European Commission

EOPEN T7.3: User Training

PUC 2: EOPEN User Training Material Outlook

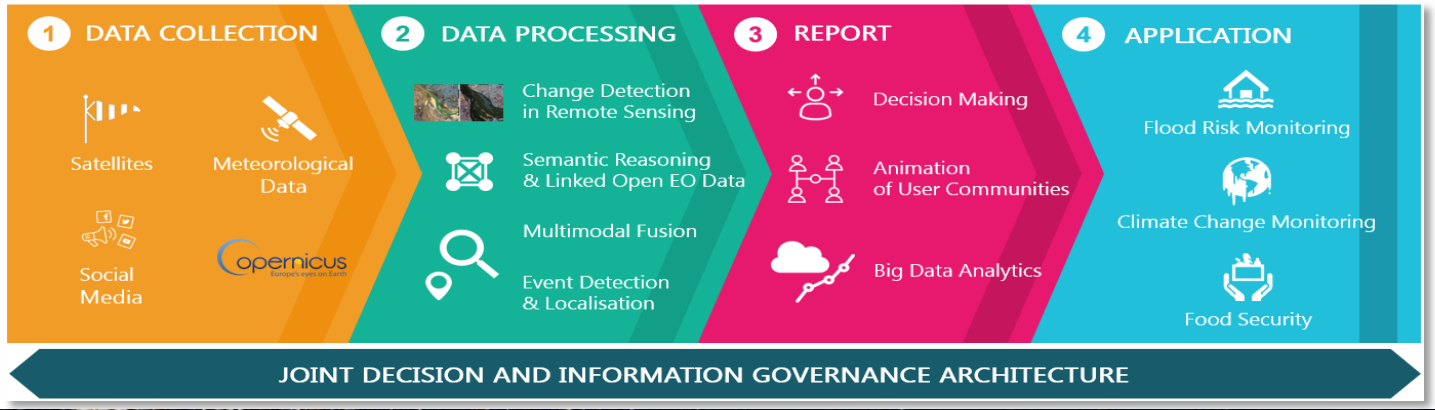
KU-eGIS/RS

Table of Contents

1. Background

1) Background of EOPEN

- EOPEN provides a platform targeting non-expert Earth Observation (EO) data users (non-traditional user communities), experts and the SME community.
- The platform makes Copernicus data and services easy to use for Big Data applications by providing EO data analytics tools, decision making, and infrastructure.
- It includes three PUCs, namely, PUC 1 - flood risk assessment and prevention, PUC 2 - food security through EO datasets and PUC 3 - monitoring climate change through EO, EOPEN provides big data analytics and supports decision making mainly in monitoring agricultural areas.



1. Background

2) Objective of the Training Material

- Allowing the use of the results of EOPEN to end-users, and interested third-parties.
- Providing a detailed description of how the platform works for the efficient use of platform.
- Promoting additional engagement activities, training activities, pursuing a significant widen use of the EOPEN platform, contributing to the uptake of EO data from the community.
- Favoring EOPEN's various capacity building activities that will outlast the project time frame, contributing to the sustainability of EOPEN approach and system.

2. Practical Training

1) Introduction

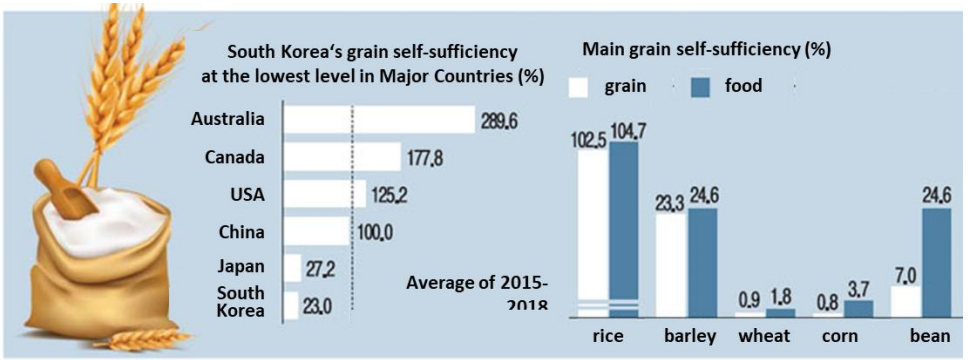
- "Food Security" is a denomination introduced by the Food and Agriculture Organization (FAO) of the United Nations.
- The problem is really complex and comprises several different components
  - Food access
  - Food distribution
  - Food supply stability,
  - Use of food

4 pillars of Food security, 2018, Food security Cluster

2. Practical Training

1) Introduction

- South Korea has low food self-sufficiency that is decreasing in the long term due to being dependent on the import of most major grains.
- South Korea is maintaining 50% of food self-sufficiency thanks to the influence of high self-sufficiency of rice, but grain self-sufficiency rate has decreased to 20% due to the increasing import of feed grain.
- Domestic rice production decreased from 4,103,135 tons in 2011 to 3,763,340 tons in 2018 and the future output of rice predicted by Korea Rural Economic Institute (KREI) is expected to decrease due to the effects of climate change.



MAFRA(left), KREI(right), 2016

2. Practical Training

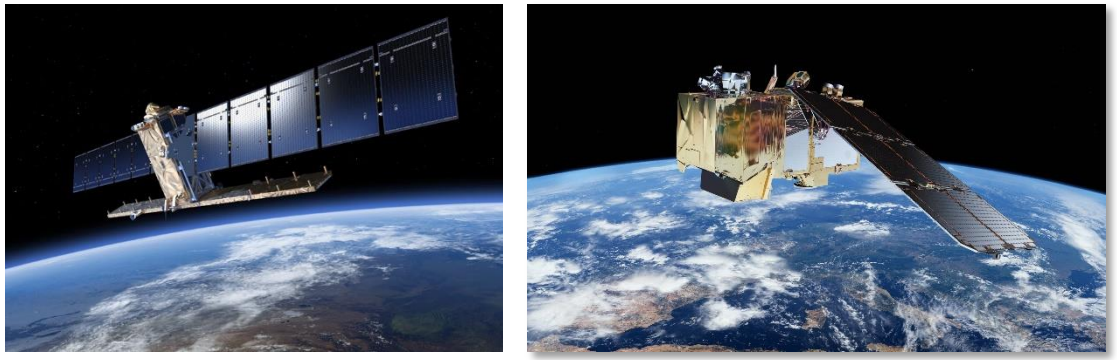
1) Introduction

- Satellite data have been applied on the occurrence of severe agriculture events since 1972.
- It is applied to agriculture in several ways such as a means of estimating crop yields.
- It can provide an accurate picture of the areas being cultivated, while also differentiating between crop types and determining their health and maturity.
- This information helps to inform the market, and provide early warning of crop failure or famine.
- Satellites are used as a management tool through the practice of precision agriculture, where satellite images are used to characterize a farmer's fields.
- Although early prediction of rice production using satellite images is carried out as the basis for the decision-making for improving rice supply and demand stability, low resolution satellites used in most previous studies have limitations in observing fragmented land in Asia and Europe.

2. Practical Training

1) Introduction

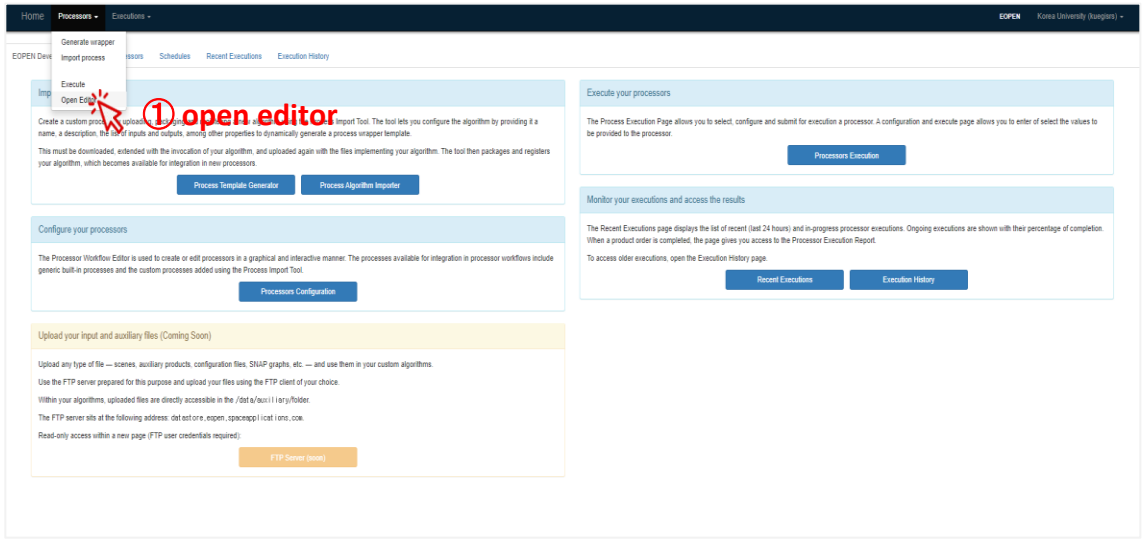
- Copernicus program including Sentinel missions is the most ambitious Earth observation initiative and can have a great impact and contribution also in the field of food security.
- We implemented rice mapping for yield production using Sentinel-1 and 2 satellites.
- This training material shows two different approaches for classification of rice based on Recurrent Neural Network (RNN) and Random Forest .
- Two different methods are produced to examine the relevant differences in processing complexity, accuracy, and generalization.



