

# Классификация растровых данных при помощи DTclassifier для QGIS

[Обсудить в форуме](#) Комментариев — 87

Эта страница опубликована в основном списке статей сайта  
по адресу <http://gis-lab.info/qa/dtclassifier.html>

Описание и пример использования нового инструмента классификации растров.

Тематическая классификация растров с обучением является довольно распространенной задачей. Типичная классификация использует на входе набор растров, чаще всего данных дистанционного зондирования и набор тренировочных территорий, определяющих искомые и все остальные (фоновые) объекты. После сбора этой информации методики собственно построения модели классификации могут быть разными, параметрическими и непараметрическими. Первые подразумевают наличие некоторого статистического распределения в тренировочных данных (например, нормального), последние таких допущений не делают и являются поэтому более гибкими и менее ограниченными. На сайте уже есть несколько статей, описывающих непараметрическую классификацию растровых данных при помощи [деревьев классификации в R](#) и [метода опорных векторов в imageSVM](#). Кроме того, помимо проприетарных решений, для решения этой задачи существуют и другие программные продукты с открытым исходным кодом, например, инструменты классификации есть в OTB и GRASS.

В большинстве случаев подобные решения являются самостоятельными приложениями и не интегрируются с ПО ГИС, также отсутствуют удобные инструменты подготовки и редактирования тренировочных наборов данных, что усложняет процесс их подготовки.

DTclassifier (Decision Tree classifier) является расширением свободной ГИС QGIS, которое позволяет пройти все этапы классификации данных дистанционного зондирования не покидая ГИС. Расширение использует алгоритмы библиотеки компьютерного зрения [OpenCV](#), а именно классификацию при помощи [деревьев решений](#).

Разработка расширения выполнена NextGIS в рамках проекта по мониторингу лесов сертифицированных FSC (Forest Stewardship Council).

Создано в  Разработка открытого ПО ГИС и реализация проектов

## Содержание

- [1 Получение и установка](#)
  - [1.1 Бинарная сборка](#)
  - [1.2 Исходный код](#)
- [2 Работа с расширением](#)
  - [2.1 Выбор тренировочных данных](#)
  - [2.2 Выбор классифицируемых данных](#)
  - [2.3 Дополнительные настройки](#)
- [3 Пример: детектирование рубок](#)
  - [3.1 Поиск и подготовка растровых данных](#)
  - [3.2 Сбор тренировочных данных](#)

### [3.3 Классификация](#)

- [4 Пример: анализ изменений](#)
- [5 Заключение](#)
- [6 Контакты](#)
- [7 Ссылки по теме](#)

## Получение и установка

Расширение доступно как в виде исходного кода C++, так и в бинарной форме.

### Бинарная сборка

Для работы с программой в ОС Windows понадобится:

- загрузить и установить QGIS версии 1.8.0 ([подробнее](#)). На данный момент поддерживается только эта версия QGIS.
- [загрузить](#) архив с расширением и необходимыми библиотеками
- извлечь содержимое архива в каталог модулей QGIS (обычно это C:\OSGeo4W\apps\qgis-dev\plugins и C:\OSGeo4W\bin)

Для работы с модулем в Linux необходимо:

- установить QGIS версии 1.8. Если пакеты для вашего дистрибутива отсутствуют — необходимо скомпилировать QGIS из исходных кодов
- загрузить архив с расширением:
- 32-разрядная версия ([загрузить](#)). **Примечание:** модуль собран в 32-х разрядной Slackware 13.1 с использованием OpenCV 2.3.1
- (опционально) проверить контрольную сумму при помощи md5sum или аналогичной программы:  
b3089f69602b9b55652380728f7b2a3c dtclassifier-linux.tar.bz2
- извлечь содержимое архива в каталог модулей QGIS (обычно это /usr/lib/qgis/plugins)

После установки нужно запустить QGIS и активировать расширение DTclassifier в Менеджере модулей («Модули → Управление модулями»).

### Исходный код

Исходный код модуля (лицензия GNU GPL v2) можно получить через [репозиторий на GitHub](#), или выполнив команду:

```
git clone git@github.com:nextgis/dtclassifier.git
```

