

## Extended Morphological Profile:

O código completo está dividido em duas partes, pré-processamento e código principal.

Na parte de pré-processamento existem dois scripts em python: **convert\_HS\_to\_Array.py** e **pca.py**.

- **convert\_HS\_to\_Array.py**

O script **convert\_HS\_to\_Array.py** realiza a conversão dos formatos utilizados por imagens hyperspectrais em arrays N-Dimensionais, sendo utilizado da seguinte forma:

Argumentos de entrada:

image - Imagem de entrada N-Dimensional (ENVI extension) [.img]  
header - Cabeçalho da imagem de entrada [.hdr]  
output - Diretório de saída -

Saída:

image2 - Matriz N-Dimensional salva no formato [.npz]

### Exemplo de utilização:

```
python convert_HS_to_Array.py input/TelopsDatasetCityLWIR_Subset.img  
input/TelopsDatasetCityLWIR_Subset.hdr output/
```

- **pca.py**

O script **pca.py** realiza a extração das M primeiras principais componentes do método de redução de dimensionalidade PCA.

Argumentos de entrada:

image - Matriz N-Dimensional no formato [.npy].  
num\_components - Número de componentes principais desejado.  
output - Diretório de saída

Saída:

image2 - Matriz M-Dimensional salva no formato [.npz]

### Exemplo de utilização, para a extração das 3 primeiras componentes do exemplo acima:

```
python pca.py output/TelopsDatasetCityLWIR_Subset.img.npy 3 output/
```

O código principal apresentado não foi modularizado, logo todas as funções e declarações utilizadas estão dentro do script **emp.py**

Argumentos de entrada:

data	- Matriz N-Dimensional no formato [.npy]
se_size	- Tamanho do elemento estruturante
se_size_increment	- Incremento do tamanho do elemento estruturante
num_openings_closings(K)	- Número de aberturas e fechamentos
output	- Diretório de saída

Saída:

image2 - Matriz  $N \times (2 \times K + 1)$ -Dimensional salva no formato [.npy]

**Exemplo de utilização, para a criação do perfil morfológico extendido das 3 componentes principais extraídas anteriormente:**

```
python emp.py preprocess/output/PC_TelopsDatasetCityLWIR_Subset.img.npy 2 3 5 output/
```

## **BIBLIOTECAS UTILIZADAS:**

**scikit-image:** Image processing in Python<sup>1</sup>

Instalação na plataforma linux:

```
easy_install -U scikit-image
```

**NumPy** - Fundamental package for scientific computing with Python<sup>2</sup>

Instalação na plataforma linux:

```
sudo apt-get install python-numpy python-scipy
```

**Spectral Python (SPy)** is a pure Python module for processing hyperspectral image data.<sup>3</sup>

Instalação na plataforma linux:

```
easy_install spectral
```

## **DATASET UTILIZADO:**

Para realização de testes foi utilizado uma imagem hyperspectral de 84 bandas (TelopsDatasetCityLWIR\_Subset.img) contida na base de dados do Data Fusion Contest 2014. Para maiores informações sobre a base de dados utilizada, segue em anexo a documentação da mesma.

## **Referências:**

<sup>1</sup> <http://scikit-image.org/>

<sup>2</sup> <http://www.numpy.org/>

<sup>3</sup> <http://www.spectralpython.net/>

**Paper do extended morphological profile:** Classification of Hyperspectral Data From Urban Areas Based on Extended Morphological Profiles Jón Atli Benediktsson, Fellow, IEEE, Jón Aevar Palmason, Student Member, IEEE, and Johannes R. Sveinsson, Senior Member, IEEE