



Deliverable D4.7

Software of the final operational LAI prototype

V. 1.0



The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement n° 730074

Document information

Project Number	730074	Acronym	SENSAGRI
Full Title	Sentinels Synergy for Agriculture		
Project URL	http://sensagri.eu		
Project Coordinator	José F. Moreno. IPL – University of Valencia (Spain)		
EU Project Officer	Massimo Ciscato		

Deliverable	Number	4.07	Title	Software of the final operational LAI prototype
Work Package	Number	4	Title	Leaf Area Index (LAI) Service

Date of Delivery	Contractual	M4	Actual	M4
Status				Final <input type="checkbox"/>
Type¹	R <input checked="" type="checkbox"/>	DEM <input type="checkbox"/>	DEC <input type="checkbox"/>	OTHER <input type="checkbox"/> ETHICS <input type="checkbox"/>
Dissemination Level²	PU <input checked="" type="checkbox"/>	CO <input type="checkbox"/>	EU-RES <input type="checkbox"/>	EU-CON <input type="checkbox"/> EU-SEC <input type="checkbox"/>

Responsible partner	UVEG			
Responsible Author	Name	Eatidal Amin	E-mail	eatidal.amin@uv.es
	Partner	UVEG	Phone	+34 963 54 40 67
Other authors				

Abstract (for dissemination)	This document serves as User Manual for the final version of an automated processing chain for independent LAI <i>green</i> and LAI <i>brown</i> retrieval from Sentinel-2(S2) images. The software is developed within MATLAB, following Object-Oriented program design. According to this programming scheme, both data components and operations performed on that data are implemented by defining classes. MATLAB class definition comprises code blocks that determine properties and methods. Properties store data and methods determine the operations that can be performed on that data. The workflow contained in a class can be executed by creating objects, which are specific instances of a class. The retrieval software requires the user to provide the directory which contain the S2 images to be processed and the output folder where the delivery products will be saved.
Keywords	LAI green, LAI brown, S2, prototype software, automated retrieval

Version Log			
Issue Date	Rev. No.	Author	Change
6 May 2019	0.1	Eatidal Amin	First version
7 May 2019	1.0	Antonio Ruiz	Revised

¹ R = Document, report; DEM = Demonstrator, pilot, prototype; DEC = Websites, patent fillings, videos, etc; OTHER; ETHICS = Ethics requirement

² PU = Public; CO = Confidential (Consortium and Commission Services); EU-RES = Restreint UE; EU-CON Confidential UE; EU-SEC = Secret UE (Commission Decision 2005/444/EC)

Table of Contents

Document information.....	3
Table of Contents	4
List of Figures	4
List of Tables.....	4
1. Revision History.....	5
2. Programming syntax	6
3. Software Package	7
4. Instructions.....	8

List of Figures

Figure 1. Example of definition class syntax.	6
Figure 2. Content of the software package folder Software_LAIGB_v0.3.	7
Figure 3. Content of auxiliar_class folder.....	7
Figure 4. Content of models folder.	7
Figure 5. Capture of MATLAB session with the current working directory set to the software package folder.....	8
Figure 6. Input folder content and output folder containing the delivery product.....	9

List of Tables

Table 1. Software revision history.....	5
---	---

1. Revision History

Table 1. Software revision history.

Version	Date	Revision Description	Authors
0.1	02/11/17	First implementation and release of the prototype software for automated LAI <i>green</i> and LAI <i>brown</i> retrieval and its documentation	Juan Pablo Rivera, Eatidal Amin, Jochem Verrelst
0.2	10/04/18	Reading and loading S2 jp2 images (SAFE) Save output according to naming convention. LAI total calculation Save output as GeoTIFF cloud and vegetation mask (from S2 scene classification band) Adding a description text file for each processed S2 image	Eatidal Amin, Juan Pablo Rivera
0.3	29/05/19	Only S2 images in SAFE format are accepted as input Correction of mask handling Modification in LAI total calculation Error on the writing S2 SCL as mask band Adding date and time tags in the output product	Eatidal Amin

2. Programming syntax

MATLAB classes definition is based on a code organized into modular blocks (Figure 1), delimited by keywords. All keywords have an associated *end* statement that closes each block:

- `classdef ... end`. Class definition block which contains all the other class components subblocks. MATLAB file name and class name must be the same.
- `properties ... end`. In this sub-block, all the properties names and assignment of default values are specified. Each property refers to a certain variable and stores data.
- `methods ... end`. Contains operations defined as MATLAB functions. The constructor method has the same name as the class.

```
1 classdef ClassName
2     properties
3         PropertyName
4     end
5     methods
6         function obj = ClassName(arg1,...)
7             obj.PropertyName = arg1;
8             ...
9         end
10        function ordinaryMethod(obj,arg1,...)
11            ...
12        end
13    end
14    methods (Static)
15        function staticMethod(arg1,...)
16            ...
17        end
18    end
19 end
20
```

Figure 1. Example of definition class syntax.

3. Software Package

The prototype software is composed of a main folder **Software_LAIGB_v0.3** (Figure 2), which includes the following files:

- **auxiliar_class**. Folder containing several MATLAB functions and auxiliary classes needed for importing and running the GPR retrieval algorithm models; and implementing the reading and writing algorithm for multiband images (data file + header file and geotiff) (Figure 3).
- **models**. Folder containing the LAI *green* and LAI *brown* GPR models (. mat files) (Figure 4).
- LAI *green* and LAI *brown* **colortables** (. mat files).
- **sensagri.m** (MATLAB Code). Main class. It takes the GPR models and the S2 images as input and performs the image processing and subsequently the LAI *green* and LAI *brown* retrieval. Through the code execution, other auxiliary classes are called. Finally, the delivery product is obtained, which is also uploaded to the FTP server.

