Распознавание сгоревших территорий с помощью деревьев решений и OpenCV

Обсудить в форуме Комментариев — 7

Использование OpenCV для работы со снимками MODIS и их классификации.

Лето 2010 года наглядно показало, что для нас с вами очень актуальна проблема природных пожаров. Для устранения их последствий необходимо знать, на каких именно территориях они произошли, знать их точное пространственное расположение. Один из перспективных подходов к решению этой задачи — классификация территорий с помощью деревьев решений по данным MODIS – Land. Для эффективной обработки растровых данных и их классификации хорошо подходит библиотека компьютерного зрения OpenCV (Open Source Computer Vision). В ней эффективно реализована необходимая нам функциональность, библиотека бесплатная, свободная (лицензия BSD), кросс-платформенная и является фактически стандартом в компьютерном зрении — количество её скачиваний превысило 2 миллиона.

В этой статье мы по шагам рассмотрим как можно использовать библиотеку OpenCV для распознавания сгоревших территорий. Для примера мы будем использовать данные MODIS на начало и конец лета 2010 года представленными двумя продуктами:

- MOD09A1 — 8-дневные композиты, содержащие 7 каналов отражающей способности земной поверхности (Surface Reflectance Bands 1–7);
- MCD45A1 — информация о сгоревшей территории с указанием даты пожаров для каждого пикселя местности.

Оба этих продукта имеют разрешение 500 метров на один пиксель местности. В качестве тренировочной выборки мы возьмём данные по республике Марий-Эл, а в качестве тестовой — данные по Нижегородской области.

Содержание

Импорт данных MODIS

Прежде всего нужно перевести данные MODIS из формата HDF в GeoTIFF — про это подробно написано в статье «Импорт продуктов MODIS уровней 2G, 3, 4 с помощью MRT». После этого мы сможем загрузить данные в OpenCV как обычное изображение в формате GeoTIFF. Поскольку изображение — это набор яркостей пикселей, то в OpenCV он будет хранится как матрица. Для этого используется класс Mat — главный класс для работы с матрицами и изображениями в OpenCV:

```
Mat modisBand = imread("sur_refl_b01.tif", CV_LOAD_IMAGE_UNCHANGED);
//читаем изображение из файла "sur_refl_b01.tif" и записываем его в матрицу modisBand
```

Подгрузив аналогично 7 каналов MODIS на начало лета и 7 каналов на конец лета, мы можем собрать все данные в одну 14-канальную матрицу, то есть прямоугольную матрицу элементами которой являются вектора из 14 элементов:

```
vector<Mat> allBands;
allBands.push_back(modisBand);
... //подгружаем по очереди все каналы и добавляем их в allBands
Mat data;
merge(allBands, data); //объединяем все каналы в одну матрицу data
```

Для обучения деревьев решений и валидации полученных результатов нам необходима информация о том, какие же территории действительно сгорели за это лето. С помощью OpenCV мы легко сможем получить эту информацию из продукта MCD45A1, используя логические операции между матрицами и скалярами, которые выполняются попиксельно:

```
Mat burnDate = imread(burnDateFilename, CV_LOAD_IMAGE_UNCHANGED);
//элемент матрицы burnDate содержат информацию о дате пожара, произошедшего в
соответствующем пикселе местности
Mat unburnedArea = (burnDate == 0);
//элемент матрицы unburnedArea будет равен 255, если пиксель не сгорел (дата пожара
равна нулю), или 0 в противном случае.
uint16_t minBurnDay = 153; //2 июня
uint16_t maxBurnDay = 248; //5 сентября
Mat burnedArea = (burnDate >= minBurnDay) & (burnDate <= maxBurnDay);
//элемент матрицы burnedArea будет равен 255, если в нём был пожар в указанном
диапазоне дат, или 0 в противном случае
```

