Ireland Forest Disturbance from Earth Observation (IForDEO) Python module user manual

Guy Serbin, Environment, Soils, and Land Use Department, CELUP, Teagasc, Johnstown Castle, Co. Wexford Y35 TC97, Ireland.

Telephone: +353-53-917-1201

Email: guy.serbin@teagasc.ie

Abstract

An open-source Python module has been developed to assess forest disturbance using remotely sensed Landsat imagery, and referred to as the Ireland Forest Disturbance from Earth Observation (IForDEO). This module is designed to ingest georeferenced surface reflectance data and cloud masks, and produce forestry land use change maps for the entirety of Ireland.

Keywords: Remote sensing, Landsat, forestry, land use, land cover, LCLUC

Introduction

The Ireland Forest Disturbance from Earth Observation (IForDEO) model utilises Landsat imagery to determine forest parcel land use change. The model performs a number of steps, including calculation of individual scene and yearly land cover classes, and time-series analyses of yearly land cover data to produce land cover/ land use change maps. IForDEO denotes the new name of a spectrally-based decision tree classifier named Decision Tree Number 4, or DT4 for short, which was then modified to first to DT4a, then to DT4b. The difference between DT4 and DT4a/b has to do with partitioning of land cover classes between mature forest and grass/ cropland classes (separating grass and cropland pixels can be difficult without phenological data). DT4 uses a set threshold value between the two classes, which was originally set between 30% to 40% near infrared (NIR) reflectance values at 2.5% intervals to identify the ideal threshold value. In DT4a/b, any NIR reflectance for a vegetation spectrum that falls between these values is initially assigned to a separate fuzzy class that is later assigned to forest or grass/ cropland in the yearly aggregation stage.

This code was developed as part of the CForRep project.

Installation

It is essential that each of the installation steps detailed from hereon are performed in order. They have only been tested on Anaconda Python 3.4 – 3.6 (Continuum Analytics, https://www.anaconda.com/download/) distributions running on Microsoft Windows 7 and 2012 Server 64 bit operating systems. However, the software may work on any Python version from 2.7 or greater, or operating system, though this hasn't been properly tested.

Python

It is recommended, but not required to use Anaconda Python distribution (Continuum Analytics, https://www.anaconda.com/download/). Once installed, the software will also require installation of the following additional Python modules:

- Numexpr
- Numpy
- GDAL

It is recommended that the modules be set up in Anaconda Python via the following from a command prompt:
> conda install –c conda-forge gdal numpy numexpr

The Irish Earth Observation (IEO) Python module

Every piece of software described from hereon is dependent on the Irish Earth Observation (IEO) Python module (https://github.com/Teagasc/ieo). The installation is as follows:

1. Download the code to an installation directory, e.g., "D:\install\ieo". The most recent installation instructions will be in the "INSTALL" file, and take precedence over those in this manual.
2. In the Anaconda Prompt command line environment, change directory to the one in which the IEO installation files reside.
3. At the command line window type:
   > python setup.py install
4. The installer may ask you where imagery data are or will be located on your systems. These directories will be:
    a. The base directory, e.g., on a Windows system "D:\data" or "E:\Imagery". This directory should already exist on the hard drive, and have at minimum several terabytes of free disk space. In this directory, a "Landsat" subdirectory will be created to house Landsat imagery data (includes Fmask, SR, BT, NDVI, EVI subdirectories).
    b. You can optionally specify directories specific functions, though these will automatically be created in the base directory by default. These directories are for:
        i. Data catalogues;
        ii. Landsat LEDAPS tar.gz format) ingestion (if you order and download new LEDAPS-corrected Landsat data, those files go here) and archiving (where they go after ingestion);
        iii. Log files.

IEOtools

Following the successful installation of the IEO module, you'll need to download a set of Landsat processing scripts known as IEOtools from https://github.com/Teagasc/IEOtools to a pre-determined script directory, e.g., "D:\scripts". IEOtools are used to identify, download, process, and manage the local Landsat archive that will be used to run IForDEO.

The IForDEO Python Module

The installation for IForDEO (https://github.com/DrGuy/IForDEO) is very similar to the IEO module:

1. Download the code to an installation directory, e.g., "D:\install\IForDeo". The most recent installation instructions will be in the "INSTALL" file, and take precedence over those in this manual.
2. In the Anaconda Prompt command line environment, change directory to the one in which the IForDEO installation files reside.
3. At the command line window type:
   > python setup.py install
4. You may be asked by the installation for the following directory or locations:
   a. Base directory for IForDEO output data. If asked, then you need input this. It should be a something like "D:\data\IForDEO".
   b. Optional: an alternate file path for the IForDEO data catalogue. By default, it will be a subdirectory in the main IForDEO data output directory.
   c. Optional, but highly recommended: the file path of a raster forestry mask, where 1 = forestry parcel, 0 = not a forestry parcel. This isn't required, but having one will greatly reduce model processing times.
   d. Optional: the location of a shapefile in Irish Transverse Mercator (ITM) projection. You'll only need this if you wish to create your own processing tile grids. Be aware that a 29 x 30 km grid for all of Ireland is included with this distribution and referenced in the code.

