МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра: Программной инженерии**

Направление подготовки: «Программная инженерия»

«Обработка изображений»

**Отчёт по лабораторной работе**

|  |  |
| --- | --- |
|  | Выполнил:  студент группы 381908-3  Имя  Ермаков Павел Андреевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Проверила:  Гетманская А.А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Нижний Новгород  
2021 г.

Содержание

[Задача 3](#_Toc88545578)

[Работа 3](#_Toc88545579)

[Результаты и сравнения 4](#_Toc88545580)

[Код main.py 6](#_Toc88545581)

[Вывод 8](#_Toc88545582)

# Задача

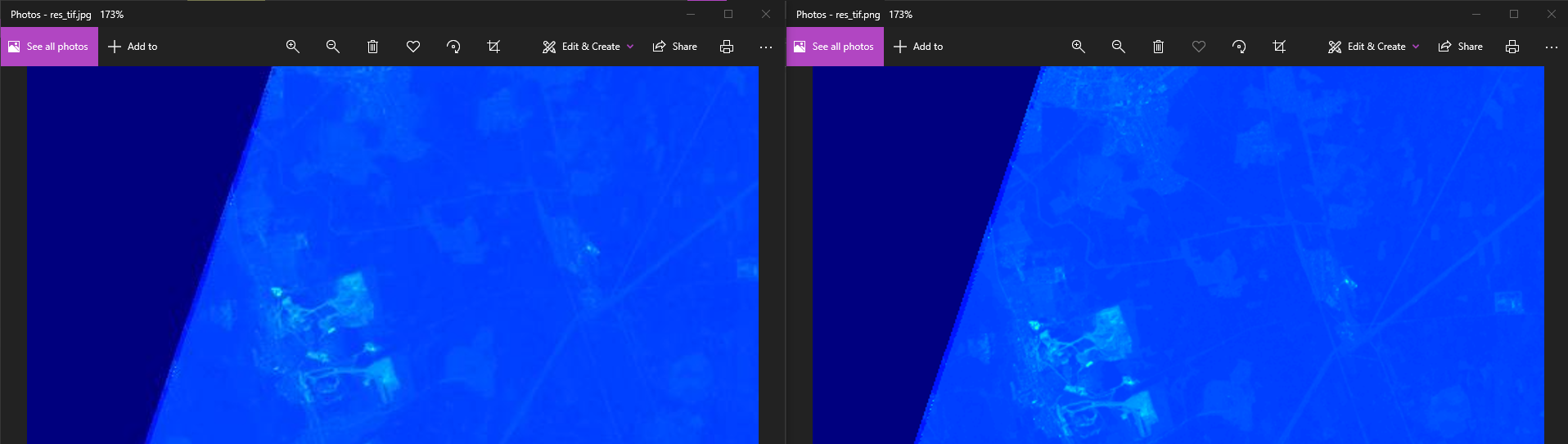
Объединить tif файлы из присланного архива, а затем конвертировать в RGB.

# Работа

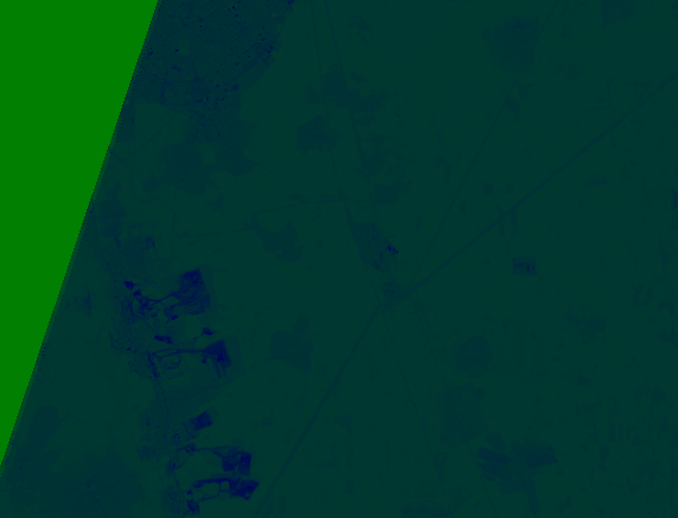
В данной работе я смог объединить .tif файлы в один и его дальнейшая конвертация в .png. После конвертации в .png с использованием colormaps из Matplotlib. Информация про colormaps была взята с данного [сайта](https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html). В работе были проверены разные colormaps, а также типы конвертации изображения. Конвертация в .jpg не изменила разрешение изображения, но значительно ухудшила его качество и уменьшила размер на 97%. Конвертация в .png не изменила разрешение, а также не потеряла в качестве; изначальный размер уменьшился на 90%. Для дальнейшей работы был выбран тип .png.