Обучение деревьев решений

В качестве классификатора можно использовать простой и популярный подход — дерево решений. Но более точные результаты можно попробовать получить, если использовать ансамбль деревьев решений (Random Forest), который обобщает народную мудрость «одна голова хорошо, а две лучше». Сейчас данные загружены в виде 14-канальной матрицы. Для тренировки деревьев решений нам нужна 1-канальная матрица, в которой каждая строка — это один элемент тренировочной выборки, а в столбцах хранятся числовые значения его признаков (соответствующих каналов). То есть нам нужно преобразовать исходную матрицу в 1-канальную матрицу из 14 столбцов, в которой строка соответствует данным MODIS для одного пикселя местности:

```
Mat trainData = data.reshape(1, data.total());
//сделать из data 1-канальную матрицу с числом строк, равным числу пикселей в data
Mat responses = burnedArea.reshape(1, burnedArea.total());
//аналогично поступаем с метками классов
```

Теперь можем запустить тренировку ансамбля деревьев решений:

```
CvRTrees classifier;
classifier.train(trainData, CV_ROW_SAMPLE, responses);
//CV_ROW_SAMPLE указывает, что элементы выборки для классификации хранятся по строкам
```

Совершенно аналогично мы можем натренировать обычное дерево решений для сравнения:

```
CvDTree decisionTree;
decisionTree.train(trainData, CV_ROW_SAMPLE, responses);
```

Распознавание

Следующий шаг — классификация каждого пикселя местности тестовых данных с помощью натренированного ансамбля деревьев решений:

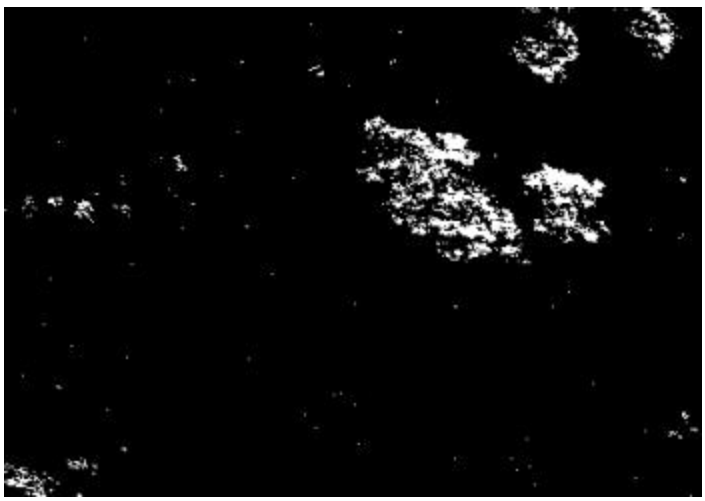
```
int rowsCount = testData.rows;
//количество строк в матрице testData, равное количеству классифицируемых пикселей местности
Mat prediction = Mat(rowsCount, 1, CV_32SC1);
//создаём матрицу размера rowsCount x 1 и типом элемента int
for(int i=0; i<rowsCount; i++)
{
  Mat sample = testData.row(i);
  //извлекаем i-ую строчку из матрицы testData - значения 14 каналов MODIS для соответствующего пикселя местности
  prediction.at<int>(i, 0) = classifier.predict(sample);
  //записываем результат классификации в элемент (i, 0) матрицы prediction
}
```

Пост-обработка

Деревья решений классифицируют каждый пиксель независимо друг от друга. Это приводит к тому, что хотя все соседи пикселя классифицированы как сгоревшие, он сам может может быть классифицирован как несгоревший — для реального распространения пожара это маловероятно. Для учёта результатов классификации соседних пикселей можно использовать операции математической морфологии:

```
dilate(predictedBurnedArea, predictedBurnedArea, Mat(), Point(-1, -1), 5); //применяем наращивание 5 раз. Используется структурирующий элемент по умолчанию - квадратный элемент 3x3 erode(predictedBurnedArea, predictedBurnedArea, Mat(), Point(-1, -1), 7); //применяем эрозию 7 раз dilate(predictedBurnedArea, predictedBurnedArea, Mat(), Point(-1, -1), 3); //применяем наращивание 3 раза
```

Результаты обработки (белым цветом показаны сгоревшие территории):



