Методы ускорения классификации больших объемов данных ДЗЗ

Авраменко Ю.В., Фёдоров Р.К.
avramenko@icc.ru, fedorov@icc.ru
ИДСТУ СО РАН

ИКИ РАН
Москва, 2022

Актуальность

Оперативная обработка мультиспектральных снимков земной поверхности имеет важное значение в принятии решений по развитию региона в лесном, сельском, градостроительном, экологическом и других направлениях.

Список реализованных Web-сервисов

Open Geospatial Consortium, Inc. (OGC) Web Processing Service (WPS)

- Получения данных SRTM
- Расчета Elevation, Slope, Aspect на основе SRTM
- Расчета спектральных индексов
- Классификации методом SVM и CNN
- Обучения SVM и CNN
- Расчета временной доступности географических объектов
- Расчета загрязнения парниковыми газами для определенной области
- Создания shape файлов из табличных данных
- Растеризации shape файлов
- Получения снимков Landsat 7-8, Sentinel-2
- Поиска объектов по запросу на языке SOQL
- Инструментов растровой алгебры

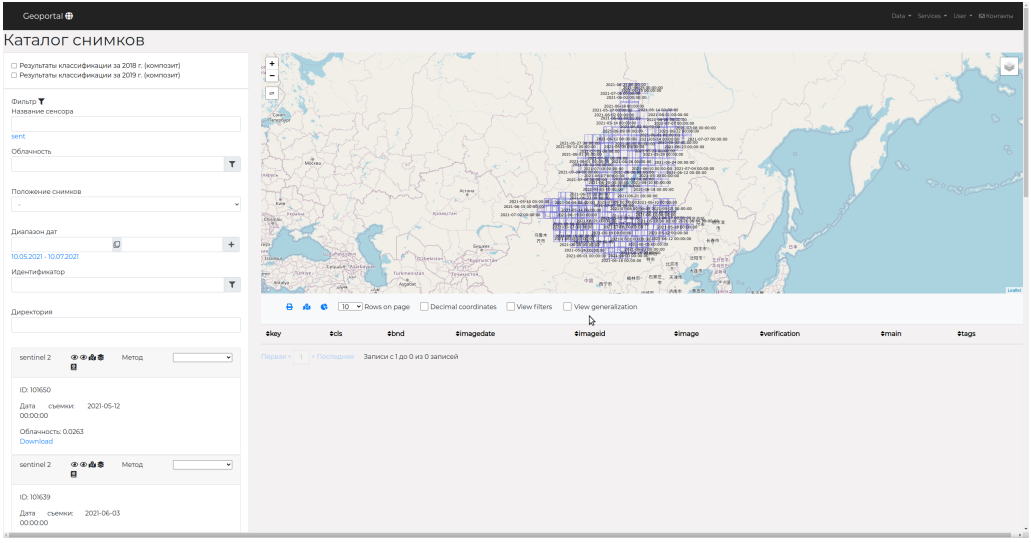
Текущие состояние работы

В ИДСТУ СО РАН имеется обновляемый каталог космоснимков Sentinel-2 на территорию Иркутской области с 2018 года.

Список задач:

- Классификация типов земельного покрова - выполнено
- Подстчет статистических данных - в процессе
- Анализ временной серии космоснимков - в планах

Каталог


Geoportal
Data
Services
User
Контакты
Каталог снимков
Результаты классификации за 2018 г. (композит)
Результаты классификации за 2019 г. (композит)
Фильтр
Название сенсора
sent
Облачность
Положение снимков
Диапазон дат
10.05.2021 - 10.07.2021
Идентификатор
Директория
sentinel 2
Метод
ID: 101650
Дата съемки: 2021-05-12 00:00:00
Облачность: 0.0263
Download
sentinel 2
Метод
ID: 101639
Дата съемки: 2021-06-03 00:00:00
10
Rows on page
Decimal coordinates
View filters
View generalization
$key
$cls
$bnd
$imagedate
$imageid
$image
$verification
$main
$tags
Первая
Последняя
Записи с 1 до 0 из 0 записей

Используемые технологии

Nextcloud

docker

jupyterhub

kubernetes

Существующие подходы

Google Colab, Google Earth Engine

- Проблемы передачи данных
- Ограниченность вычислительных мощностей
- Скорость обработки данных
- Использование собственных форматов данных

Аппаратная часть оптимизации вычислений

Была создана собственная система на подобная Google Colab.

- Чтение данных - заменена система хранения.
- Обучении нейросети - приобретены графические станции.