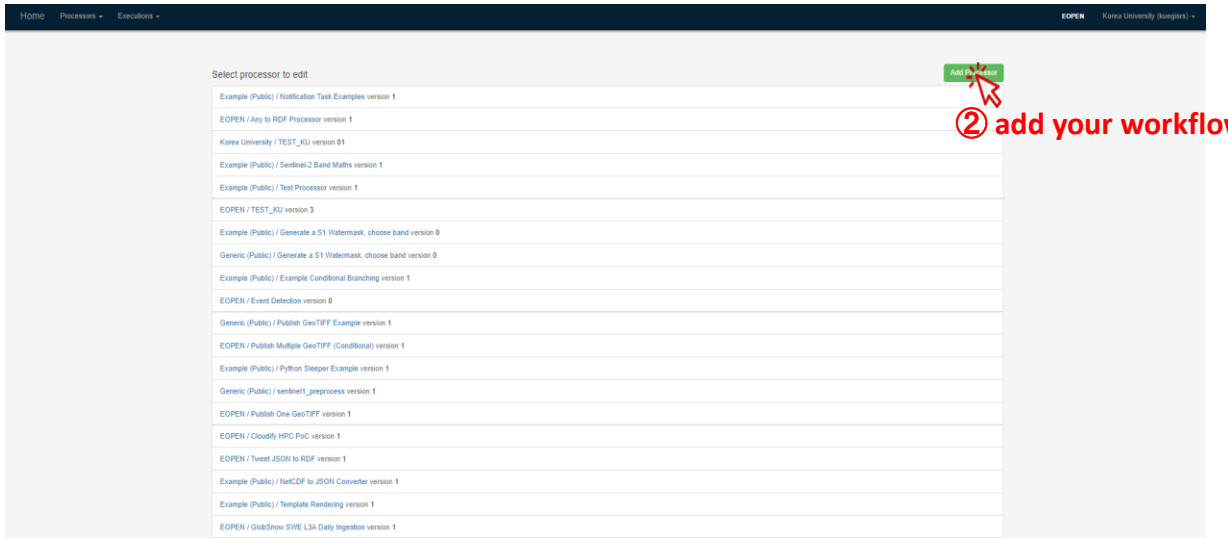
Sentinel-1 (left), 2 (right), ESA

2. Practical Training

2) Method 1 (RNN) – Getting started



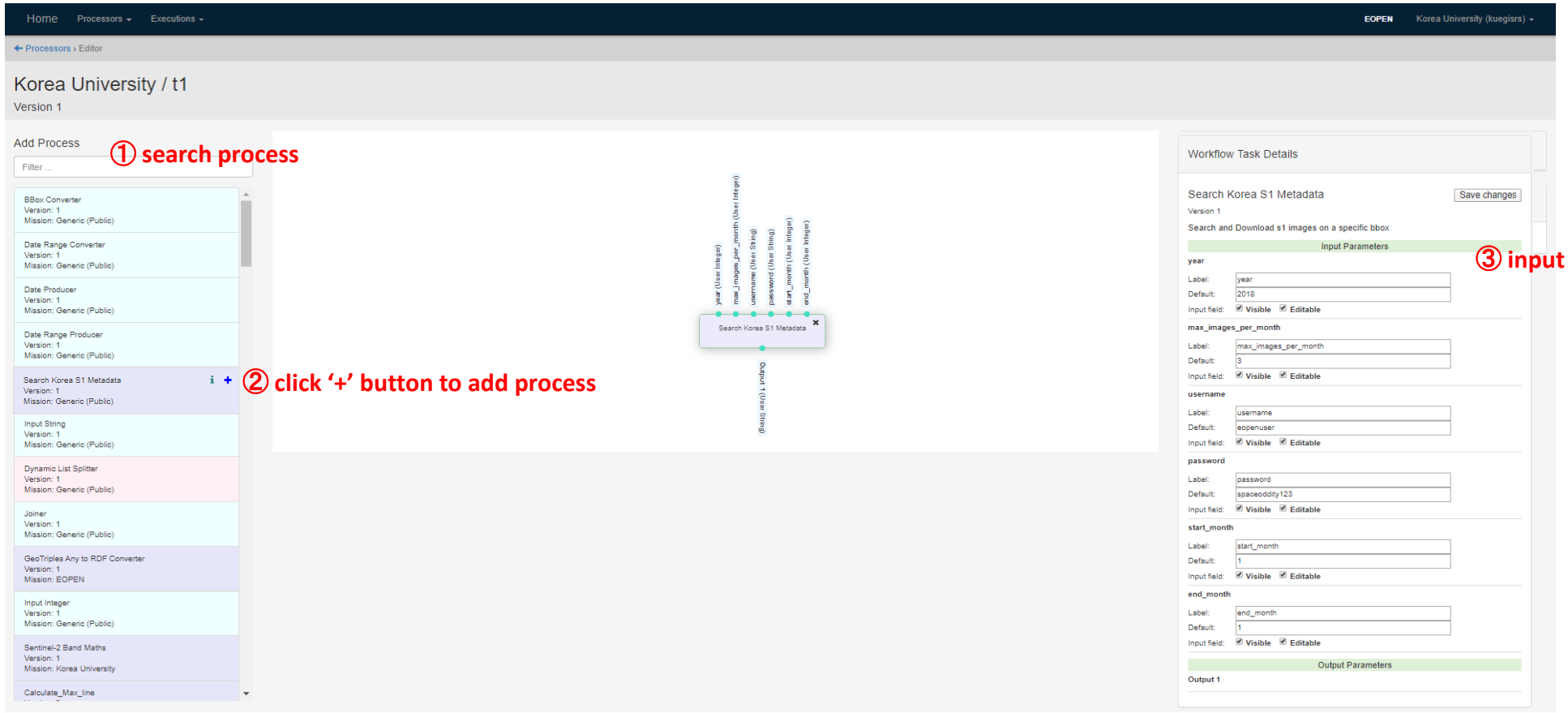
EOPEN Developer Panel



Select or Add processor to edit

2. Practical Training

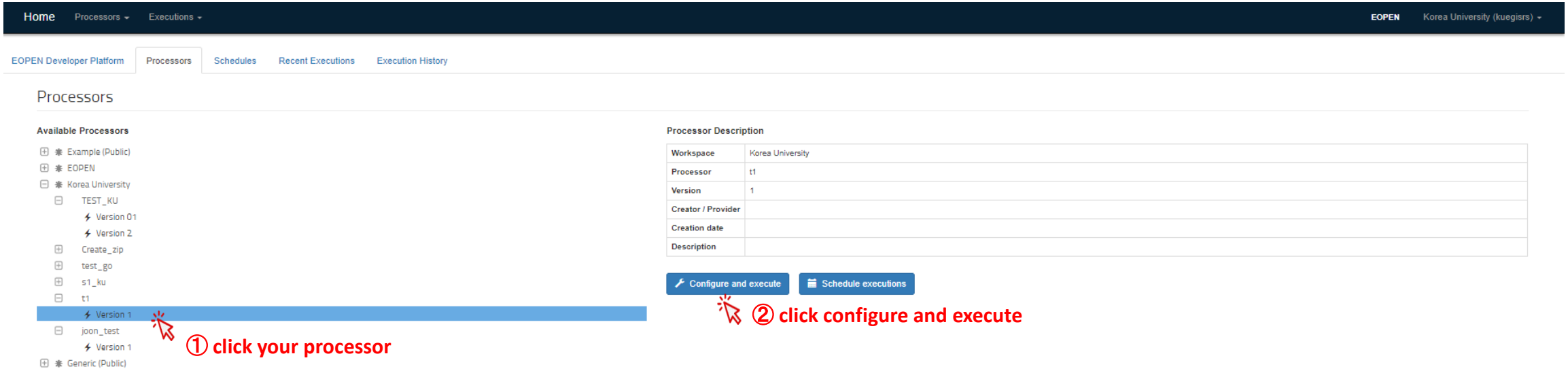
2) Method 1 (RNN) – Getting started



Editor of process

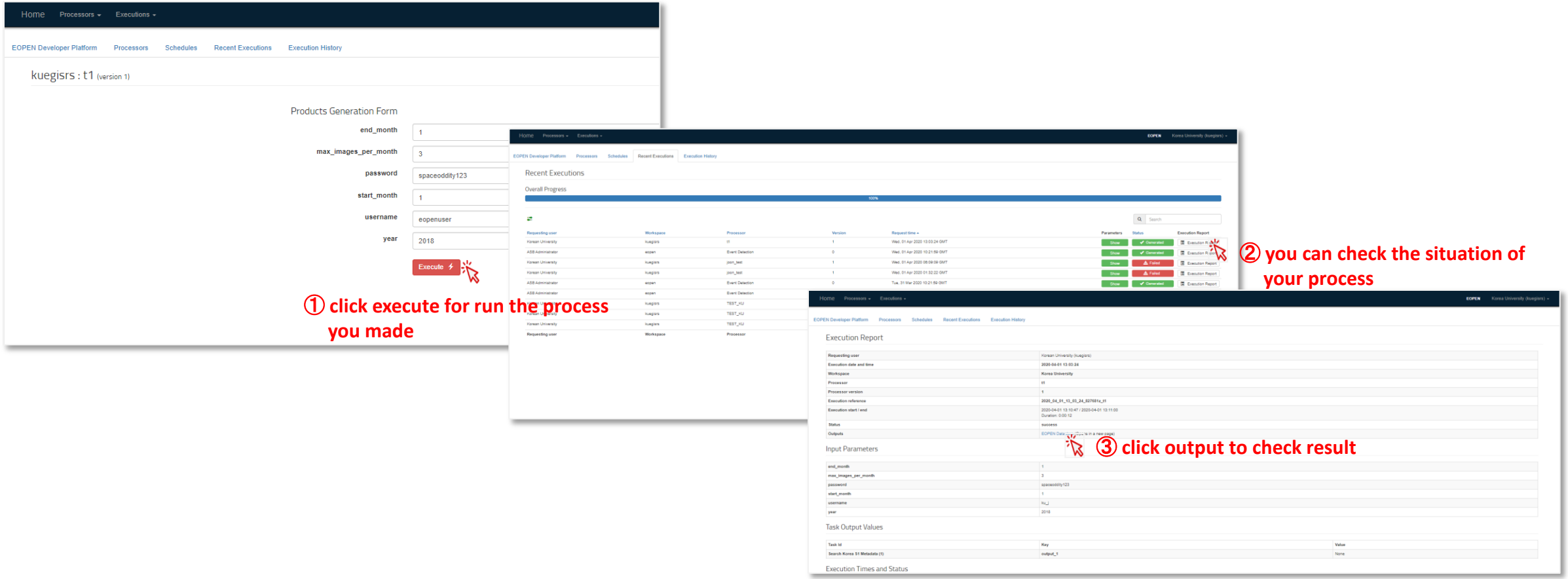
2. Practical Training

2) Method 1 (RNN) – Getting started



2. Practical Training

2) Method 1 (RNN) – Getting started



2. Practical Training

2) Method 1 (RNN) – Getting started

Structure

Demonstrating overall structure of each workflows.
RNN method 1 consists of four workflows.
1. Download sentinel-1 products
2. Preprocess sentinel-1 products
3. Rice paddy detection
4. RNNs model training

Parameter

Explaining parameters of each processes.