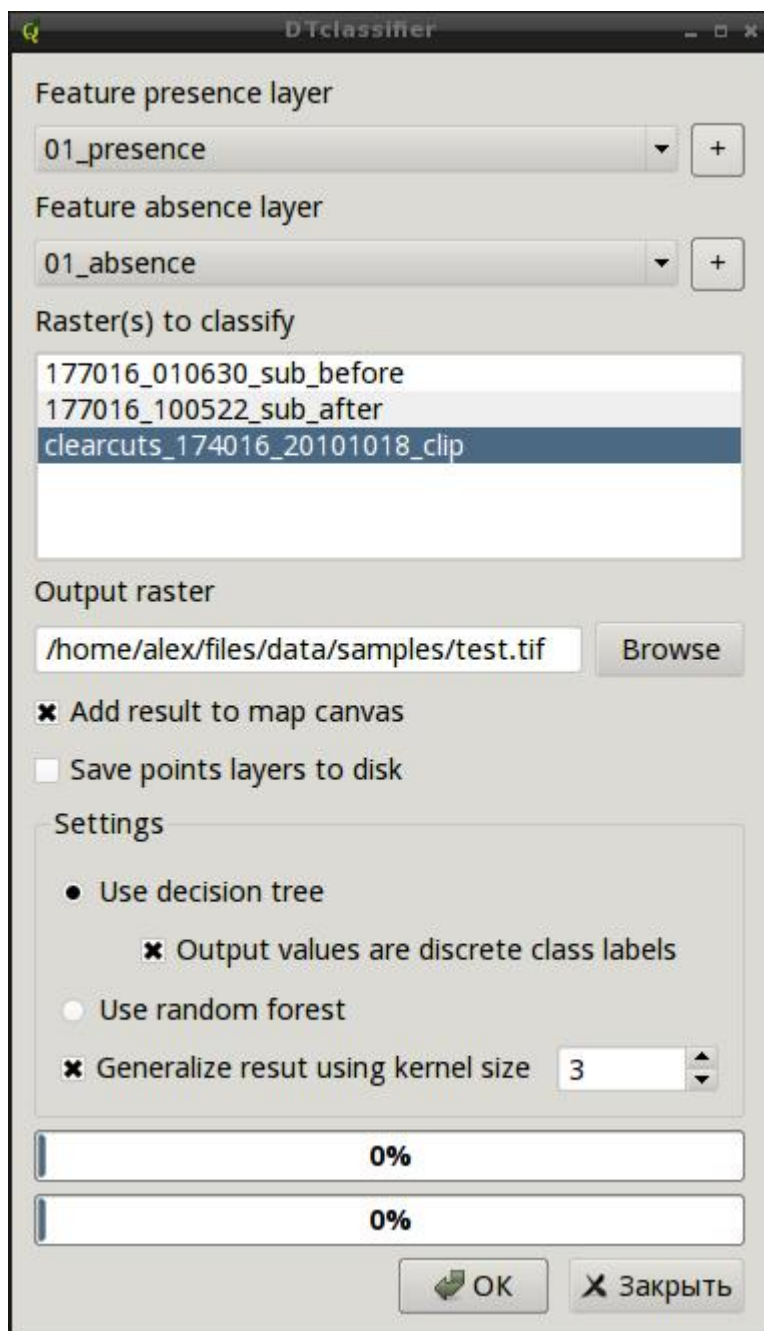
Также можно загрузить [архив](#) с данными, использовавшимися при подготовке статьи.

## Работа с расширением

После подключения и запуска расширения с помощью кнопки



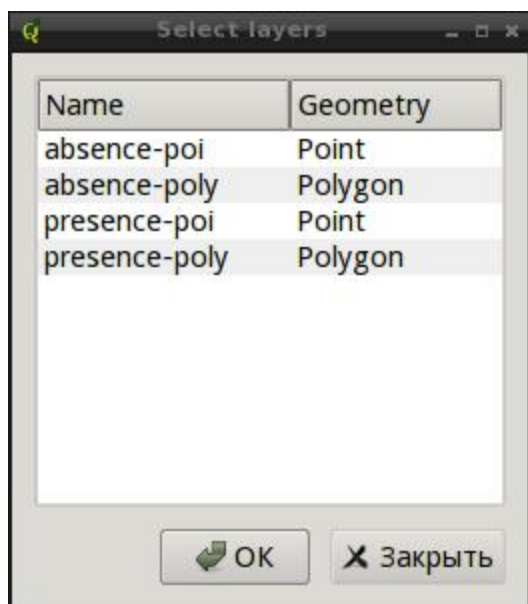
появится главное окно



### Выбор тренировочных данных

При помощи выпадающих списков «Feature presence layer» и «Feature absence layer» выбираются тренировочные слои, отмечающие области, где соответственно присутствует (presence) или отсутствует (absence) интересующих нас признак. Эти слои создаются заблаговременно, перед началом классификации. Создавать их можно прямо в QGIS по снимкам, используя всю функциональность для работы с векторными и растровыми данными. Подробнее о создании векторных данных можно прочитать в [Руководстве пользователя QGIS](#).

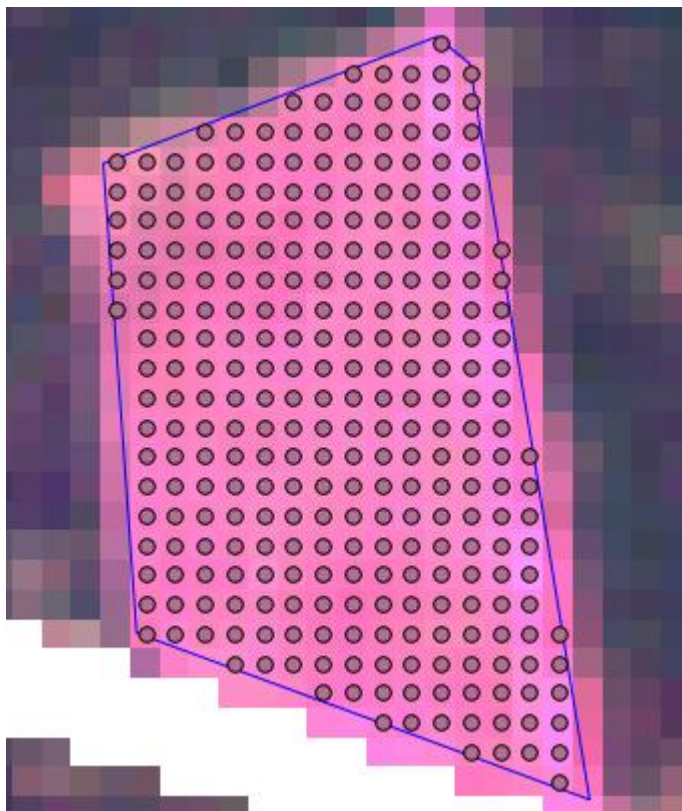
Расширение работает во всеми видами геометрий: точками, линиями и полигонами. Кнопка «+» рядом с каждым списком позволяет выбрать несколько слоёв с тренировочными данными одновременно, причем геометрии этих слоев могут быть различными.