Workflow

The first task in the workflow is to identify needed Landsat scenes, download, and process them. The best option is to rely on surface reflectance products provided by the USGS/EROS/ESPA, but this can be augmented by scenes that area available through ESA/EOLi. However, ESA/EOLi does not provide data in surface reflectance, and thus, any scenes from them must be processed locally.

Landsat data acquisition from USGS/EROS/ESPA:

1. updateshp.py – This script downloads all available scene metadata from the USGS/EROS Data Center, and adds new metadata and scene footprint shapes to the catalogue in WRS2_Ireland_scenes.shp, and new scene thumbnail images to the "Thumbnails" subdirectory of the catalogue. It needs to be updated to account for changes in filenames instituted by the USGS/EROS earlier this year. This is designed to run as-is, with no command line options.
2. MakeESPAproclist.py – This script checks the available scenes in the shapefile from the previous script, and existing surface reflectance data in the "Landsat/SR" subdirectory. It then creates a list of new scenes to download, excluding scenes that cannot be processed by LEDAPS due to low sun elevation angles (< 15°) or lack of ancillary data (see https://landsat.usgs.gov/landsat-surface-reflectance-high-level-data-products). As with the previous script, it also needs to be updated to account for changes in Landsat filenames. This includes a number of command line options, which are described by:
   > python MakeESPAproclist.py --help

3. The list produced in step 2 is uploaded to https://espa.cr.usgs.gov/, and the following products are requested:
    a. Input metadata
    b. Surface reflectance
    c. Brightness temperature
    d. Cloud mask (CFmask, note that this is scheduled to become part of a QA/QC layer in future versions of Landsat data)
    e. ENVI format (algorithms will work with GeoTiff as well, but not HDF).
4. Once the order is complete, processed scenes are downloaded using the ESPA Bulk Downloader software (https://github.com/USGS-EROS/espa-bulk-downloader).

Optional: Landsat data from ESA/EOLi (requires a Linux machine for the LEDAPS/ CFmask processing software):

1. Scene data are discovered using the EOLi-SA software package (https://earth.esa.int/web/guest/eoli) and saved to a CSV file.
2. The CSV file from step 1 is processed using MakeEOLIdllist.py. This function creates an HTML file that can be used to download scenes.
3. The HTML file is opened with the Mozilla Firefox browser, and the DownThemAll! plugin (https://addons.mozilla.org/en-US/firefox/addon/downthemall/) is used to download the scenes.
4. Once on disk, scenes are processed to surface reflectance using LEDAPS software (https://github.com/USGS-EROS/espa-surface-reflectance) and the CFmask clear land mask are calculated (https://github.com/USGS-EROS/espa-cloud-masking) on a Linux machine.

LEDAPS data import to remote sensing data archive:

1. newespaimport.py scans for new available scenes in pre-defined ingest directories. The script does the following tasks:
    a. Calls the IEO Python library (https://github.com/DrGuy/ieo).
    b. Extracts scene data from tar.gz archives if needed.
    c. Creates virtual image band stacks (VRTs) of surface reflectance, and if needed, thermal band data.
    d. Warps CFmask, reflectance, and thermal data from Universal Transverse Mercator (UTM) zones 29 or 30 North to Irish Transverse Mercator (ITM) projection and stores them in appropriate subdirectories of D:\Spatial Analysis Unit\Archive\Landsat (Fmask, SR, and BT, respectively). Landsat 8 TIRS thermal data are saved to the BT\Landsat8 subdirectory.
    e. Calculates the Normalised Difference Vegetation Index (NDVI) and Enhanced Vegetation Index (EVI), and saves them in the NDVI and EVI subdirectories, respectively.
    f. A summary of available data on the server may be seen in Table 1.

Running the IForDEO De/Af/Re-forestation model:

Within Python code, or from a Python prompt (e.g., IDLE, iPython, Jupyter Notebooks), import the library via:

import ifordeo

From here, the following functions need to be executed in order. Running them as-is will process the data for all of Ireland.