Explanation

Providing further information about structures and parameters, such as baseline algorithm and way to accessing the output file.

2. Practical Training

2) Method 1 (RNN) – Getting started

- The Processor and Workflow concepts have been merged in favour of Workflow. This removes the ambiguity that existed between a resource type (Processor) and its definition (Workflow).
- A user who creates a Process or a Workflow is automatically registered as its owner. By default, Processes and Workflows are only visible and may only be managed by their owner.
- The Workspace and user Role concepts have been introduced to allow sharing resources, including Processes and Workflows. The fundamental rule is that a particular resource is only visible by the users who are given a role in one of the workspace the resource belongs to. To share a resource, a user who has the right to do so assigns that resource to one or more workspaces.

2. Practical Training

2) Method 1 (RNN) – 1. Basic processes

- Deep learning based rice paddy detection consists of the following workflow and processes

Download Sentinel-1 products…

- Search Korea S1 Metadata

Preprocess Sentinel-1 products…

- Sentinel 1 Preprocessing

2. Practical Training

2) Method 1 (RNN) – 1. Basic processes

- Deep learning based rice paddy detection consists of the following workflow and processes
  - Input String/Integer: Provide string/integer as an input

Rice paddy detection…

  - Monthly_mosaic: Mosaic downloaded images to produce a Monthly S-1 mosaic
  - rp_detection: Apply RNN model to the time-series array to detect rice paddies

RNNs model training…

  - Time_series_list: Gather input image files with a provided regular expression
  - import_x_y: Concatenate image files and labeled data into a time-series array
  - Separate_Tr_Va: Divide the time-series array into training and validation data
  - concatenate_set: Merge multiple time-series arrays into a single array
  - RNN: Train RNNs model with the provided time-series arrays

2. Practical Training

2) Method 1 (RNN) – 2. Download Sentinel-1 Products

Structure

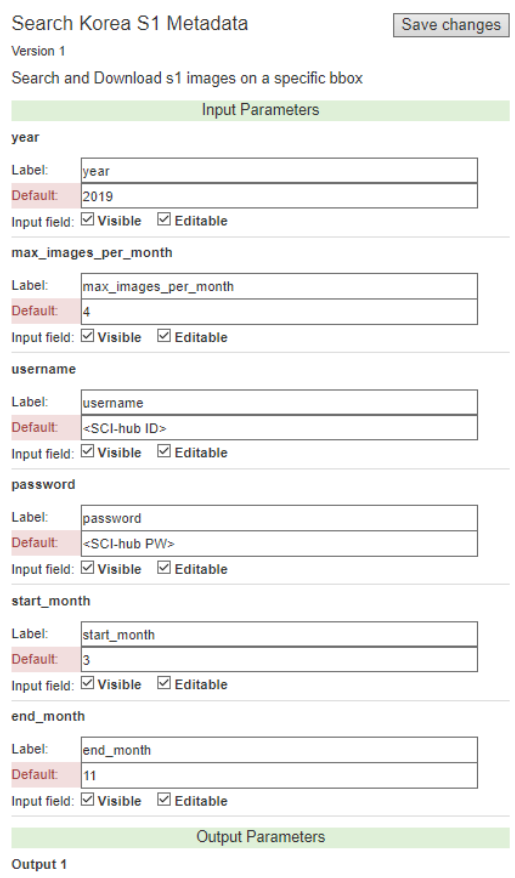
Parameter

▷ The workflow consists of a single process
: Search Korea S1 Metadata

- year: Target year for searching images
- max_imges_per_month: Maximum number of images in a month
- username/password: ID and password for https://scihub.copernicus.eu/dhus/#/home
- start_month/end_month: Set a searching period

The rice paddy detection algorithm requires images from March to November

The area of interest is Dangjin, South Korea



2. Practical Training

2) Method 1 (RNN) – 2. Download Sentinel-1 Products

Explanation

- The images are downloaded at https://proto1.eopen.spaceapplications.com/public/noa_test/Download/

Index of /dev/public/noa_test/Download/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
S1B_IW_GRDH_1SDV_20190714T213152_20190714T213217_017135_0203C2_1BE0.SAFE/	-	2019-Oct-15 08:06
S1B_IW_GRDH_1SDV_20181128T213149_20181128T213214_013810_01998E_1CD7.SAFE/	-	2019-Oct-15 09:46
S1B_IW_GRDH_1SDV_20181128T213124_20181128T213149_013810_01998E_A05F.SAFE/	-	2019-Oct-15 01:49
S1B_IW_GRDH_1SDV_20181023T213125_20181023T213150_013285_0188FB_74A9.SAFE/	-	2019-Oct-15 09:44
S1B_IW_GRDH_1SDV_20181016T213946_20181016T214001_013183_0185CB_5F78.SAFE/	-	2019-Oct-15 10:06
S1B_IW_GRDH_1SDV_20181011T213150_20181011T213215_013110_018390_3D56.SAFE/	-	2019-Oct-15 08:29
S1B_IW_GRDH_1SDV_20180917T213124_20180917T213149_012760_0178E3_ADF3.SAFE/	-	2019-Oct-15 09:19
S1B_IW_GRDH_1SDV_20180905T213124_20180905T213149_012585_017386_A66E.SAFE/	-	2019-Oct-15 09:45
S1B_IW_GRDH_1SDV_20180829T213944_20180829T214000_012483_01705B_FC01.SAFE/	-	2019-Oct-15 01:50
S1B_IW_GRDH_1SDV_20180719T213147_20180719T213212_011885_015E06_A8E7.SAFE/	-	2019-Oct-15 07:43
S1B_IW_GRDH_1SDV_20180625T213145_20180625T213210_011535_015332_5A4B.SAFE/	-	2019-Oct-15 07:21
S1B_IW_GRDH_1SDV_20180613T213144_20180613T213209_011360_014DC6_E984.SAFE/	-	2019-Oct-15 06:59
S1B_IW_GRDH_1SDV_20180601T213143_20180601T213208_011185_01485B_78E0.SAFE/	-	2019-Oct-15 06:34

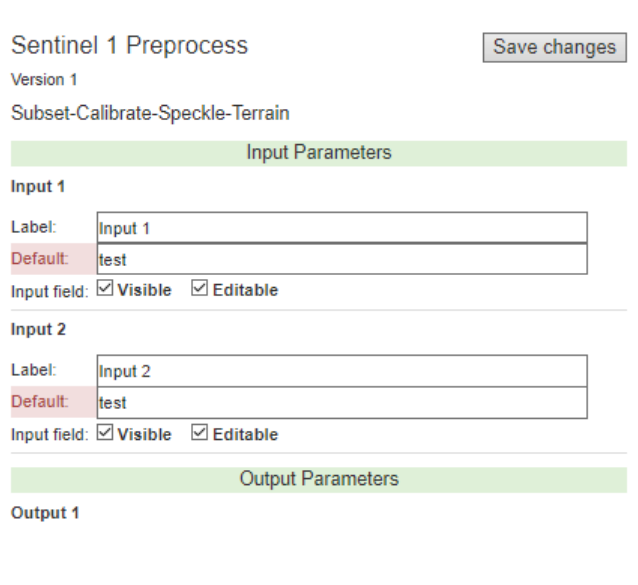
2. Practical Training

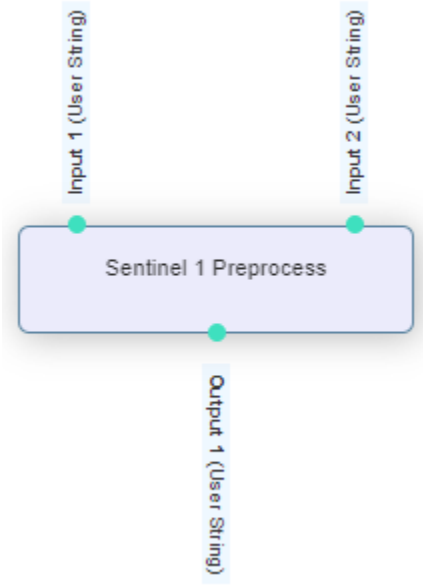
2) Method 1 (RNN) – 3. Preprocess Sentinel-1 Products

Structure
Parameter

▷ The workflow consists of a single process: Sentinel 1 Preprocess

- Input 1: String “test” is required in the current version
- Input 2: String “test” is required in the current version





2. Practical Training

2) Method 1 (RNN) – 3. Preprocess Sentinel-1 Products

Explanation

▷ A list of preprocessing is applied to the downloaded images

- Apply orbit file
- Thermal noise removal
- Radiometric calibration
- Terrain correction

▷ The preprocessed products are saved at https://proto1.eopen.spaceapplications.com/public/noa_test/Download/*<Image Name>*/Results/

Index of /dev/public/noa_test/Download/S1B_IW_GRDH_1SDV_20190714T213152_20190714T213217_017135_0203C2_1BE0.SAFE/Results/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
Processed_VV.tif	572.2 MiB	2019-Oct-15 08:23
Processed_VH.tif	572.2 MiB	2019-Oct-15 08:12