Программная часть оптимизации вычислений

Были проанализированы слабые места библиотеки FastAI

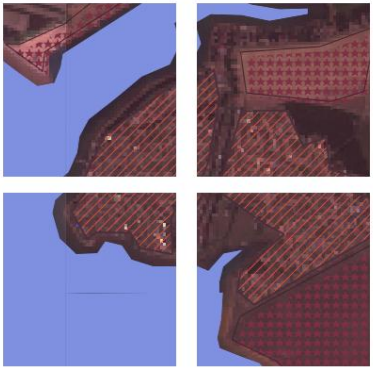
- Реализована процедура подготовки файлов.
- Разработана структура файлов.
- Реализована процедура чтения файлов.
- Разработан и реализован обработчик серии космоснимков.
- Ускорение классификации одного изображения.
- Произведена оптимизация вычислений.

Процедура подготовки файлов

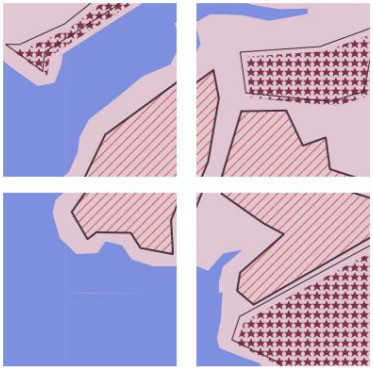
```
def change_resolution(i):
        subprocess.run(['gdalwarp', '-ts', str(i[0]), str(i[1]), i[2] + '/' + i[3], i[4]])

for i in range(len(bands)):
        fname = imagefolder + filenames[i]
        rasterfile = fulllmagepath + '/' +bands[i]
        if os.path.exists(fname):
                if i in [0,4,5,6,8,9,10,11,12]:
                        tmpfile=imagefolder + os.path.basename(bands[i])
                        bands_list.append([height, width, fulllmagepath, bands[i], tmpfile])

with Pool() as p:
        p.map(change_resollution, bands_list)
```

Структура файлов

Исходный снимок Sentinel-2 (10980x10980) был разбит на части размером 64x64 в соответствии с маской и сохранен в формате npy.



Исходное изображение



Маска

Процедура чтения файлов

```
class SentinelList(ImageList):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
    @classmethod
    def from_slmage(cls, slmage, step_size):.
        cls.slmage = slmage
        newitems=[]
        cell_size=64
        height, width, channels = slmage.shape.......
        center_shift= step_size // 2.
        for i in range(round((width - cell_size) / step_size)):
            for j in range(round((height -cell_size) / step_size)):
                x=i*step_size
                y=j*step_size
                cell = (x,x+cell_size,y,y+cell_size)
                newitems.append(cell)
        cls.slmageCnt = len(newitems)
        cls.imgitems=newitems
        return cls(items=newitems)
    def label_from_array(self, array, label_cls=None, **kwargs):
        return self._label_from_list(array,label_cls=label_cls,**kwargs)....
    def get(self, i):
        cell = self.imgitems[i]
        ROI=self.slmage[ cell[0]:cell[1], cell[2]:cell[3],:]
        ROI = np.swapaxes(ROI, 0, 2).astype(np.float32)
        return ROI
```

Обработка серии космоснимков

```
check_iter = True
while check_iter:
        file_lock = 'mass.txt.lock'
        file_name = 'mass.txt'
        lock = filelock.FileLock(file_lock)
        with lock:
                try:
                        f = open(file_name, 'rb')
                        mass = pickle.load(f)
                        f.close()
                        mas = next(mass)
                        print (mas)
                        f = open(file_name, 'wb')
                        pickle.dump(mass, f)
                        f.close()
                except StopIteration:
                        check_iter = False
                finally:
                        if lock.is_locked:
                                print('UnLock')
                                lock.release()
                        time.sleep(5)
```

