas as imagens usando a função cv2.calcHist. Isso fornece

informações sobre a distribuição de cores em cada canal (Figura 45).

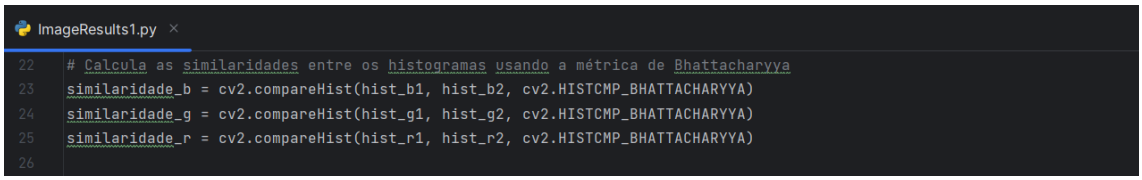
Figura 45: Cálculo dos histogramas



Fonte: Elaborado pelo Autor

5. A similaridade entre os histogramas dos canais R, G e B é calculada usando a métrica de Bhattacharyya com a função cv2.compareHist. Isso fornece uma medida de quão semelhantes são os histogramas (Figura 46).

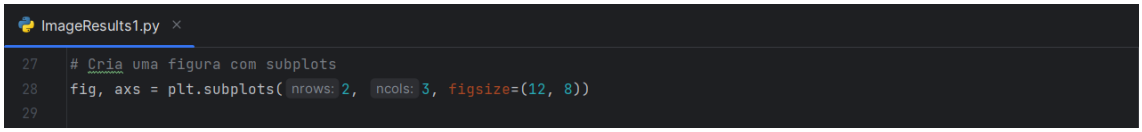
Figura 46: Cálculo da similaridade entre histogramas.



Fonte: Elaborado pelo Autor

6. Uma figura com 2 linhas e 3 colunas é criada usando plt.subplots(Figura 47).

Figura 47: Criação de uma figura com subplots.



Fonte: Elaborado pelo Autor

7. Configuração dos subplots para os histogramas (Figura 48):

- Os primeiros dois subplots (axs[0, 0] e axs[0, 1]) exibem os histogramas dos canais R, G e B das duas imagens.

- Os histogramas de cada canal são sobrepostos no mesmo gráfico.
- Os rótulos, título e legendas são configurados.

Figura 48: Configuração dos subplots para os histogramas.

ImageResults1.py

```
# Configura o primeiro subplot para o histograma
axs[0, 0].plot(hist_r1, color='r', label='R (Imagem 1)')
axs[0, 0].plot(hist_g1, color='g', label='G (Imagem 1)')
axs[0, 0].plot(hist_b1, color='b', label='B (Imagem 1)')
axs[0, 0].set_title('Histograma - Canal RGB (Imagem 1)')
axs[0, 0].set_xlabel('Bins')
axs[0, 0].set_ylabel('% dos pixels')
axs[0, 0].legend()

# Configura o segundo subplot para o histograma
axs[0, 1].plot(hist_r2, color='r', label='R (Imagem 2)')
axs[0, 1].plot(hist_g2, color='g', label='G (Imagem 2)')
axs[0, 1].plot(hist_b2, color='b', label='B (Imagem 2)')
axs[0, 1].set_title('Histograma - Canal RGB (Imagem 2)')
axs[0, 1].set_xlabel('Bins')
axs[0, 1].set_ylabel('% dos pixels')
axs[0, 1].legend()

axs[0, 2].axis('off')
```

Fonte: Elaborado pelo Autor

8. Exibição das imagens (Figura 49):
    - Os próximos dois subplots (axs[1, 0] e axs[1, 1]) exibem as imagens originais.
    - As imagens são convertidas de BGR para RGB antes da exibição.

Figura 49: Exibição das imagens:

```
# Exibe as imagens no quarto e quinto subplots
axs[1, 0].imshow(cv2.cvtColor(imagem1, cv2.COLOR_BGR2RGB))
axs[1, 0].set_title('Imagem 1')
axs[1, 0].set_xlabel('Bins')
axs[1, 0].set_ylabel('% dos pixels')

axs[1, 1].imshow(cv2.cvtColor(imagem2, cv2.COLOR_BGR2RGB))
axs[1, 1].set_title('Imagem 2')
axs[1, 1].set_xlabel('Bins')
axs[1, 1].set_ylabel('% dos pixels')
```

Fonte: Elaborado pelo Autor

9. Adição das similaridades (Figura 50):

- O último subplot (axs[1, 2]) não exibe nada, exceto o texto que mostra a similaridade entre os histogramas de cada canal.