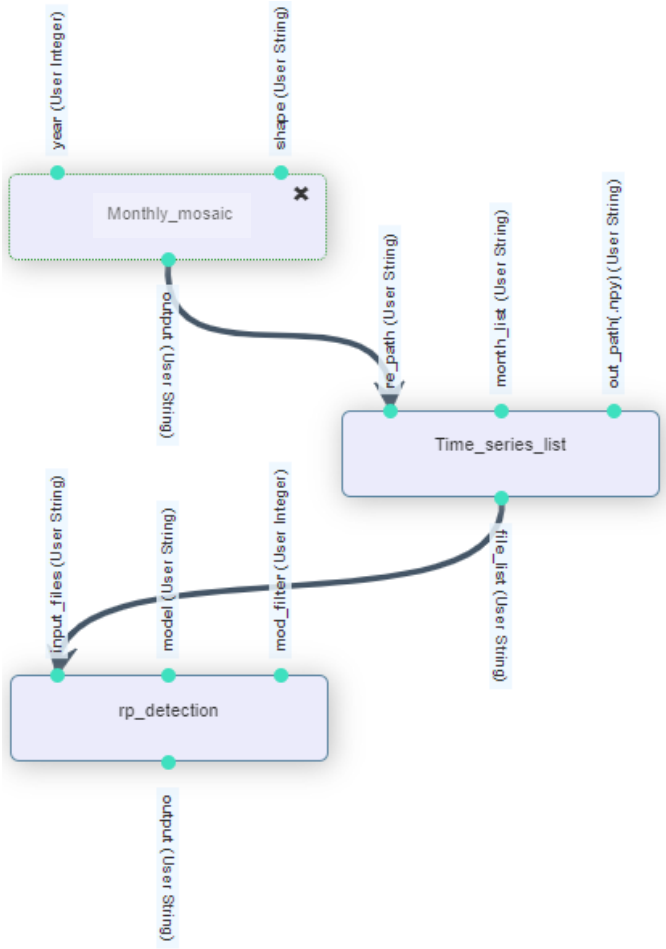
2. Practical Training

2) Method 1 (RNN) – 4. Rice Paddy Detection

Structure

▷ The workflow consists of following processes: Monthly_mosaic, Time_series_list, rp_detection

- "Monthly_mosaic" mosaics the downloaded images to produce monthly mean composite images and delivers a temporary output path, which contains the monthly composite images, to the "Time_series_list"
- "Time_series_list" produces a list of input data with the provided path
- "rp_detection" produces binary raster files, which maps rice paddy, by processing the time-series Sentinel images in the provided list



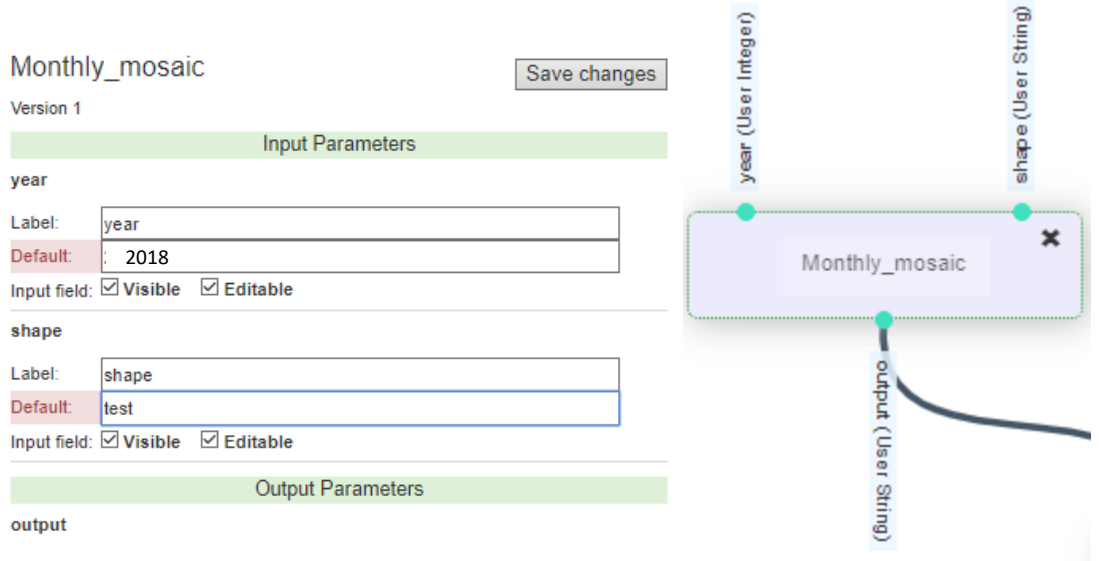
2. Practical Training

2) Method 1 (RNN) – 4. Rice Paddy Detection

Parameter

▷ Monthly_mosaic

- year: Target year of rice paddy detection (Preprocessed time series images are required)
- shape: Boundary of interested area where mean value composite for producing time series data will be processed
  (Put “test” for setting the boundary to Dangjin city)



2. Practical Training

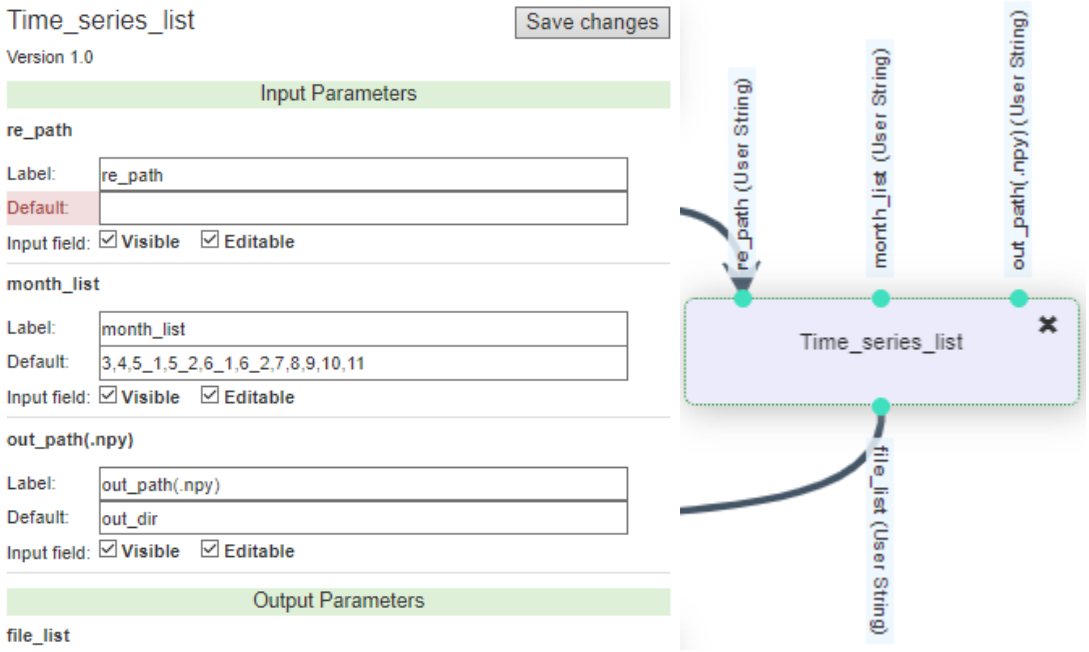
2) Method 1 (RNN) – 4. Rice Paddy Detection

▷ Time_series_list

- re_path: Regular Expression for producing a list of input data (Take input from “make _timeseries”)
- Month_list: A list of months to be analyzed (From March to November are needed, while May and June need to be separated into 2 part as in the figure)
- Out_path: A path where the list of file name to be saved (Put “out_dir” for saving it as a temporary file”)

*Example: re_path = ‘./PUC_2/*_’, Month_list = ‘4, 5_1’
Images to be searched = ‘./PUC_2/*_4.tif’ and ‘./PUC_2/*_5_1.tif’

Parameter



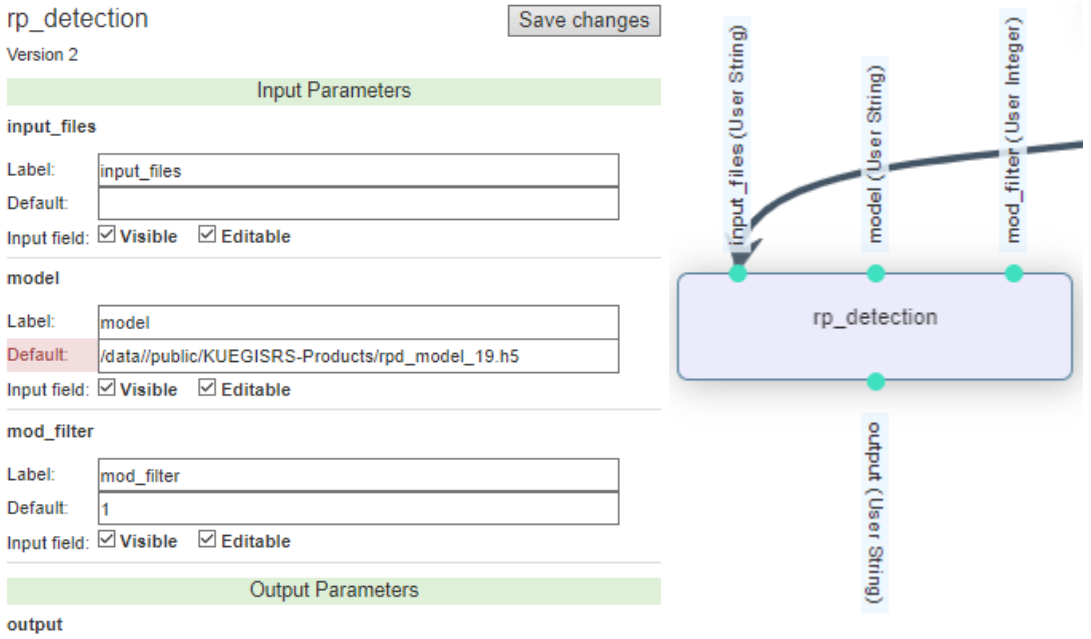
2. Practical Training

2) Method 1 (RNN) – 4. Rice Paddy Detection

▷ rp_detection

- input_files: A list of files to be analyzed (Take input form “Time_series_list”)
- model: Trained RNNs model
- mod_filter: Number of applying majority filter, which will reduce salt and pepper noise