1. ifordeo.batchdt4(): This function scans for Landsat surface reflectance and CFmask files. Scenes that have a minimum number of usable pixels (default = 1000) are classified to land cover types using a rule-based spectral classification algorithm in ifordeo.dt4b(), Figure 1. Existing classifications are overwritten if the "overwrite" variable is set. The "foresttograss" value is a reflectance value above which any vegetation signature is considered to be grassland. This is varied from 3000 (reflectance = 0.300) to 4000 (reflectance = 0.400). If the "dt4b" Boolean variable is set to true (default), then a fuzzy confusion class is created between these values, but if not, then the respective classifications are created at increments of 250 (reflectance = 0.025) in separate subdirectories. A summary of these may be seen in Table 2.
2. ifordeovrt.py: this script looks for classifications on a specific day, and creates virtual mosaics. A summary of these may be seen in Table 2. It also creates catalogue files and shapefiles of these and single scenes for all available dates in:
    a. Simple catalogue CSV files, created in the ifordeo.catdir subdirectory.
    b. Catalogue shapefiles, created in the "shp" subdirectory of ifordeo.catdir.
3. ifordeo.makemaps() then reads in a shapefile containing a processing tile grid for the island of Ireland, which is included with the IForDEO model. Iterating over each tile, it calls the ifordeo.proctile() function. This function ignores data from dates that were found with geometric accuracy issues in Table 3, and calls these functions:
    a. Loops through available years and calls (Figure 1):
        i. ifordeo.calcprobabilityraster(): This reads the catalogue shapefiles created in step 2, and calculates land cover class probabilities for a given year for said tile using intersecting DT4b VRT and individual scene files. Data are no longer in WRS-2 scene space, but now in tile space.
        ii. ifordeo.Yearlydt4(): from probabilities, a yearly majority land cover class is determined.
        iii. ifordeo.forestryclass(): This simplifies the yearly majority land cover classes into four classes: No data, not forest, possible forest, and forest.
    b. Calls ifordeo.calcyearlychange(), which reads in time series forestry class data for all years. It calls ifordeo.cleansignal() to remove noise and simplify pixels with data to two classes- not forest and forest. From these series, afforestation, clearcut, and reforestation events are determined by year. Clearcut events that have not been reforested in ten or more years are classed as deforested, those between five to nine years as possible deforestation, and those as less than five years as recent clearcut. A forestry land use change map is then created for the tile.

Known issues and workarounds

1. Depending upon your local machine/ operating system setup, the IEO or IForDEO modules may not load when scripts are run from command line. Workarounds for this:

a. Run a Python environment, e.g., iPython, and then load the module from within:
import ifordeo

b. Call the original installation files in your script (may not properly access configuration data if you haven't run setup.py) via:
import sys
sys.path.append(r'<path to IEO module')
sys.path.append(r'<path to IForDEO module')
import ifordeo

2. The modules cannot access data within the Python eggs, e.g., shapefiles (note- this will be addressed in future releases of the code). In this case, you can manually copy the data files in the installation directories to another folder, then set the module variables after the module has been loaded in the script or environment via:
ifordeo.margs.shp = r'<path to shapefile directory>\IRL_tiles_30.shp'

Glossary

Item	Description
BT	Thermal infrared brightness temperature
EROS	Earth Resources Observations and Science Center
ESPA	EROS Science Processing Architecture
ETM or ETM+	Landsat Enhanced Thematic Mapper +
EVI	Enhanced Vegetation Index
Fmask	Clear land, water, snow/ice, cloud, and cloud shadow mask
INRT	Ireland National Raster Tile
ITM	Irish Transverse Mercator
LEDAPS	Landsat Ecosystem Disturbance Adaptive Processing System
NDVI	Normalised Difference Vegetation Index
OLI	Landsat Operational Land Imager
OLI/TIRS	Landsat Operational Land Imager/Thermal Infrared Sensor
SAVI	Soil Adjusted Vegetation Index
SR	Surface reflectance
SR	Surface Reflectance
TIRS	Landsat Thermal Infrared Sensor
TM	Landsat Thematic Mapper
TOA	Top Of Atmosphere
USGS	United States Geological Survey
UTM	Universal Transverse Mercator
WRS	Landsat World Reference System

Tables

Table 1. Landsat data sets and their storage locations at Teagasc.

Data type	Subdirectory	Regular scenes	L1G scenes
Brightness temperature	BT	2924	1534
Surface reflectance	SR	4455	1657
Fmask	Fmask	4817	1661
NDVI	NDVI	4334	1387
EVI	EVI	4332	1274

Table 2. Processed DT4a land cover scenes in Teagasc's IForDEO archive per foresttograss value. "DT4a" denotes classifications that use a confusion class for vegetation with a maximum NIR reflectance between 30 – 40%. "VRTs" denote virtual rasters comprising one or more single scenes, and "bad dates" denote the number of dates that were found that contain serious geometric or radiometric errors.

foresttograss value	Single scenes	VRTs	Bad dates
DT4a	2345	925	10
3000	2183	914	
3250	2183	914	
3500	2183	914	
3570	2183	914	
4000	2183	914	

Table 3. List of dates for Irish Landsat L1T scenes with known geometric errors. YYYY denotes year, JJJ denotes day of year. It should be noted that some data may be correctly located for part of the WRS-2 path, but that in other areas serious errors were discovered.