Выпадающий список работает только если слой с тренировочными данными — один. При выборе нескольких слоёв выпадающий список блокируется и если необходимо вернуться к работе с одним слоем нужно нажать кнопку «+» еще раз.

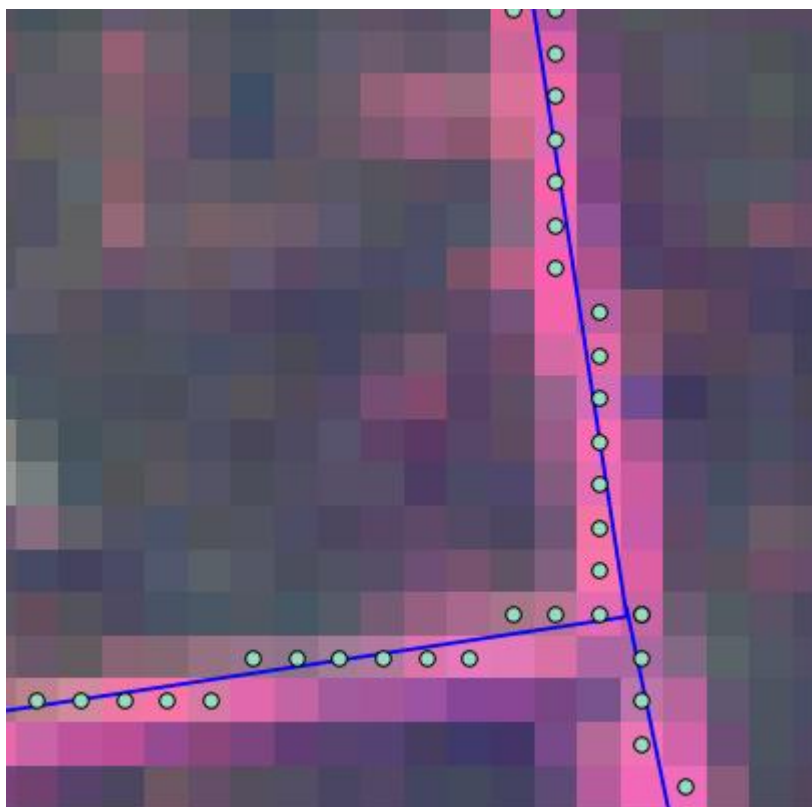
Все исходные слои будут собраны в один точечный shape-файл. При этом точечные слои будут записаны «как есть», а линейные и полигональные объекты будут сконвертированы в точечные так, чтобы получить точки во всех пикселях, попадающих в полигон или на линию.

Например для полигонального слоя:



Для определения точек, попадающих на линию, используется следующий алгоритм. Вокруг линии строится буферная зона шириной в 1/2 пикселя; в центре каждого пикселя, попадающего или пересекающего буферную зону ставится точка, затем отбираются точки, полностью попадающие в эту зону.

Пример для линейного слоя:



**ВАЖНО!** Правильные тренировочные данные — залог успеха. Необходимо внимательно следить за выбираемыми слоями. Если один и тот же слой отнести к обоим классам, результат классификации будет некорректным.

### Выбор классифицируемых данных

Поле «Raster(s) to classify» позволяет указать растр, который необходимо классифицировать. Поддерживается выделение нескольких растров с зажатым Shift или Ctrl, это может быть полезным при анализе изменений, когда используются растры «до» и «после». Если выбрано несколько растров, то тренировочные данные будут выбираться из каждого снимка.

**ВАЖНО!** При выборе нескольких растров необходимо убедиться, что на диске, где располагается временный каталог, достаточно места (объем свободного места должен быть примерно равен двойному объему анализируемых растров).

В поле «Output raster» указывается файл, в который будет сохранен результат классификации.

### Дополнительные настройки

Если установить флаг «Add result to map canvas», то по окончании процесса результат классификации будет загружен в QGIS. Установка флага «Save point layer to disk» заставит расширение сохранить тренировочные данные в точечный shape-файл рядом с итоговым файлом. Для каждой точки в файл записываются значения каналов растра и класс, которому принадлежит точка. Они могут быть использованы для построения модели и классификации в другом программном обеспечении.

Группа «Settings» служит для настройки классификации. Здесь можно выбрать используемый метод: одиночное дерево решений («Use decision tree») или [ансамбль деревьев](#) («Use random forest»). Обычно, классификация при помощи ансамбля деревьев дает более точные результаты. При использовании одиночного дерева становится доступным флаг «Output values are discrete class labels», установка которого приведет к тому, что исходные данные будут трактоваться как набор дискретных величин (меток). Если флаг не установлен, то исходные данные трактуются как непрерывные.

Здесь же можно активировать «сглаживание» результатов классификации с настраиваемым размером окна («Generalize result using kernel size»). При включенном сглаживании будет создан не один, а два растра: классифицированный и сглаженный (имя содержит суффикс «\_smooth»).

Помимо результатов классификации расширение сохраняет в формате YAML используемое при классификации дерево (имя файла формируется из имени файла назначения и суффикса «\_tree»).

Нажатие кнопки «Ok» запустит процесс извлечения тренировочных данных, обучение классификатора и собственно классификацию. Ход классификации отображается при помощи двух индикаторов: верхний отображает прогресс текущей операции, а нижний — общий прогресс.

## **Пример: детектирование рубок**

Рассмотрим работу с расширением на примере решения одной из задач мониторинга лесохозяйственной деятельности: детектировании рубок.