Figura 50: Adição das similaridades.

```
# Adicione as similaridades entre os histogramas
axs[1, 2].axis('off')
axs[1, 2].text(0.5, 0.5, f'Similaridade R: {similaridade_r:.3f}\nSimilaridade G: {similaridade_g:.3f}\nSimilaridade B: {similaridade_b:.3f}',
               horizontalalignment='center', verticalalignment='center', transform=axs[1, 2].transAxes)

```

(Elaborado pelo Autor).

10. Ajuste de layout e exibição da figura (Figura 51):

- plt.tight_layout() é chamado para ajustar a disposição dos elementos na figura.
- plt.show() é usado para exibir a figura com os gráficos e as imagens.

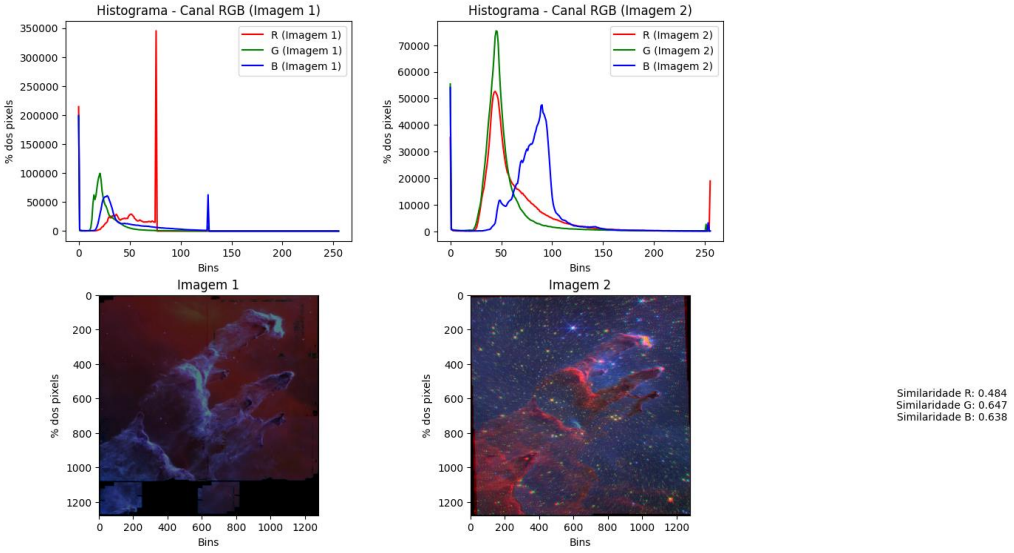
Figura 51: Ajustando o layout e exibindo a figura.

```
# Ajuste o layout
plt.tight_layout()

# Exibe a figura
plt.show()