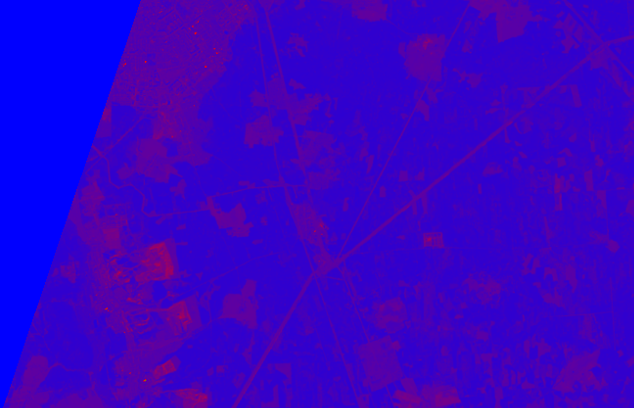
# Результаты и сравнения

На данном изображение приведено сравнение .jpg и .png с использованием colormap «jet». Изображение увеличено до 173%. Отчетливо можем видеть разницу в качестве изображений. У .jpg больше шумов, меньше четкость, чем у .png



Пример использования colormap «ocean»



Пример использования colormap «brg»

Пример использования colormap «coolhot»



# Код main.py

import glob

import numpy as np

import rasterio

from rasterio.merge import merge

import os

import matplotlib.image as mplimg

import skimage.io

def merge\_tif(path,ftype,outfile, resfolder):

    alltif=glob.glob(os.path.join(path,'\*.%s'%ftype))

    bigtif=[]

    for tif in alltif:

        src=rasterio.open(tif)

        bigtif.append(src)

        print(tif)

    mosaic, out\_trans = merge(bigtif)

    outmeta=src.meta.copy()

    outmeta.update({'driver':'GTiff','height':mosaic.shape[1],'width':mosaic.shape[2],'transform':out\_trans})

    os.makedirs(path+'\\merged')

    outpath=path+'\\merged\%s.tif'%outfile

    with rasterio.open(outpath,'w',\*\*outmeta) as dest:

        dest.write(mosaic)

def to\_png(foldername,colourmap, resfolder):

    files = []

    for file in sorted(os.listdir(foldername)):

        files.append(file)

    if not os.path.exists(resfolder +'RES'):

        os.makedirs(resfolder +'RES')

    \_min=1E8

    \_max=-1E6

    filemax=''

    filemin=''

    for filenow in files:

        img = skimage.io.imread(foldername+filenow, plugin='tifffile')

        img\_arr = np.array(img)

        if np.max(img\_arr)>\_max:

            \_max=np.max(img\_arr)

            filemax=filenow

        if np.min(img\_arr)<\_min:

            \_min=np.min(img\_arr)

            filemin=filenow

    for filenow in files:

        print('PROCESSING '+filenow+' right now')

        img = skimage.io.imread(foldername+filenow, plugin='tifffile')

        img\_arr = np.array(img)

        mplimg.imsave(resfolder+'RES/'+filenow+'.png',img\_arr,cmap=colourmap,vmin=\_min,vmax=\_max)

path=r'./tiffiles/'

resfolder = r'./'

outfile='res\_tif'

ftype='tif'

colourmap = 'coolwarm'

merge\_tif(path,ftype,outfile)

to\_png(path, colourmap,resfolder)

# Вывод

По результатам проведенной работы я понял как работать с .tif файлами, узнал про colormap. Понял на практике, что .png лучше .jpg.