Parameter



2. Practical Training

2) Method 1 (RNN) – 4. Rice Paddy Detection

Explanation

▷ Binary rice paddy detection result is produced (Execution Report -> EOPEN Datastore -> ./outputs)

Requesting user	Workspace	Processor	Version	Request time ▲	Parameters	Status	Execution Report
Korean University	kuegisrs	TEST_KU	2	Tue, 10 Mar 2020 09:42:56 GMT	Show	Generated	Execution Report

EOPEN Developer Platform | Processors | Schedules | Recent Executions | Execution History

Execution Report

Requesting user	Korean University (kuegisrs)
Execution date and time	2020-03-10 09:42:56
Workspace	Korea University
Processor	TEST_KU
Processor version	2
Execution reference	2020_03_10_09_42_56_647570z_test_ku
Execution start / end	2020-03-10 09:45:45 / 2020-03-10 10:00:40 Duration: 0:14:55
Status	success
Outputs	EOPEN Datastore (Opens in a new page)

Index of /processor-run-2020_03_10_09_42_56_647570z_test_ku/wps-run-kuegisrs-rp-detection-2-f58d78d1-661a-471d-bd31-fe0f697f41ff/outputs/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
rp_detection.tif	70.5 MiB	2020-Mar-10 10:01

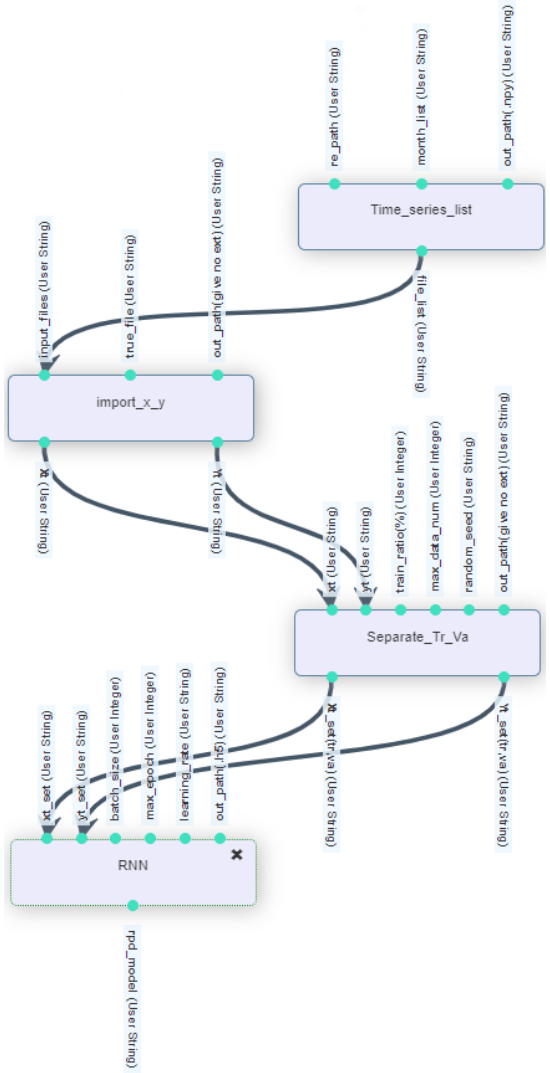
2. Practical Training

2) Method 1 (RNN) – 5. RNNs Model Training

Structure

▷ The workflow consists of the following processes
: Time_series_list, import_x_y, separate_Tr_Va, RNN

- "Time_series_list" produces a list of input data with the provided regular expression
- "import_x_y" concatenate the listed image files and labeled data into a time-series array
- "Separate_Tr_Va" divide the time-series array into training and validation data
- "RNN" train RNNs model with the provided training and validation data



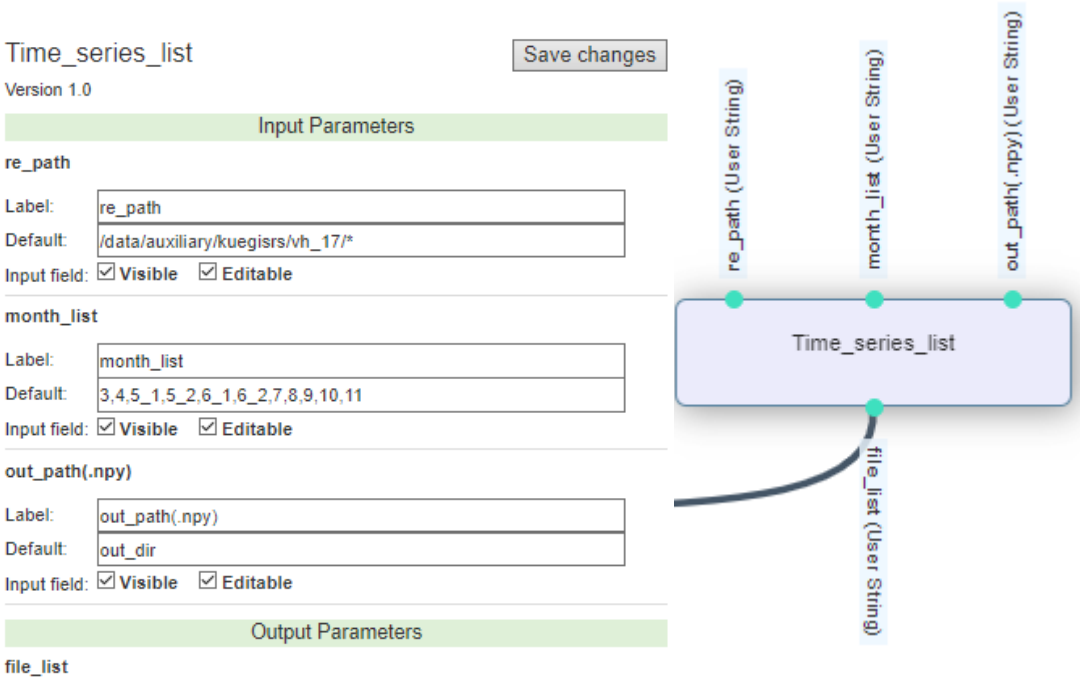
2. Practical Training

2) Method 1 (RNN) – 5. RNNs Model Training

Parameter

▷ Time_series_list

- re_path: Regular Expression for producing a list of input data
  (The input data should be compatible with a labeling data, which will be provided at the next process)
- Month_list: A list of months to be analyzed
  (From March to November are needed, while May and June need to be separated into 2 part as in the figure)
- out_path: A path where the list of file name to be saved
  ("out_dir" for saving it as a temporary file")



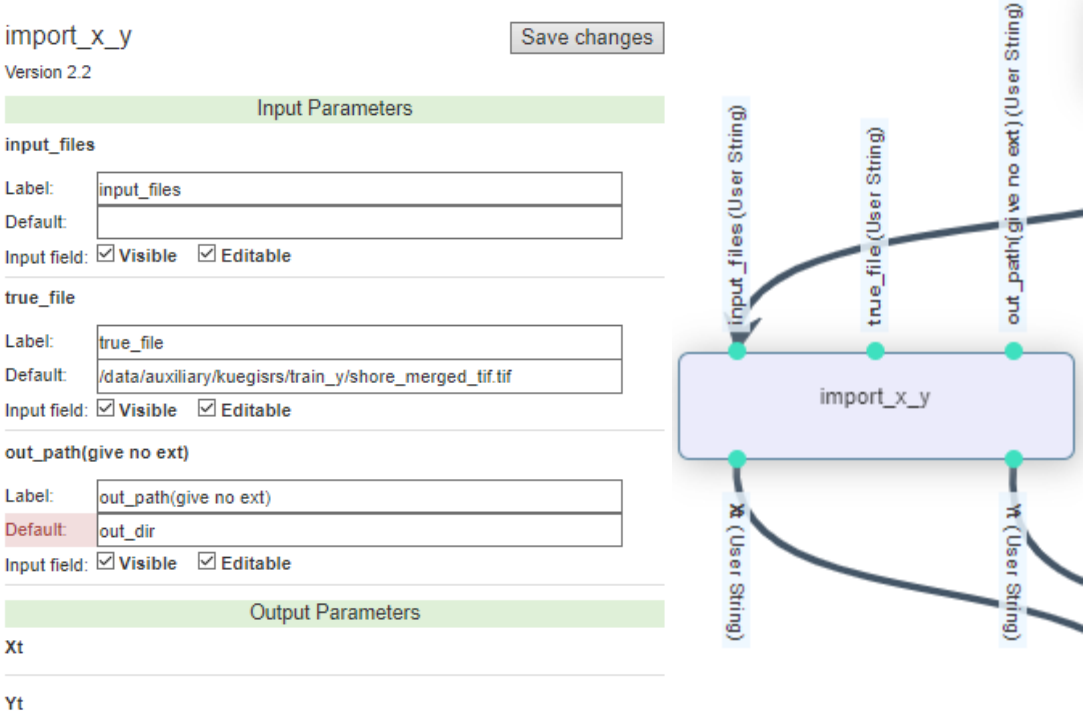
2. Practical Training

2) Method 1 (RNN) – 5. RNNs Model Training

▷ import_x_y

- input_files: A list of image files to be analyzed (Take input form “Calculate_Max_line”)
- true_file: A rice paddy labeling data (0: Others, 1: Rice paddy, 2: Rice paddy-optional)
- out_path: A path where the image data(x) and labeling data(y) to be saved (No extension needed. The out put will be generated as “./~xt.npy” and “./~yt.npy”)