```

Fonte: Elaborado pelo Autor

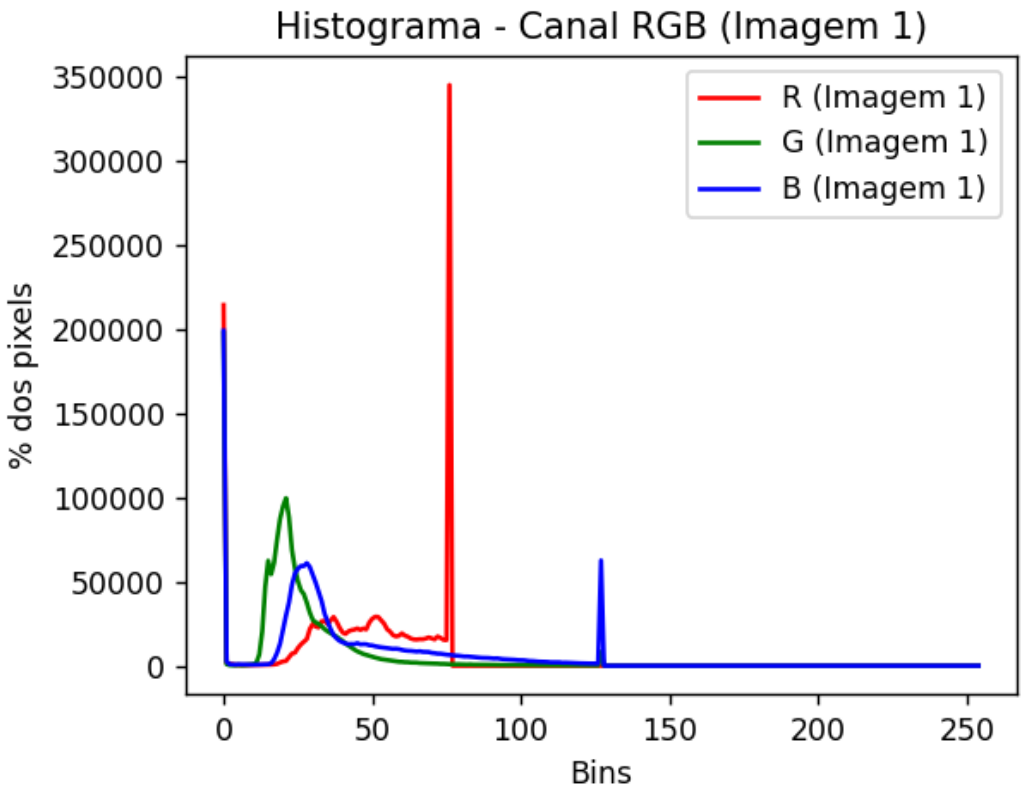
Resultados obtidos após a comparação dos histogramas:

Figura 52: Resultado das similaridades entre os histogramas.



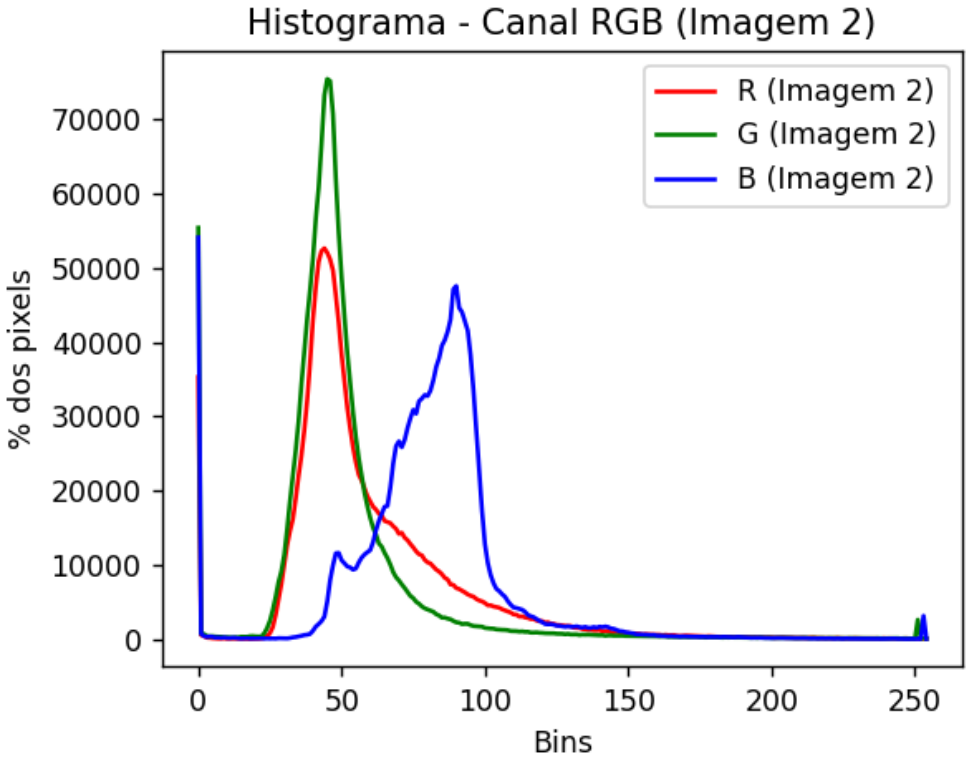
Fonte: Elaborado pelo Autor

Figura 53: Histograma dos canais RGB da primeira imagem processada.



