

# Paralelización en GPU para la maximización de la ganancia espectral

## Manual de Instalación

### Tabla de contenido

1	Descripción.....	2
2	Requisitos.....	2
3	Instalación .....	2
3.1	Instalar la tarjeta gráfica.....	2
3.2	Instalar CUDA y PyCUDA.....	3
3.3	Clonar repositorio .....	3
3.4	Instalar librerías de Python .....	3
3.5	Iniciar aplicación .....	3

# 1 Descripción

Esta aplicación permite la ejecución de la transformada Parallel Fusimagery a la cual se le pasan cuatro parámetros siendo los dos primeros las imágenes multiespectral y pancromática, las cuales por medio de un tratamiento digital de imagen se unirán para obtener una imagen satelital de mayor calidad, el tercer parámetro es el nivel y el cuarto el nombre de la imagen de salida. Esta aplicación ya ha sido implementada de forma secuencial, por ello en esta versión se realiza la paralelización de la transformada con el objetivo de optimizar los tiempos de respuesta.

## 2 Requisitos

Para el funcionamiento de este software, es necesario el siguiente hardware:

1. Tarjeta gráfica: NVIDIA Tesla K80

También es necesario el siguiente software:

1. Python 2.7: <https://www.python.org/download/releases/2.7/>
2. Numpy 1.16.0: <https://pypi.org/project/numpy/>
3. Pillow 5.3.0: <https://pillow.readthedocs.io/en/latest/installation.html#basic-installation>
4. Matplotlib 3.0.2: <https://matplotlib.org/users/installing.html>
5. Pycuda 2018.1.1: <https://pypi.org/project/pycuda/>
6. Skcuda 0.5.2: <https://scikit-cuda.readthedocs.io/en/latest/install.html>

## 3 Instalación

Para llevar a cabo la instalación, tenga en cuenta que es necesario que ya tenga instalado Python versión 2.7, de lo contrario, no será posible llevar a cabo los siguientes pasos. Cuando se instala Python, tendrá instalado PyPI (Python Package Index) por defecto, que es el que permite instalar paquetes de Python, con el comando *pip*.

### 3.1 Instalar la tarjeta gráfica

Primero se debe verificar si el sistema operativo ya reconoce la tarjeta gráfica, para eso se digita el siguiente comando y se verifica que la tarjeta gráfica NVIDIA Tesla K80 se encuentre en la lista:

```
lisci | grep -i nvidia
```

En caso de que no aparezca se digita el siguiente comando para actualizar los controladores conectados al computador

```
sudo update-pciids
```

Comprobando que la tarjeta gráfica ya se encuentra instalada se procede a realizar la instalación de los drivers por medio de los siguientes comandos:

```
sudo apt-add-repository ppa:xorg-edgers/ppa
sudo apt-add-repository ppa:ubuntu-x-swat/x-updates
sudo apt-get update
sudo apt-get install nvidia-current nvidia-settings
```

### 3.2 Instalar CUDA y PyCUDA

Para correr la aplicación es necesario instalar CUDA y PyCUDA, el primero se instala por medio de los siguientes dos comandos:

```
sudo apt-get install cuda
sudo apt-get install nvidia-cuda-toolkit
```

Con esto instalado se procede a realizar la configuración de dos variables de entorno, la primera PATH y la segunda CUDA\_ROOT de la siguiente forma:

```
export PATH=/usr/local/cuda/bin:$PATH
export CUDA_ROOT=/usr/local/cuda
```

Con estas dos variables de entorno se podrá ejecutar el siguiente comando para instalar PyCUDA:

```
pip install pycuda
```

### 3.3 Clonar repositorio

Como primer paso, clone el repositorio de GitHub en su equipo.

```
git clone https://github.com/dataScienceAndres29/fusion-wavelet-transform-GPU.git
```

### 3.4 Instalar librerías de Python

Con el comando *pip* debe instalar las librerías de python, de la siguiente forma:

```
pip install numpy
```

```
pip install pillow
```

```
pip install matplotlib
```

```
pip install pycuda
```

```
pip install scikit-cuda
```

### 3.5 Iniciar aplicación

Para iniciar la aplicación, ubique su carpeta a través de la consola. Ahora, utilice el siguiente comando para realizar la ejecución de la aplicación en CPU:

```
python fusion_cpu.py paramatro1 parametro2 parametro3 salida
```

Y el siguiente comando para realizar la ejecución de la aplicación en GPU:

```
python fusion_gpu.py paramatro1 parametro2 parametro3 salida
```

Como se puede evidenciar tienen la misma cantidad de parámetros, a continuación se hace la descripción de cada uno de ellos

- parametro1: String con la ruta absoluta de la imagen multiespectral a utilizar de tamaño  $A \times A$  donde A es un número diádico, es decir,  $2^n$ .
- parametro2: String con la ruta absoluta de la imagen pancromática a utilizar de igual tamaño que la multiespectral.
- parametro3: Entero con el nivel, es decir, la cantidad de veces que se aplicará la transformada.
- salida: Nombre de la imagen de salida después de realizar la ejecución de la aplicación.