Parameter



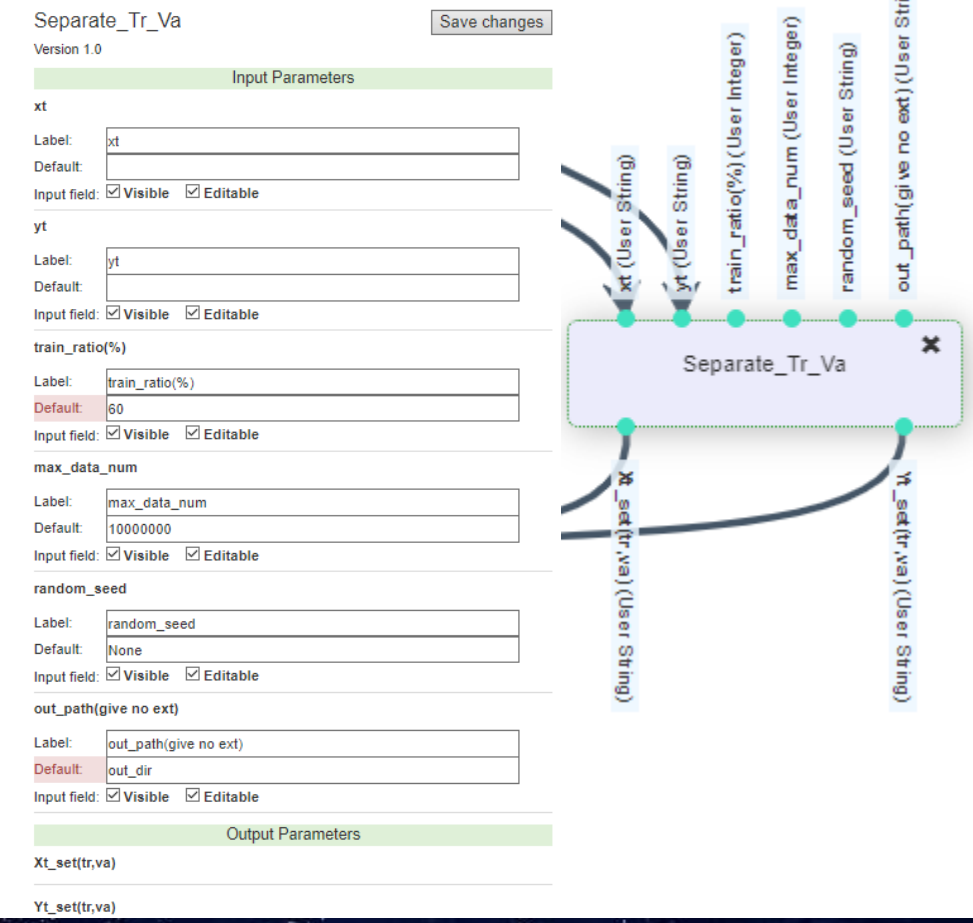
2. Practical Training

2) Method 1 (RNN) – 5. RNNs Model Training

▷ Separate_Tr_Va

- xt/yt: Array of image/labeling data which is provided by “import_x_y”
- train_ratio: Percentage of training data (The rest of data will be used as validation data)
- max_data_num: Limit the number of array(pixel) not to exceed memory capacity
- random_seed: Random seed for array separation
- out_path: A path where data to be saved, separated into training and validation data (No extension needed. The out put will be generated as “./~xt.npz” and “./~yt.npz”)

Parameter



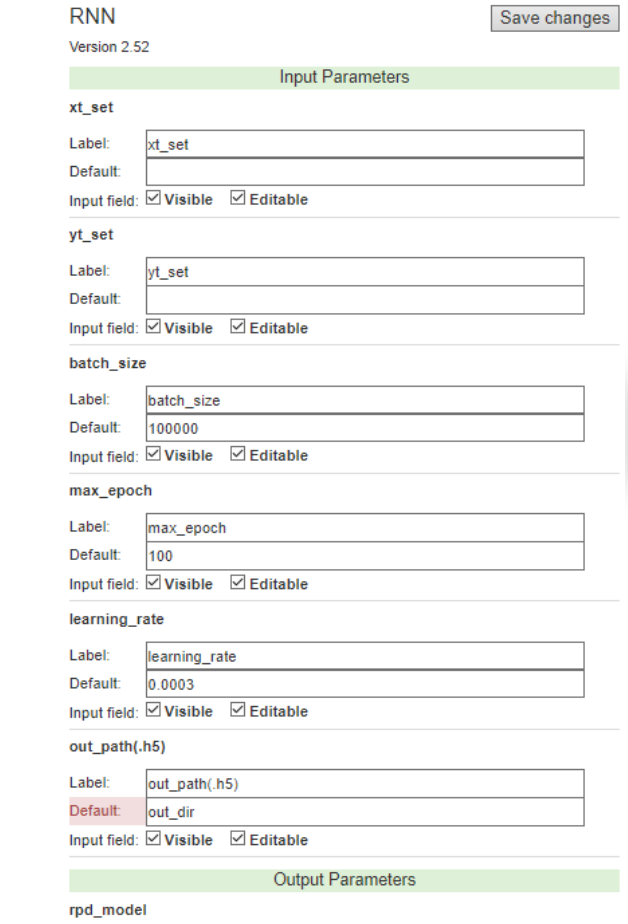
2. Practical Training

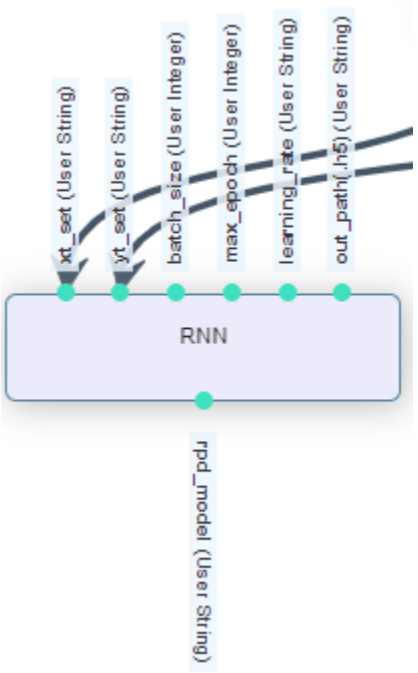
2) Method 1 (RNN) – 5. RNNs Model Training

▷ RNN

- xt_set/yt_set: Sets of x/y data which is provided by “Separate_Tr_Va” (training/validation zipped)
- batch_size: Batch size of training (hyper-parameter, default: 100000)
- max_epoch: Maximum number of training epoch. Training will stop if loss does not decrease (hyper-parameter, default: 100)
- learning_rate: Scale of parameter update per training (hyper-parameter, default, 0.0003)
- out_path: A path where trained RNN model to be saved (“out_dir” for saving it as a temporary file)

Parameter





2. Practical Training

2) Method 1 (RNN) – 5. RNNs Model Training

Explanation

▷ Trained model is produced (Designated path or Execution Report -> EOPEN Datastore -> ./outputs)

Requesting user	Workspace	Processor	Version	Request time ▲	Parameters	Status	Execution Report
Korean University	kuegisrs	TEST_KU	2	Tue, 10 Mar 2020 09:42:56 GMT	Show	✔ Generated	Execution Report

EOPEN Developer Platform | Processors | Schedules | Recent Executions | Execution History

Execution Report

Requesting user	Korean University (kuegisrs)
Execution date and time	2020-03-10 09:42:56
Workspace	Korea University
Processor	TEST_KU
Processor version	2
Execution reference	2020_03_10_09_42_56_647570z_test_ku
Execution start / end	2020-03-10 09:45:45 / 2020-03-10 10:00:40 Duration: 0:14:55
Status	success
Outputs	EOPEN Datastore (Opens in a new page)

Index of /processor-run-2020_03_10_11_31_06_564688z_test_ku/wps-run-kuegisrs-rnn-2-52-30f479bf-1959-4dbd-bc67-602250a156dd/outputs/

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
rpd_model_14.h5	155.7 KiB	2020-Mar-10 12:11
rpd_model_13.h5	155.7 KiB	2020-Mar-10 12:11

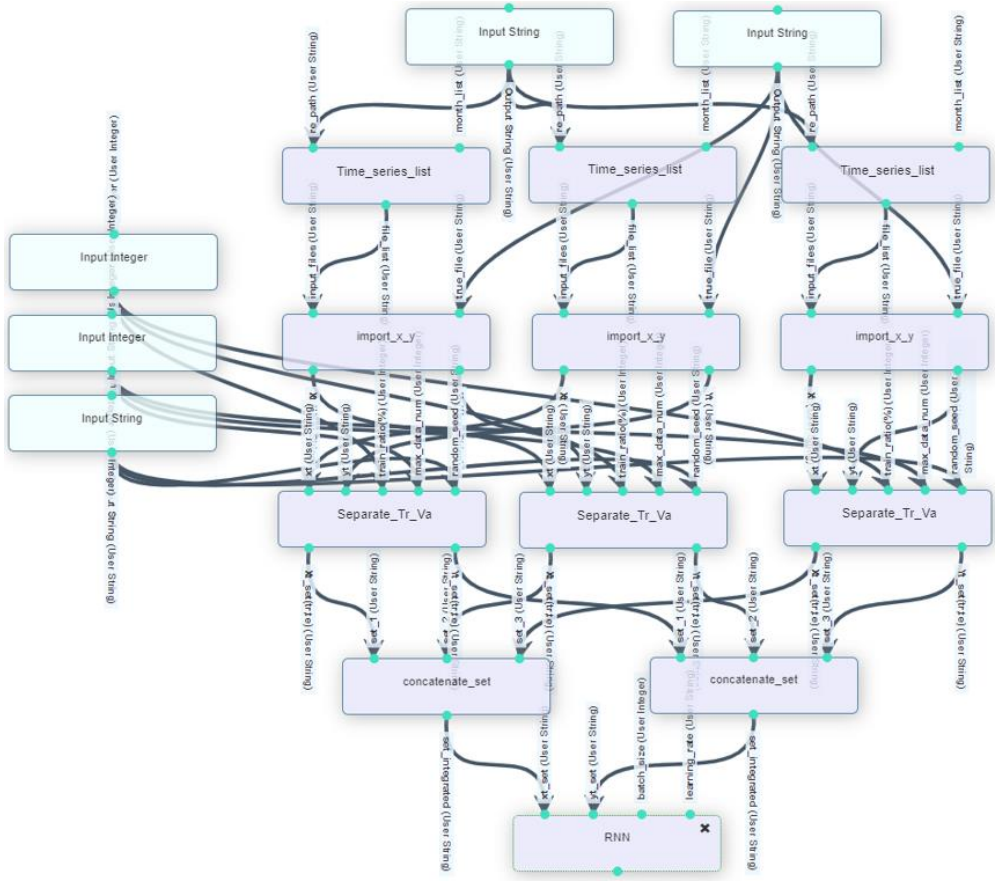
2. Practical Training

2) Method 1 (RNN) – 6. RNNs Model Training_DA

▷ The processor consists of following processes : Input String, Time_series_list, import_x_y, Separate_Tr_Va, concatenate_set, RNN