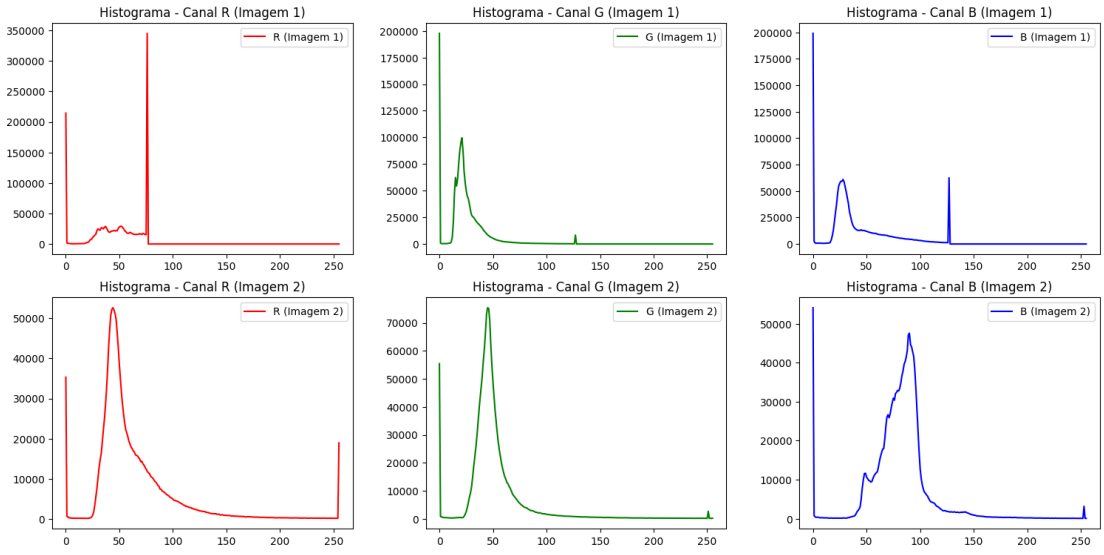
Fonte: Elaborado pelo Autor

Figura 54: Histograma dos canais RGB da segunda imagem processada.



Fonte: Elaborado pelo Autor

Figura 55: Histograma dos canais de cores RGB separados.



Fonte: Elaborado pelo Autor

Resultados obtidos:

- Canal Vermelho (R): Similaridade: 0.484
    - A similaridade moderada (0.484) no canal vermelho sugere que há algumas diferenças notáveis entre os histogramas, mas ainda existe uma quantidade significativa de características em comum nas imagens processadas com MIRICam e NIRCam. Pode haver variações nas intensidades de cores vermelhas, mas ainda há uma sobreposição considerável.

- Canal Verde (G): Similaridade: 0.647
    - A similaridade relativamente alta (0.647) no canal verde indica uma consistência mais significativa entre as imagens no que diz respeito às características representadas por esse canal. As variações no verde são menos pronunciadas do que no canal vermelho.

- Canal Azul (B): Similaridade: 0.638
    - A similaridade considerável (0.638) no canal azul sugere que as características específicas representadas pelo azul também são consistentes entre as imagens capturadas com MIRICam e NIRCam. As variações no canal azul são comparativamente menores.

4.CONCLUSÃO

O desenvolvimento deste trabalho permitiu uma melhor análise e entendimento sobre a importância do processamento de imagens na área da astronomia, onde, através do desenvolvimento de uma aplicação Python, foi possível processar e analisar a imagem da nebulosa da águia.

Como resultados, foi possível verificar imagens com informações dos arquivos FITS mais claros e precisos, sendo possível analisar as informações das imagens. Utilizando a ferramenta da comparação de histogramas, foi possível analisar as emissões de luz nos canais vermelho (R), verde (G) e azul (B) das imagens processadas.

A interpretação dos resultados obtidos poderá ser útil para compreensão dos fenômenos astronômicos na Nebulosa da Águia. A divergência do canal de cor vermelho pode ser um indicativo de regiões mais densas em material interestelar ou de processos específicos associados à formação estelar.

Enquanto isso, a similaridade nos canais verde e azul pode apontar para áreas com propriedades espectrais mais uniformes, influenciadas por fatores como densidade de gás ionizado ou presença de aglomerados estelares.

Todavia, o principal foco do trabalho foi em realizar o processamento das imagens, e espera-se que com os resultados obtidos as imagens possam ser analisadas por especialistas da área de astronomia, para que assim seja feito uma análise espectral mais detalhada dos dados ou até mesmo utilização do código para processamento de outras imagens dos cosmos.