Процесс тематической классификации состоит из нескольких этапов:

- поиск и подготовка растровых данных
- сбор тренировочных данных
- построение модели по тренировочным данным
- применение модели к растровым данным и получение результата

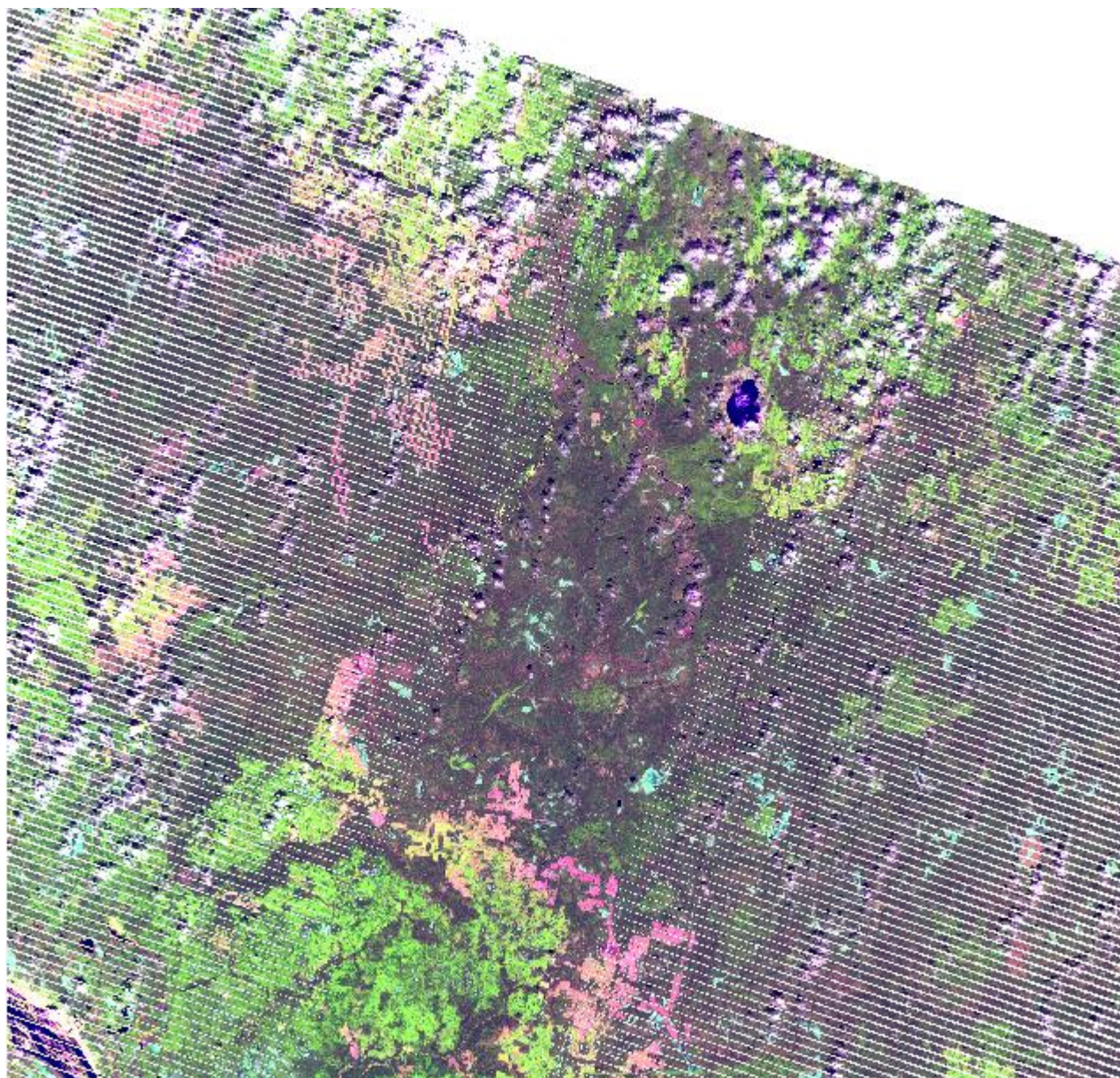
### **Поиск и подготовка растровых данных**

Прежде всего необходимо найти и привязать растры, которые в дальнейшем будут классифицироваться. Обычно, это данные космической съемки (например, Landsat), на которых хорошо видны интересующие нас объекты. Если вы не знаете где взять снимки, рекомендуем ознакомиться со статьями, описывающими процесс [получения данных Landsat](#) через сервис Glovis и их [дальнейшую обработку](#) (а именно, объединение отдельных каналов в единый файл).

Если вы хотите повторить пример, загрузите [архив](#) с исходными данными.

Для задачи детектирования рубок возьмем растр after.tif и выставим комбинацию каналов 5-4-3, чтобы было легче различать рубки (розовые на снимке).