Date (YYYYJJJ)
1984113
1986253
1988083
1988085
1988101
1988140
1988156
1988268
1989103
1989165
1989263
1990216
1990250
1994133
1994243
1998165
1998256
1999192
2011116
2014274

Figures

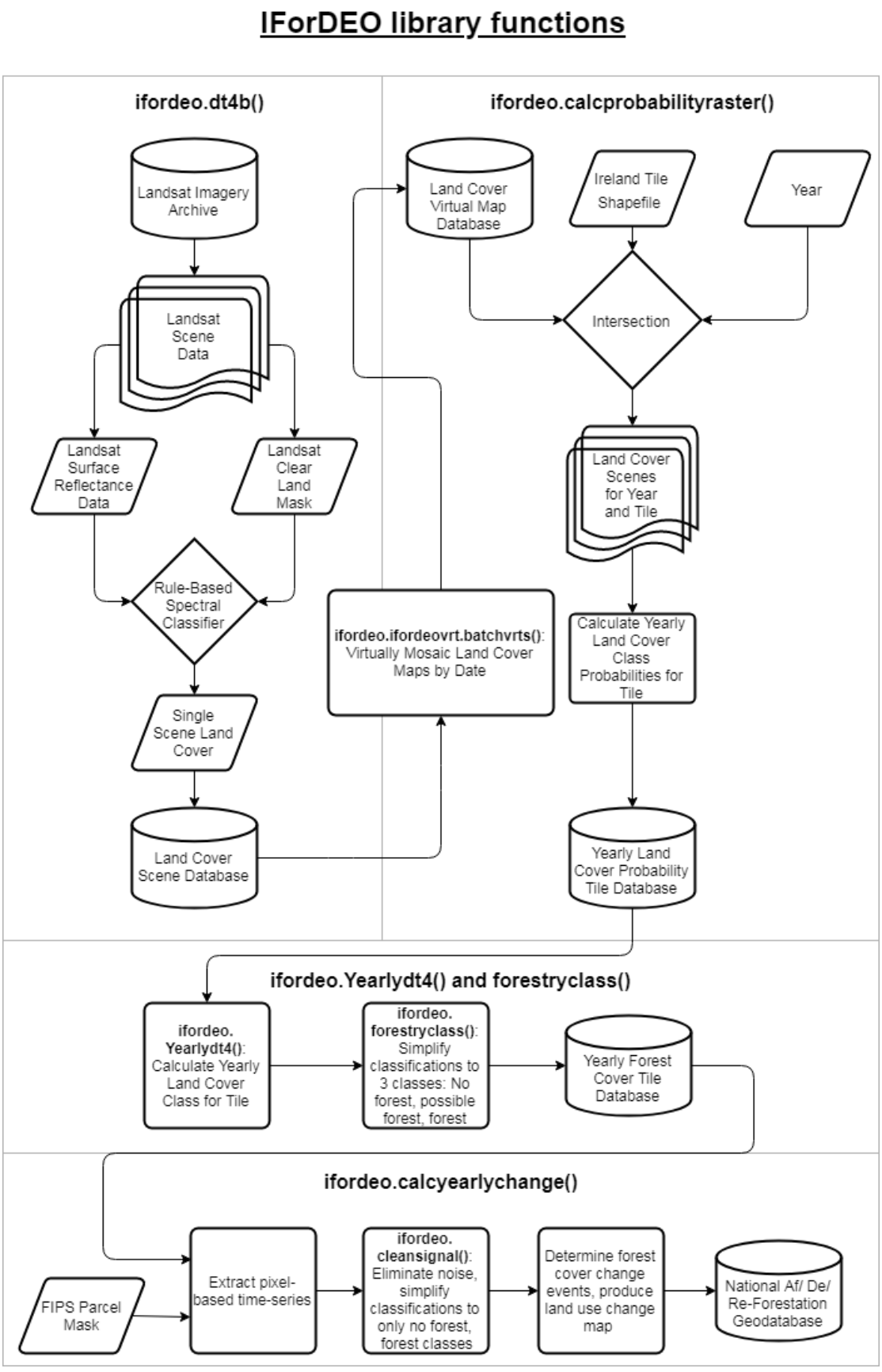

Figure 1. Flowchart detailing various ifordeo.py functions.

Changelog

8 – 9 February 2018:	Version 1.1.0: Replaced DT4a with DT4b as the default land cover algorithm and made various minor code fixes.
12 February 2018:	Version 1.1.0: (1) The makegrid() function has been incorporated into the IEO code, and is now deprecated within this module. (2) The AIRT shapefile is now part of ieo.gdb, which will be distributed with IEO 1.1.0 and higher.