- The structure is a modified version of the previous processor with Data Augmentation technique
- Three different training/validation sets are produced and concatenated before training RNNs model
- “Input String/Integer” efficiently provides same input to multiple processes
- “concatenate_set” merges diverse sets into a single set (maximum 3 sets)

Structure



2. Practical Training

2) Method 1 (RNN) – 6. RNNs Model Training_DA

Parameter

▷ Input String -> Time_series_list

Input String — Save changes

Version 1

Generic built-in task bypassing a single string to the next processes

Input Parameters

Input String

Any input string. Customize the properties (label, default value, etc.) in specific processors.

Label:	Input path
Default:	/data/auxiliary/kuegisrs/vh_17/*

Input field: ☑ Visible ☑ Editable

Output Parameters

Output String

month_list

Label:	month_list
Default:	vh_2,3,4,5_1,5_2,6_1,6_2,7,8,9,10

Input field: ☑ Visible ☑ Editable

month_list

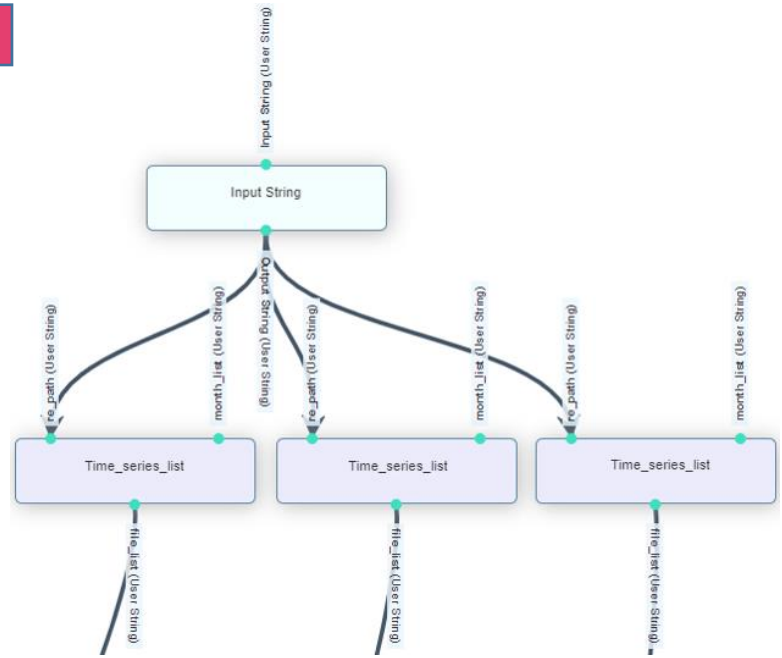
Label:	month_list
Default:	3,4,5_1,5_2,6_1,6_2,7,8,9,10,11

Input field: ☑ Visible ☑ Editable

month_list

Label:	month_list
Default:	4,5_1,5_2,6_1,6_2,7,8,9,10,11,12

Input field: ☑ Visible ☑ Editable



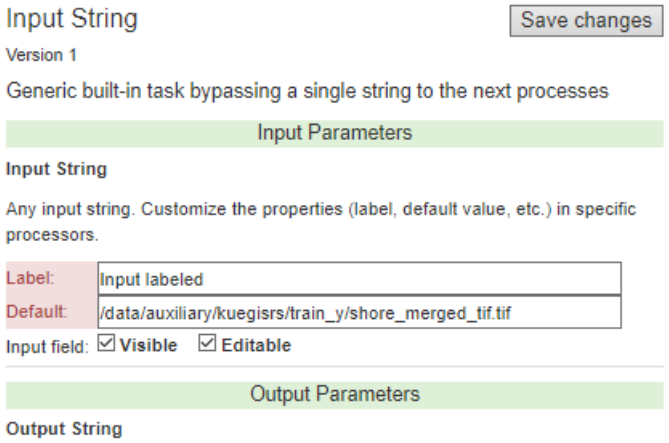
- Input String: Regular Expression for producing a list of input data, which will be commonly used in 3 different “Time_series_list”
- month_list: A list of time series to be analyzed (Feb-Oct, Mar-Nov, Apr-Dec)
- out_path: A path where the list of file name to be saved (Linked to input_files of “import_x_y”)

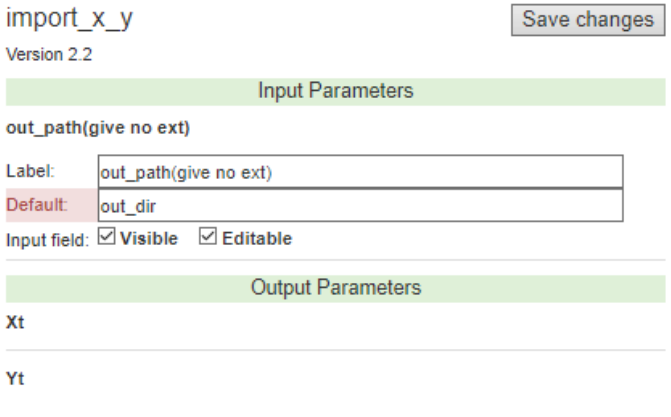
2. Practical Training

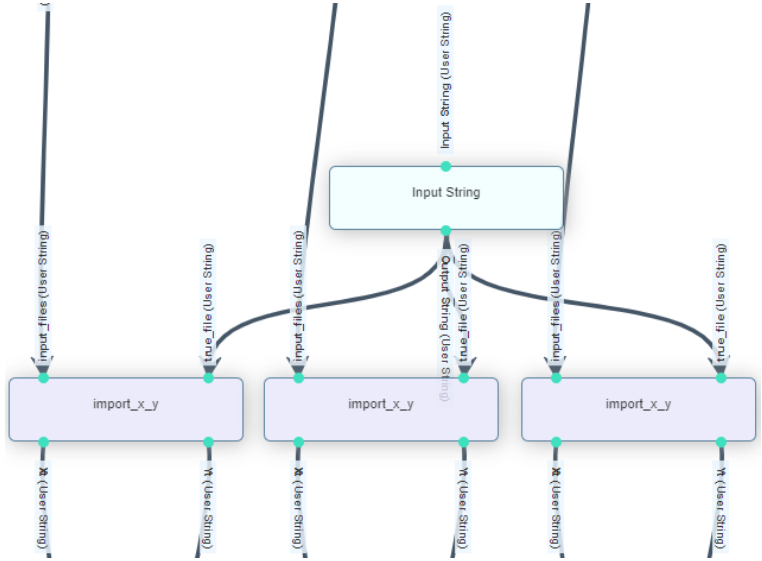
2) Method 1 (RNN) – 6. RNNs Model Training_DA

Parameter

▷ Time_series_list & Input String -> import_x_y







- Input String: A rice paddy labeling data, which will be commonly used in 3 different “import_x_y” (0: Others, 1: Rice paddy, 2: Rice paddy-optional)
- input_files: A list of image files to be analyzed (Take input form “Time_series_list”)
- out_path: A path where the image data(x) and labeling data(y) to be saved (Linked to xt/yt of “Separate_Tr_Va”)

2. Practical Training

2) Method 1 (RNN) – 6. RNNs Model Training_DA

Parameter

▷ import_x_y & Input Integer/String -> Separate_Tr_Va

Integer

Label: Train_perc
Default: 40
Input field: ☑ Visible ☑ Editable

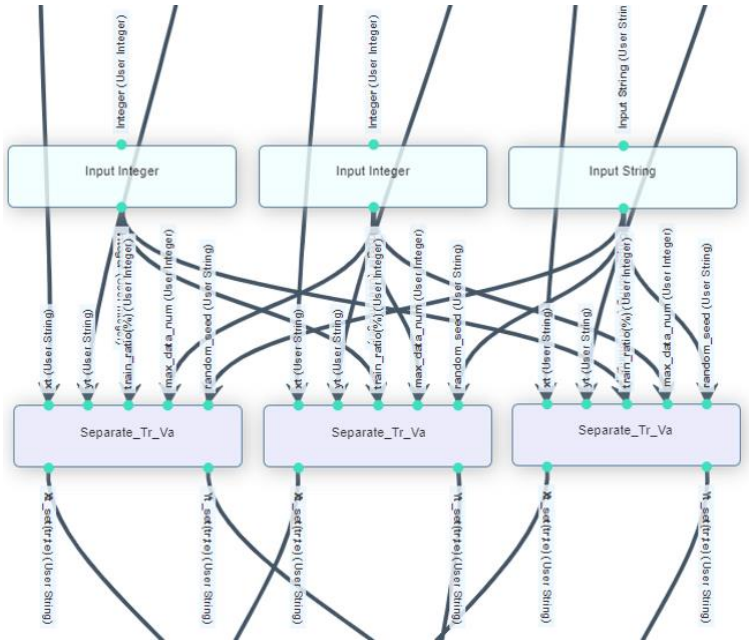
Integer

Label: max_data_num
Default: 5000000
Input field: ☑ Visible ☑ Editable

Input String

Any input string. Customize the properties (label, default value, etc.) in specific processors.