до морфологии

после морфологии

Валидация

Для оценки качества классификации воспользуемся стандартным подходом — подсчитаем матрицу ошибок (confusion matrix). Элемент (i, j) этой матрицы равен количеству объектов класса i, классифицированных как класс j. Матрицу ошибок можно посчитать, имея матрицы с реальными значениями классов (groundTruth) и с предсказанными значениями (prediction):

```
int classesCount = 2; //количество распознаваемых классов
Mat confusionMatrix(classesCount, classesCount, CV_32SC1);
//создаём матрицу размера classesCount x classesCount типа int
for(int i=0; i<classesCount; i++)
{
```

```
  for(int j=0; j<classesCount; j++)
  {
    confusionMatrix.at<int>(i, j) = countNonZero((groundTruth == i) & (prediction ==
j));
    //подсчитываем количество элементов класса i, классифицированных как j
  }
}
```

Визуализация

Каналам RGB соответствуют каналы 1, 3 и 4 спутника MODIS. Для создания снимка нужно объединить эти каналы в одно изображение. Для повышения качества полученного изображения с ним нужно сделать несложные преобразования, описанные в статье «Creating Reprojected True Color MODIS Images: A Tutorial». После этого отрисуем контуры сгоревших территорий поверх полученного изображения:

```
vector<vector<Point>> contours;
//массив для найденных контуров
findContours(burnedArea, contours, CV_RETR_LIST, CV_CHAIN_APPROX_NONE);
//найти контура по маске сгоревших территорий
drawContours(trueColorImage, contours, -1, Scalar(0, 0, 255), 2);
//нарисовать все контуры красной линией толщины 2
namedWindow("burned area contours", CV_WINDOW_NORMAL);
//создать окно с именем "burned area contours"
imshow("burned area contours", trueColorImage);
//отобразить изображение в созданном окне
waitKey();
//отображать изображение, пока пользователь не нажмёт "any key"
```

Экспорт результатов

Для экспорта результатов в GeoTIFF сначала сохраним полученное изображение в формате TIFF:

```
imwrite("burnedArea.tif", trueColorImage);
```

Теперь сохранённое изображение нужно связать с географическими координатами. Для этого есть несколько способов, в том числе есть и хорошие решения — кросс-платформенные, бесплатные и с открытым исходным кодом. Например, можно использовать библиотеку GDAL — про это подробно написано в статье «Использование GDAL для привязки растровых материалов».

Результаты экспериментов

Для повышения точности классификации помимо описанных шагов лучше добавить ещё несколько дополнительных фильтров (например, фильтрация данных по качеству и тренировка классификатора только на надёжных данных). Полную версию алгоритма с данными и исходным кодом можно скачать здесь. Теперь мы можем визуально оценить результаты работы алгоритма:

Количественную оценку дают построенные матрицы ошибок:

Матрица ошибок для одного обычного дерева решений

		Предсказанный класс территории	
		Несгоревшая	Сгоревшая
Реальный класс территории	Несгоревшая	109423	17387
	Сгоревшая	60	3372

Матрица ошибок для ансамбля деревьев решений

		Предсказанный класс территории	
		Несгоревшая	Сгоревшая
Реальный класс территории	Несгоревшая	120367	6443
	Сгоревшая	172	3260

Рассчитаем ошибки омиссии и комиссии:

		Классификатор	
		Дерево решений	Ансамбль деревьев решений
Ошибка	омиссии	0.02	0.05
	комиссии	0.84	0.66

Используя ансамбль деревьев решений, мы нашли 95% случившихся пожаров, но при этом 66% территорий, классифицированных как сгоревшие, относятся к несгоревшим по данным MCD45, которые в свою очередь отличаются большой ошибкой омиссии («Данные о сгоревших площадях MCD45: описание и получение»).

Не менее важной характеристикой алгоритма является время его работы. На Intel Core i7 960 @ 3.20GHz с использованием только одного потока время тренировки ансамбля деревьев решений составило 0.85 секунды (54000 пикселей), время обработки тестовой территории — 0.13 секунды (130 000 пикселей).

Таким образом, используя стандартные средства OpenCV, мы смогли легко получить простой и быстрый алгоритм для распознавания сгоревших территорий.

Ссылки по теме

- Библиотека компьютерного зрения OpenCV
- Данные о сгоревших площадях MCD45: описание и получение
- Мониторинг пожаров на природных территориях с помощью сервиса FIRMS

Обсудить в форуме Комментариев — 7

Последнее обновление: 2014-05-15 00:48

Дата создания: 31.03.2011
Автор(ы): Илья Лысенков (инженер по программному обеспечению, Itseez)