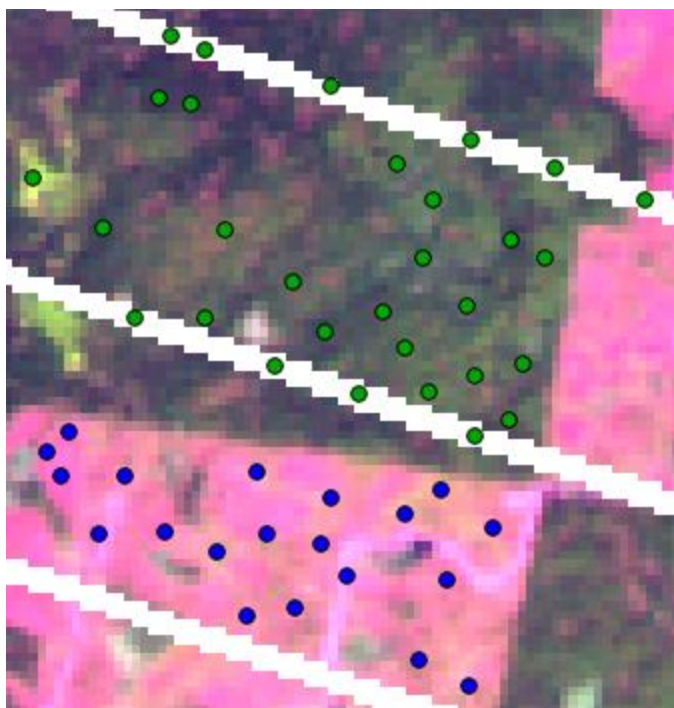


### **Сбор тренировочных данных**

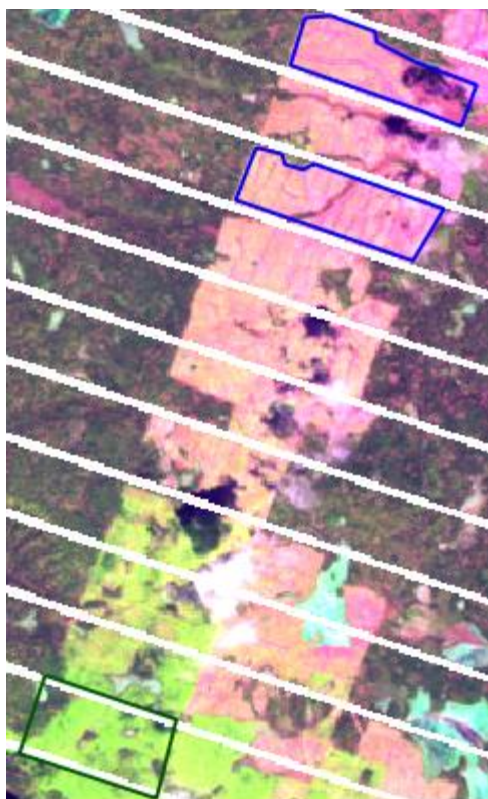
Для построения дерева решений необходимо подготовить тренировочные данные (т.н. «обучение с учителем»), содержащие объекты двух классов: присутствие и отсутствие рубок.

Нам необходимо как минимум два векторных слоя: один соответствующий рубкам, а второй — местам, где рубок нет. Так, можно создать два точечных слоя и расставить точки в нужных местах (на рисунке синие точки соответствуют рубкам, зеленые — отсутствие рубок).





Но такой способ не очень удобен, поскольку результат классификации зависит от количества тренировочных данных (чем их больше, тем точнее результат). Поэтому, рациональнее создать полигональные слои и обвести несколько рубок, аналогично поступить и с областями, где их нет (синий цвет соответствует рубкам, зеленый — рубок нет).



Так как расширение поддерживает все типы геометрии (точки, линии, полигоны), то можно использовать несколько слоев одновременно. Загрузим слои рубок `presence-poly.shp`, `presence-poi.shp` (полигоны и точки соответственно) и аналогичные слои с данными об отсутствии рубок — `absence-poly.shp`, `absence-poi.shp`.

Если включить флаг «Save point layer to disk», то точечный слой, полученный из исходных тренировочных слоев, будет сохранен для дальнейшего использования. Например, сохранив атрибутивную таблицу в формат CSV, можно выполнить классификацию этого же растра в R ([подробнее](#)), а затем сравнить результаты.

Создание точечного слоя и извлечение значений пикселей растра будет выполняться в автоматическом режиме, после запуска процесса классификации.