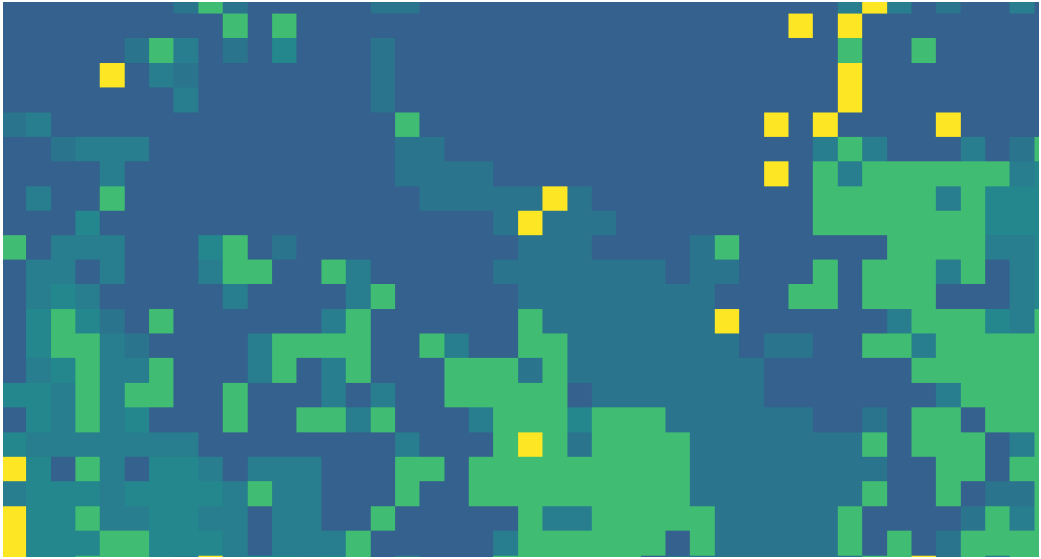
Ускорение классификации одного изображения

```
steps = [64,32,16,8,4,2]
for item in steps:
        step_size = item
        data2 = imd.get_sentinel_image_list(image2, step_size, marks)
        learn = Learner(data2, model, metrics=[accuracy]).mixup()
        learn.load('/model_120e')
        start_time = time.time()
        preds, values = learn.get_preds(ds_type=0)
        res = imd.draw_preds_v2(preds, step_size, learn.data, image2, res)
        res, marks = imd.re_draw_preds(preds, step_size, learn.data, res)
        imd.save_res(item + '_out.tif', GeoTransform2, Projection2, res)
```

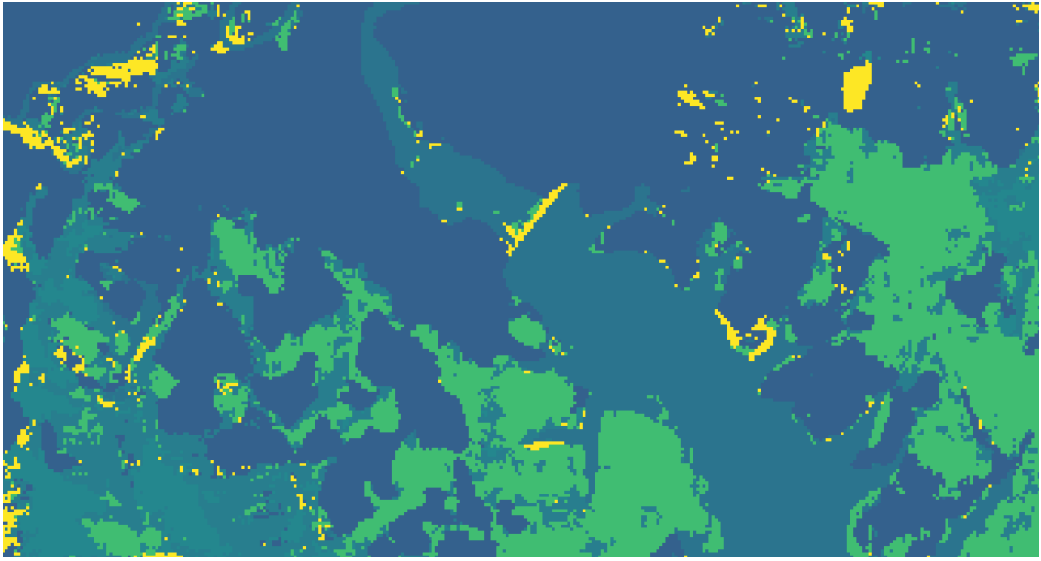
Оптимизация вычислений

```
@njit()
def calc_frequency(data):
        bands, rows, cols = data.shape
        res = np.zeros((rows, cols), dtype=np.uint8)
        for i in prange(rows):
                for j in prange(cols):
                        elements = np.arange(27, dtype=np.uint8)
                        val_gist = np.zeros(27, dtype=np.uint8)
                        for el in range(len(data[:,i,j])-1, -1, -1):
                                val_gist[data[el,i,j]] += 1
                        max_ind = -1
                        max_c = 0
                        for el in range(len(data[:,i,j])-1, -1, -1):
                                if (val_gist[el]!= 0) and (val_gist[el] != 26) and (val_gist[el] > max_c)
                                        max_c = val_gist[el]
                                        max_ind = el
                        for p in range(len(val_gist)-1, -1, -1):
                                for q in range(p):
                                        if val_gist[q] > val_gist[q + 1]:
                                                val_gist[q], val_gist[q + 1] = val_gist[q + 1], val_gist[
                                                elements[q], elements[q + 1] = elements[q + 1], elements[
                        cur_len = len(elements)
                        while cur_len > 1:
                                if (elements[cur_len-1] == 0) or (elements[cur_len-1] == 26):
                                        cur_len -= 1
                                else:
                                        break
                        res[i,j] = elements[cur_len-1]
        return res
```