:\..\Software_LAIGB_v0.3

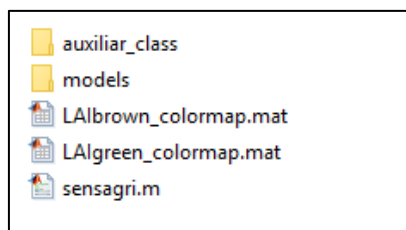


Figure 2. Content of the software package folder Software_LAIGB_v0.3.

:\..\Software_LAIGB_v0.3\auxiliar_class



Figure 3. Content of auxiliar_class folder.

:\..\Software_LAIGB_v0.3\models

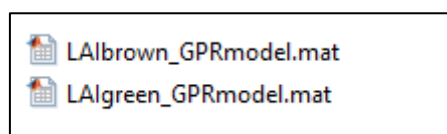


Figure 4. Content of models folder.

4. Instructions

The development and implementation of this final software version was carried out using MATLAB version 2017b. The steps to follow to run the prototype software must be conducted within MATLAB. The software can be run in any MATLAB version newer than 2013. No additional toolboxes are required.

There are several running options to be edited by the user. Each option is defined through a property of the main class *sensagri.m*. Among others, the user can choose if the S2 scene classification band is used as cloud mask or as vegetation mask; or if the output will be saved in GeoTIFF or in ENVI format.

All available configuration options are listed below:

- num_read_lines. Number of image lines per processing block. Default=100.
- image_factor_conv. Image conversion factor. Default=1.0.
- nonegativo. Output product fill-value. Default=NaN.
- tiff_option. 0=output saved as multiband image binary file (.hdr). 1=output product saved as GeoTIFF.
- mask_option. 0=no mask, 1=cloud mask, 2=only vegetation mask. Mask based on the S2 Scene Classification band (SCL).
- umbralG/umbralB. Respective uncertainty threshold applied to each LAI band when plotting the LAIG/LAIB composite map.

1. Open MATLAB. Set MATLAB current working folder to the main software folder **Software_LAIGB_v0.3** (Figure 5).

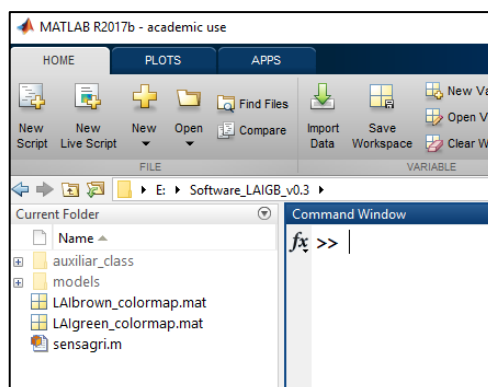


Figure 5. Capture of MATLAB session with the current working directory set to the software package folder.

2. In the Command Window, create an object (*obj*) of the main class **sensagri** using the class name. **As input, the full path of the LAI green and LAI brown GPR models must be provided**, respectively. The constructor function of the *sensagri* class will be executed and the retrieval models imported. Now, “*obj*” has been updated with both GPR models parameters. See example below:

Command Window

```
>>obj=sensagri('..\Software_LAIGB_v0.3\models\LAIgreen_GPRmodel.mat','..\Software_LAIGB_v0.3\examples\LAIgreen_GPRmodel.mat')
```

For changing the default configurations of the processing options, type in the command window *obj.*“*configuration option name*” and assign the corresponding new value , e.g.:

Command Window

```
>>obj.tiff_option=0
```

3. Next, call the *retrieval* method of *sensagri* class using the object *obj*. **The user must provide the full path of two input arguments.** First argument required is the folder containing all S2 images in SAFE format to be processed for LAI *green* and LAI *brown* retrieval. The second argument is the output folder where the user wants the processed delivery product to be saved. If everything went correctly, once the processing ends, *state* should be equal to 0. See example below:

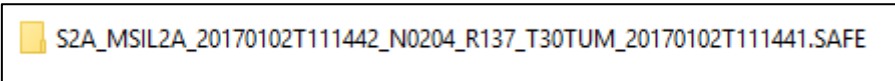
Note. In order to move towards an improved operational algorithm, software version 0.3 already implements an automated processing of S2 images (.SAFE) before applying the retrieval algorithm, so that manual ENVI file conversion is no longer needed. This reduces the user intervention and contributes to run faster the software. The new software version no longer supports input images in ENVI format (binary file + header).

Command Window

```
>>state=obj.retrieval('..\SENSAGRI\input_folder','..\SENSAGRI\output_folder')
```

4. Check if the specified output folder contains the delivery products as expected: (1) a multiband image (data file + header file), (2) a jpg RGB image of the LAI *green/brown* composite map, and (3) a README txt file (*Figure 6*).

:\...\SENSAGRI\input_folder



:\...\SENSAGRI\output_folder

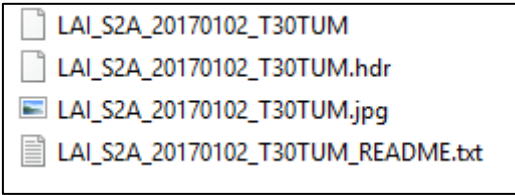


Figure 6. Input folder content and output folder containing the delivery product.