REFERÊNCIAS BIBLIOGRÁFICAS

[01]COLLINS, Karen A. et al. AstroImageJ: image processing and photometric extraction for ultra-precise astronomical light curves. **The Astronomical Journal**, v. 153, n. 2, p. 77, 2017.

[02]GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. Editora Blucher, 2000.

[03]GUIDE, step. “Image Processing.” ESA/Webb, <https://esawebb.org/about/general/image-processing>. Acesso em 2 Abr. 2023.

[04]HANISCH, Robert J.; BRISSENDEN, Roger JV; BARNES, Jeannette. Astronomical data analysis software and systems II. **Astronomical Data Analysis Software and Systems II**, v. 52, 1993.

[05]“How Are Webb's Full-Color Images Made?” WebbTelescope.org, 14 February 2023, <https://webbtelescope.org/contents/articles/how-are-webbs-full-color-images-made>. Acesso em 2 Abr. 2023.

[06]KARTTUNEN, Hannu et al. (Ed.). **Fundamental astronomy**. Berlin, Heidelberg: Springer Berlin Heidelberg, p.4, 2007.

[07]MARAZZATO, Roberto; SPARAVIGNA, Amelia Carolina. Astronomical image processing based on fractional calculus: the AstroFracTool. **arXiv preprint arXiv:0910.4637**, 2009.

[08]PRICE-WHELAN, Adrian M. et al. The Astropy Project: sustaining and growing a community-oriented open-source project and the latest major release (v5. 0) of the core package. **The Astrophysical Journal**, v. 935, n. 2, p. 167, 2022.

[09]SHARMA, Anand Kumar. James Webb Space Telescope. **Resonance**, v. 27, n. 8, p. 1355-1369, 2022.

[10]ODENWALD, Sten. **A Guide to Smartphone Astrophotography**. Greenbelt, Maryland. Disponível em: <**https://spacemath.gsfc.nasa.gov/SMBooks/AstrophotographyV1.pdf**>. Acesso em: 30 Mar. 2023.

[11]SEDMAK, G. Image processing for astronomy. **Memorie della Societa Astronomica Italiana**, v. 57, p. 149-171, 1986.

[12]RIEKE, George H. et al. The mid-infrared instrument for the james webb space telescope, i: Introduction. **Publications of the Astronomical Society of the Pacific**, v. 127, n. 953, p. 584, 2015.

[13]WRIGHT, Gillian S. et al. The JWST MIRI instrument concept. In: **Optical, Infrared, and Millimeter Space Telescopes**. SPIE, 2004. p. 653-663.

[14]SABELHAUS, Phillip A.; DECKER, John E. An overview of the James Webb space telescope (JWST) project. **Optical, Infrared, and Millimeter Space Telescopes**, v. 5487, p. 550-563, 2004.

[15]JWST Mid Infrared Instrument - JWST User Documentation. Disponível em: <https://jwst-docs.stsci.edu>.

[16]RIEKE, Marcia J. et al. Performance of NIRCam on JWST in Flight. **Publications of the Astronomical Society of the Pacific**, v. 135, n. 1044, p. 028001, 2023.

[17]VAN DER WALT, Stefan et al. scikit-image: image processing in Python. **PeerJ**, v. 2, p. e453, 2014.

[18]HUANG, Thomas S.; SCHREIBER, William F.; TRETIAK, Oleh J. Image processing. **Proceedings of the IEEE**, v. 59, n. 11, p. 1586-1609, 1971.

[19]James Webb Space Telescope. Disponível em: <https://www.stsci.edu/jwst>.

[20]ROSEBROCK, A. OpenCV Image Histograms (cv2.calcHist). Disponível em:<https://pyimagesearch.com/2021/04/28/opencv-image-histograms-cv2-calchist/>. Acesso em: 16 out. 2023.

[21]JESUS, Edison O.; COSTA JR, Roberto. A utilização de filtros gaussianos na análise de imagens digitais. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 3, n. 1, 2015.

[22]OpenCV: Histograms. Disponível em: <https://docs.opencv.org/3.4/d6/dc7/group__imgproc__hist.html>. Acesso em: 16 out. 2023.

[23]MARQUES FILHO, Ogê; NETO, Hugo Vieira. **Processamento digital de imagens**. Brasport, 1999.