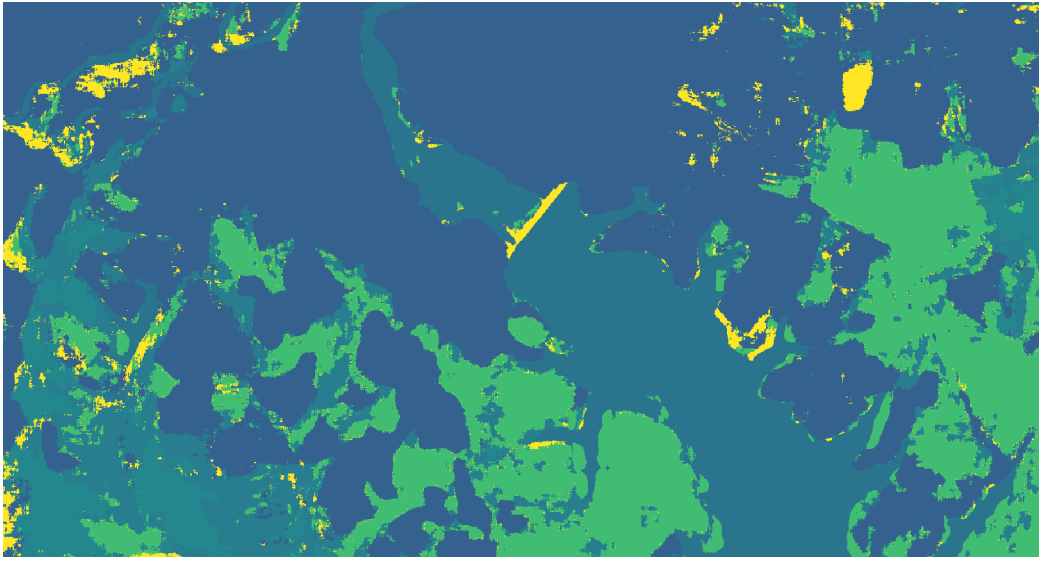
Промежутоный результат классификации



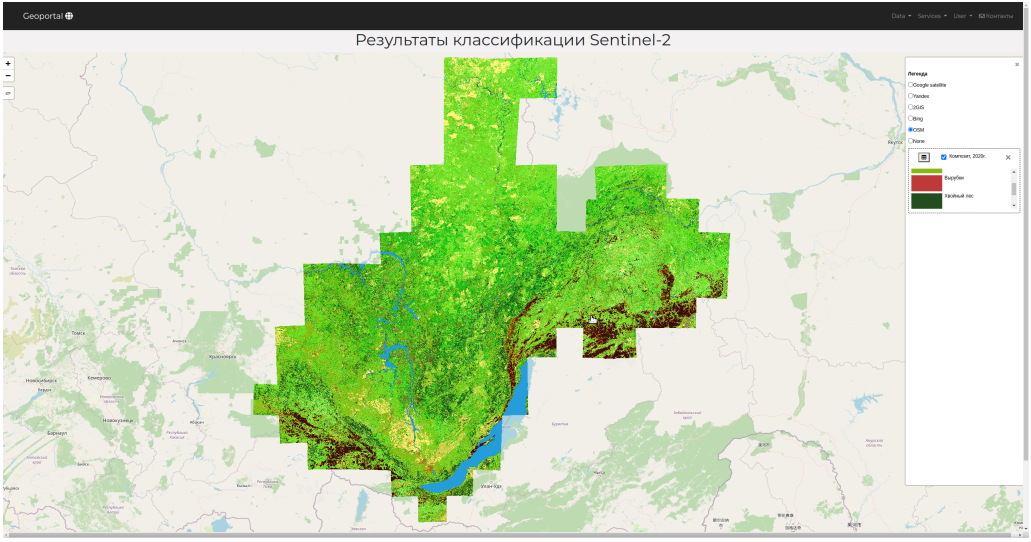
Промежутоный результат классификации



Промежутоный результат классификации



Карта земельного покрова



Заключение

- Обработана 21 000 космоснимков.
- Построена карта земельного покрова на территорию Иркутской области. Карта свободна от облачности и повежденных данных.
  Результат классификации доступен по адресу http://cris.icc.ru/classresults

Спасибо за внимание!