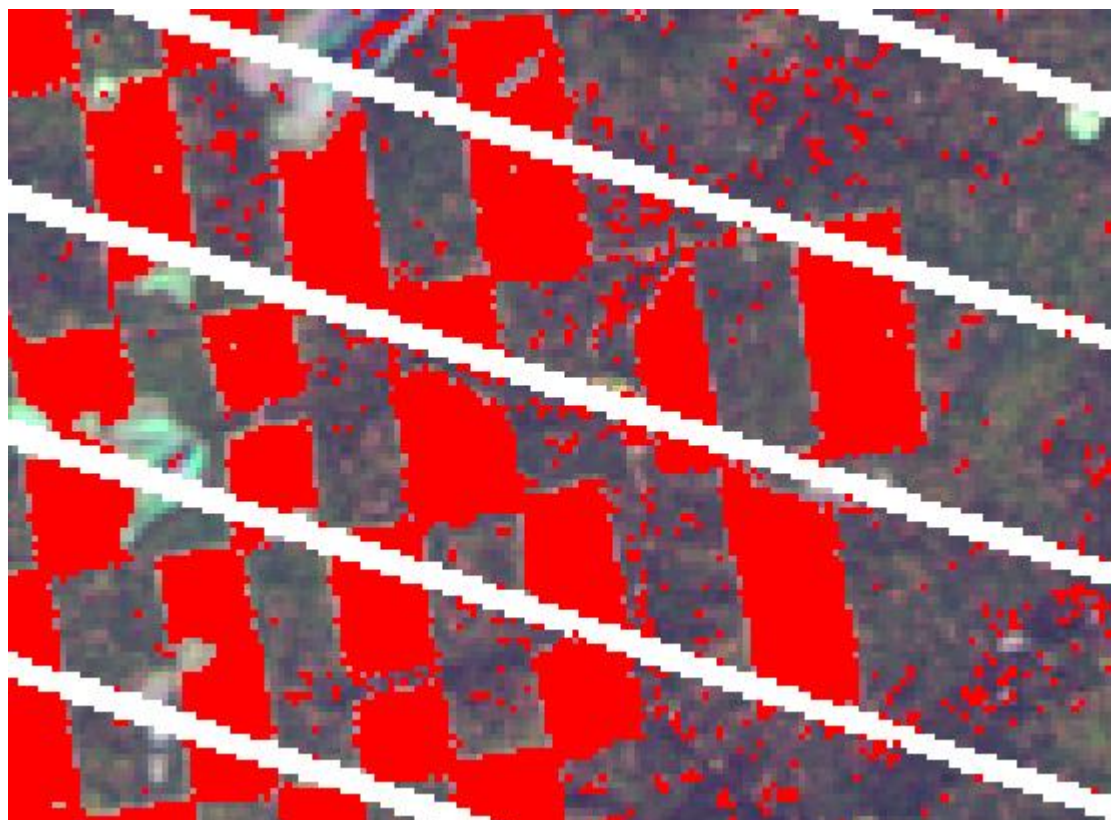
## Классификация

После подготовки тренировочных слоёв и исходного растра можно приступить к анализу. Используя кнопки «+» рядом с выпадающими списками указываем тренировочные слои (наличие рубок — presence layer; отсутствие рубок — absence layer), в списке растров выбираем растр after.tif, указываем путь для сохранения результатов.

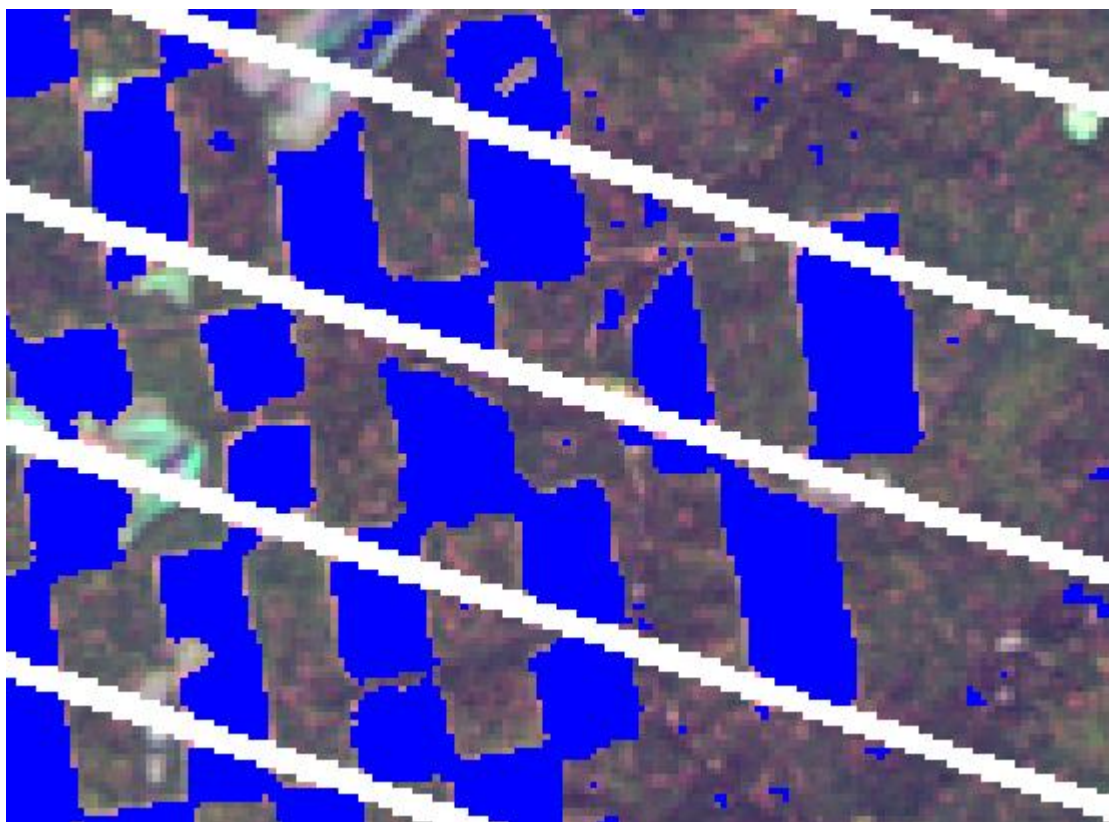
Так как значения пикселей растра являются непрерывными значениями, флаг «Output values are discrete class labels» активировать не нужно. А вот сглаживание («Generalize result using kernel size») стоит включить: это уберет «шумы» и сделает результат более четким. Размер окна по умолчанию (3) в большинстве случаев достаточен.

Нажатием на кнопку «Ок» запускаем классификацию. После непродолжительного ожидания получим два новых растра.

Фрагмент классифицированного растра:



Видно, что много отдельных пикселей было классифицировано как рубки, хотя на самом деле они не относятся к ним. Для устранения такого «шума» и нужно сглаживание. Ниже показан тот же фрагмент изображения, но уже после сглаживания:



Видим, что одиночных пикселей меньше, а границы рубок стали более четкими.

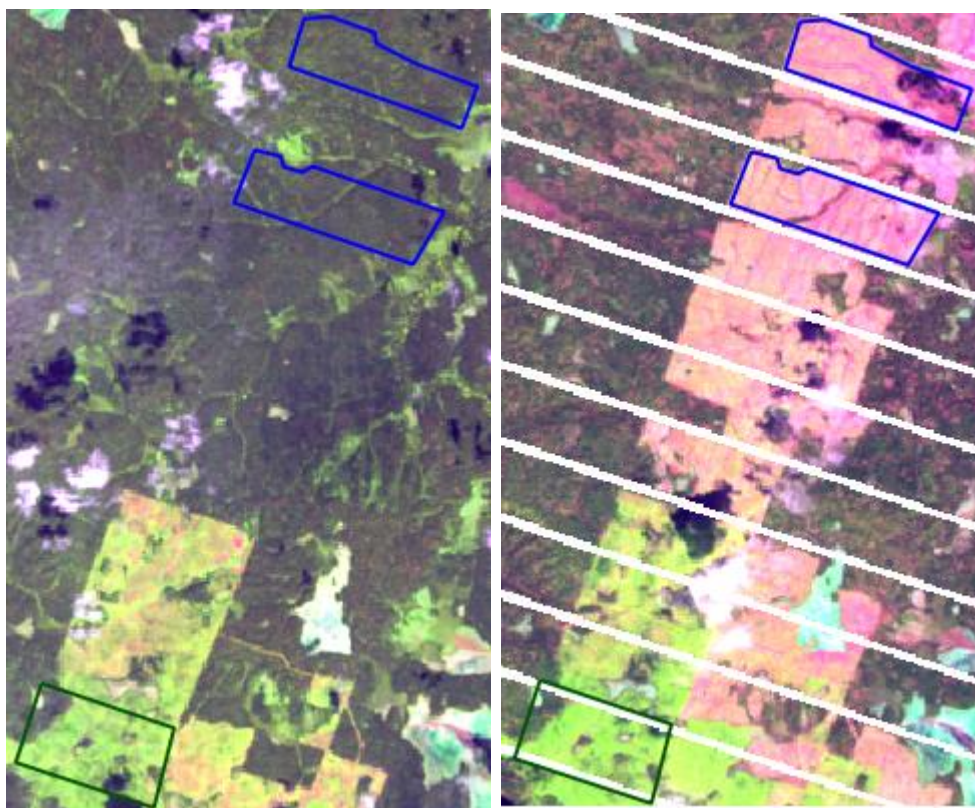
### Пример: анализ изменений

Еще одной задачей, которую можно решить при помощи DTclassifier является анализ изменений. Суть такого анализа состоит в том, что берется серия растров (в простейшем случае — пара) до и после некоторого события. Затем выделяются изменившиеся области.

Основные этапы в этом случае точно такие же, как при тематической классификации, приводить их еще раз мы не будем, а просто на примере разберем как выполняется анализ изменений при помощи DTclassifier. В качестве исходных данных возьмем все тот же архив с файлами.

Создадим новый проект в QGIS и загрузим оба растра: before.tif и after.tif. Как видно из названия, снимок before.tif выполнен до интересующего нас события (в данном случае — вырубка леса), а снимок after — после. Применив этот подход, мы можем выделить рубки появившиеся в промежутке между снимками «до» и «после».

Сбор тренировочных данных немного отличается от описаного в предыдущем примере. Если при тематической классификации данные набирались по принципу признак присутствует / признак отсутствует, то теперь данные отбираются по двум растрам. В слое изменений (presence) заносятся области, в которых признак присутствует на снимке «после», но отсутствует на снимке «до».



А вот слои отсутствия признака (absence) формируются, как и в предыдущем примере, из областей где на обоих снимках интересующий нас признак отсутствует.

В этом примере в качестве тренировочных данных будем использовать только полигональные слои presence-poly.shp и absence-poly.shp, разумеется, при необходимости допускается использование нескольких слоёв.

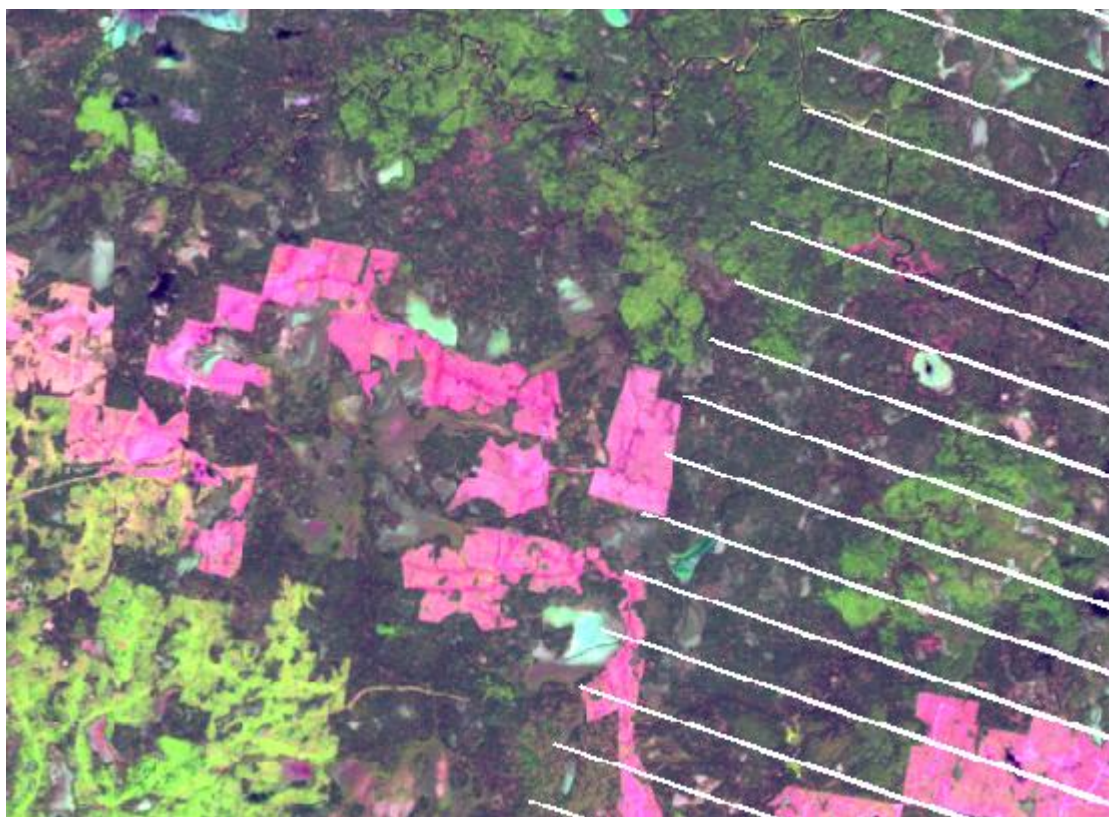
Вызовем главное окно DTclassifier, в выпадающих списках «Feature presence layer» и «Feature absence layer» выбираем слои presence-poly и absence-poly соответственно. В списке исходных растров, зажав Shift, выделяем оба растра before и after. Указываем куда сохранять результат, настройки классификации можно оставить без изменений. Нажатием на кнопку «ОК» запускаем анализ и ждем его окончания.

Ниже показан фрагмент снимка до вырубki:

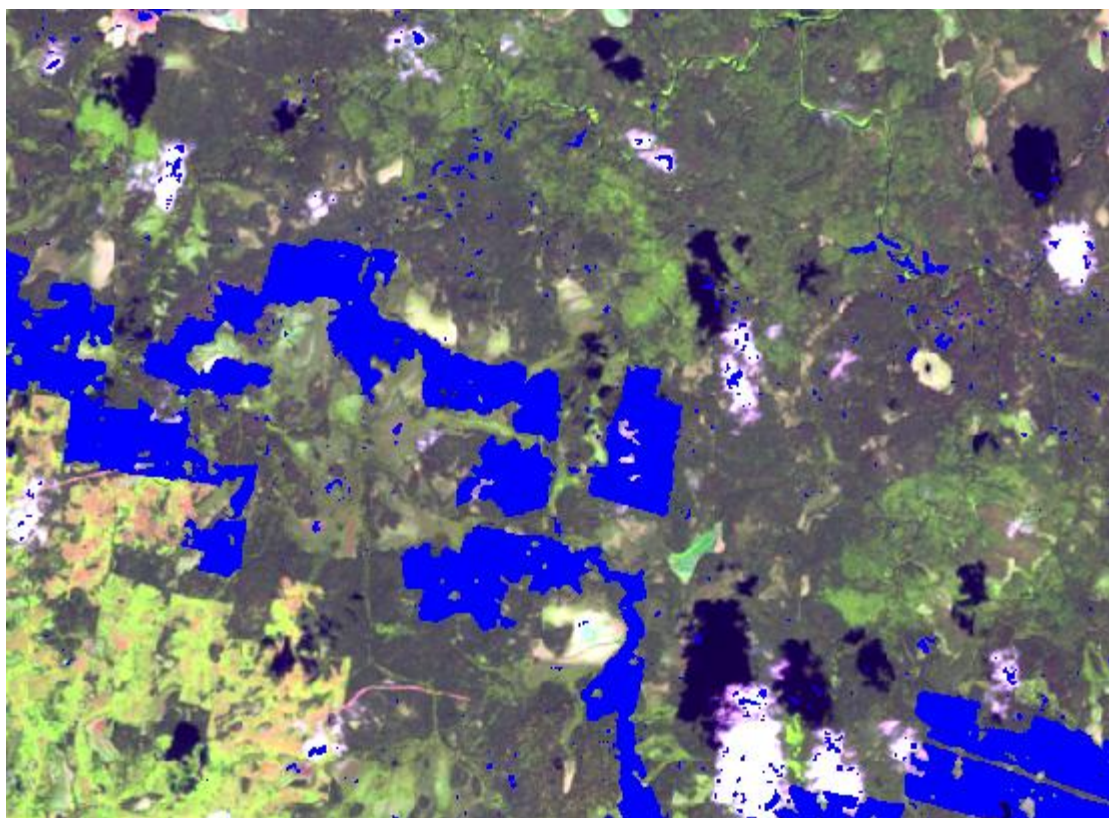




Тот же участок после рубки:



Результат анализа наложен на снимок предшествующий вырубке леса:



## Заключение

DTClassifier является достаточно простым и функциональным интегрированным расширением для решения задач классификации растровых данных на два класса (признак присутствует / признак отсутствует), а также для анализа изменений. Интеграция с QGIS позволяет быстро модифицировать тренировочные данные, и подбирать оптимальное их количество, а благодаря использованию библиотеки компьютерного зрения OpenCV расширение DTClassifier обладает высокой скоростью работы.

## Контакты

Если вы нашли ошибку, у вас есть предложения по улучшению расширения или просто вопросы по использованию — напишите [нам](#).



Создание открытого ПО ГИС и реализация проектов

## Ссылки по теме

- [OpenCV Wiki](#)
- [Классификация растровых данных с помощью деревьев решений в R](#)
- [Классификация данных ДЗЗ используя метод опорных векторов и imageSVM](#)
- [Распознавание сгоревших территорий с помощью деревьев решений и OpenCV](#)

[Обсудить в форуме](#) Комментариев — 87

Последнее обновление: 2014-05-15 00:34

Дата создания: 21.06.2011

Автор(ы): [Александр Бруй](#), [Максим Дубинин](#)