Label: Random_seed
Default: 0
Input field: ☑ Visible ☑ Editable

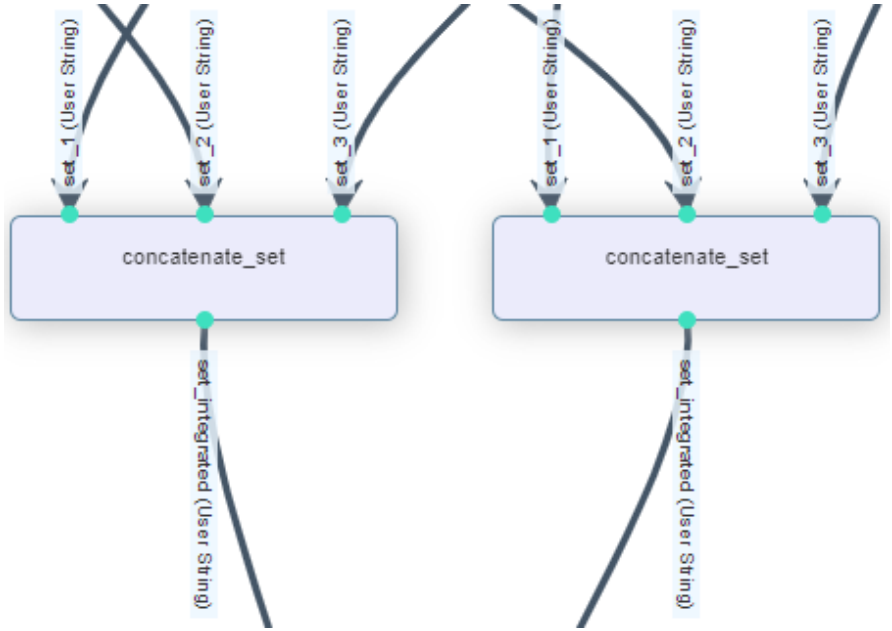


- xt/yt: Array of image/labeling data which is provided by “import_x_y”
- Input Integer: Provide common training percent to 3 “Separate_Tr_Va”
- Input Integer: Provide common max_data_num to 3 “Separate_Tr_Va”
- Input String: Provide common random_seed to 3 “Separate_Tr_Va”

2. Practical Training

2) Method 1 (RNN) – 6. RNNs Model Training_DA

Parameter



▷ Separate_Tr_Va -> concatenate_set

concatenate_set Save changes

Version 1

Input Parameters

set_1

Label: set_1
Default: None
Input field: ☑ Visible ☑ Editable

set_2

Label: set_2
Default: None
Input field: ☑ Visible ☑ Editable

set_3

Label: set_3
Default: None
Input field: ☑ Visible ☑ Editable

Output Parameters

set_integrated

- One “concatenate_set” takes all of xt from 3 “Separate_Tr_Va”
- The other “concatenate_set” takes all of yt from 3 “Separate_Tr_Va”

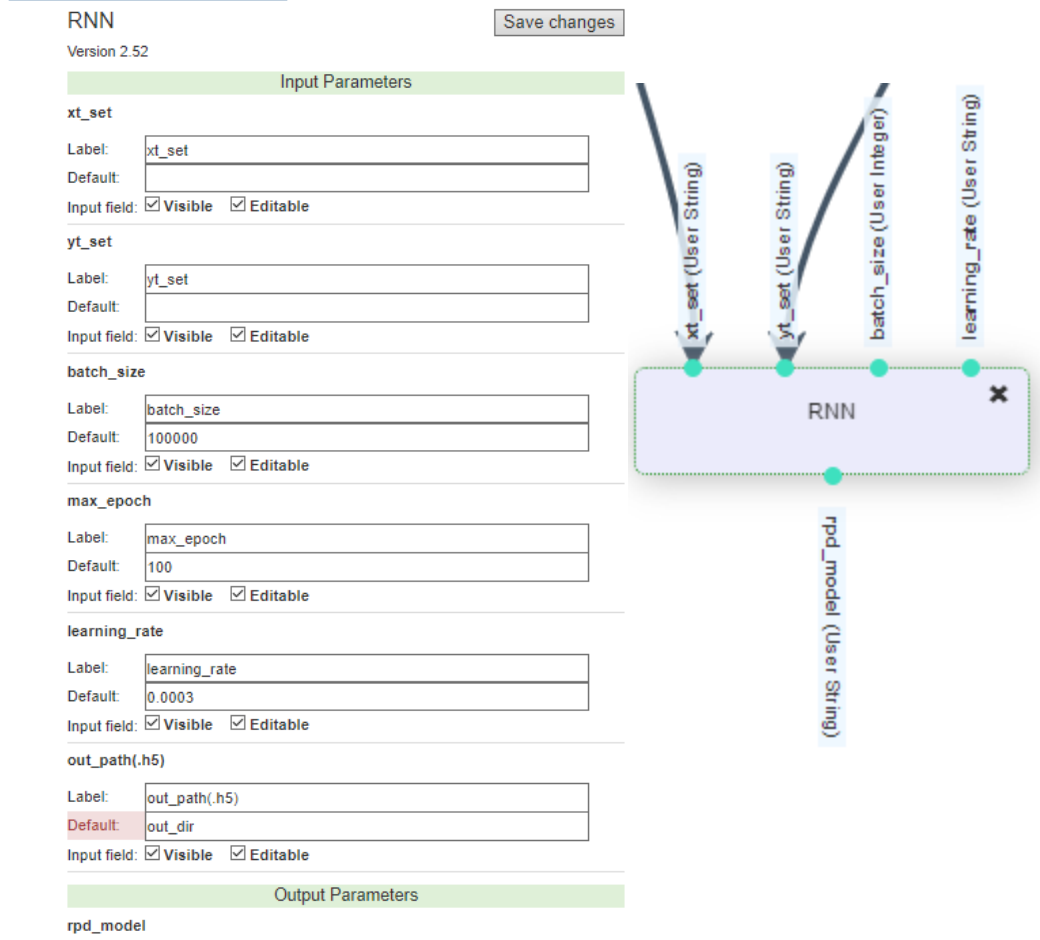
2. Practical Training

2) Method 1 (RNN) – 6. RNNs Model Training_DA

▷ concatenate_set -> RNN

- xt_set: Sets of training data which is provided by "concatenate_set" taking x data
- yt_set: Sets of validation data which is provided by "concatenate_set" taking y data
- max_epoch: Maximum number of training epoch. Training will stop if loss does not decrease (hyper-parameter, default: 100)
- learning_rate: Scale of parameter update per training (hyper-parameter, default, 0.0003)
- out_path: A path where trained RNN model to be saved ("out_dir" for saving it as a temporary file)

Parameter



2. Practical Training

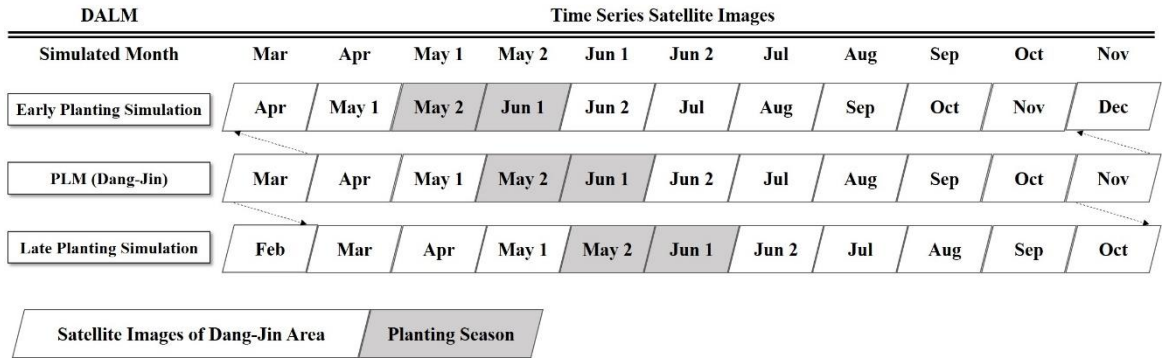
2) Method 1 (RNN) – 6. RNNs Model Training_DA

Explanation

▷ Data Augmentation

- **Large volume of training data can be acquired by applying Data Augmentation, which transforms data to the extent that they do not impair the intrinsic attributes of the classification target. By using large volume of training data, which consists of original data and transformed data, general applicability of models can be enhanced (avoid over-fitting).**
- **In case of rice paddy detection in South Korea, the Data Augmentation was applied by shifting time series because one major key feature for rice paddy detection is planting season, which varies along the regions.**
- **The optimal rice planting season in South Korea varies from May 7 to June 21, and that of Dangjin is from late May to early June. Therefore, through moving the time series of Dangjin back and forth, the phenology of other regions can be simulated, which will increase applicability of the model in the end.**

Reference: Jo, H.W., Lee, S., Park, E., Lim, C.H., Song, C., Lee, H., Ko, Y., Cha, S., Yoon, H., & Lee, W.K. (2020). Deep Learning Applications on Multi-Temporal SAR (Sentinel-1) Image Classification Using Confined Labeled Data: The Case of Detecting Rice Paddy in South Korea. *IEEE Transactions on Geoscience and Remote Sensing*, ?(?), ?-?.



Process Version

- Search Korea S1 Metadata: 1
- Sentinel 1 Preprocess: 1
- Monthly_mosaic: 1
- Time_series_list: 1.0
- import_x_y: 2.2
- Separate_Tr_Va: 1.0
- concatenate_set: 2.2
- RNN: 2.52
- rp_detection: 2

Reference

- 2nd version of the EOPEN platform includes

(a) the current operating decision- making model

(b) the EOPEN ontology

(c) the 2nd iteration of the self-assessment plan

(d) report on EOPEN's clustering techniques

(e) meteorological and climatological data

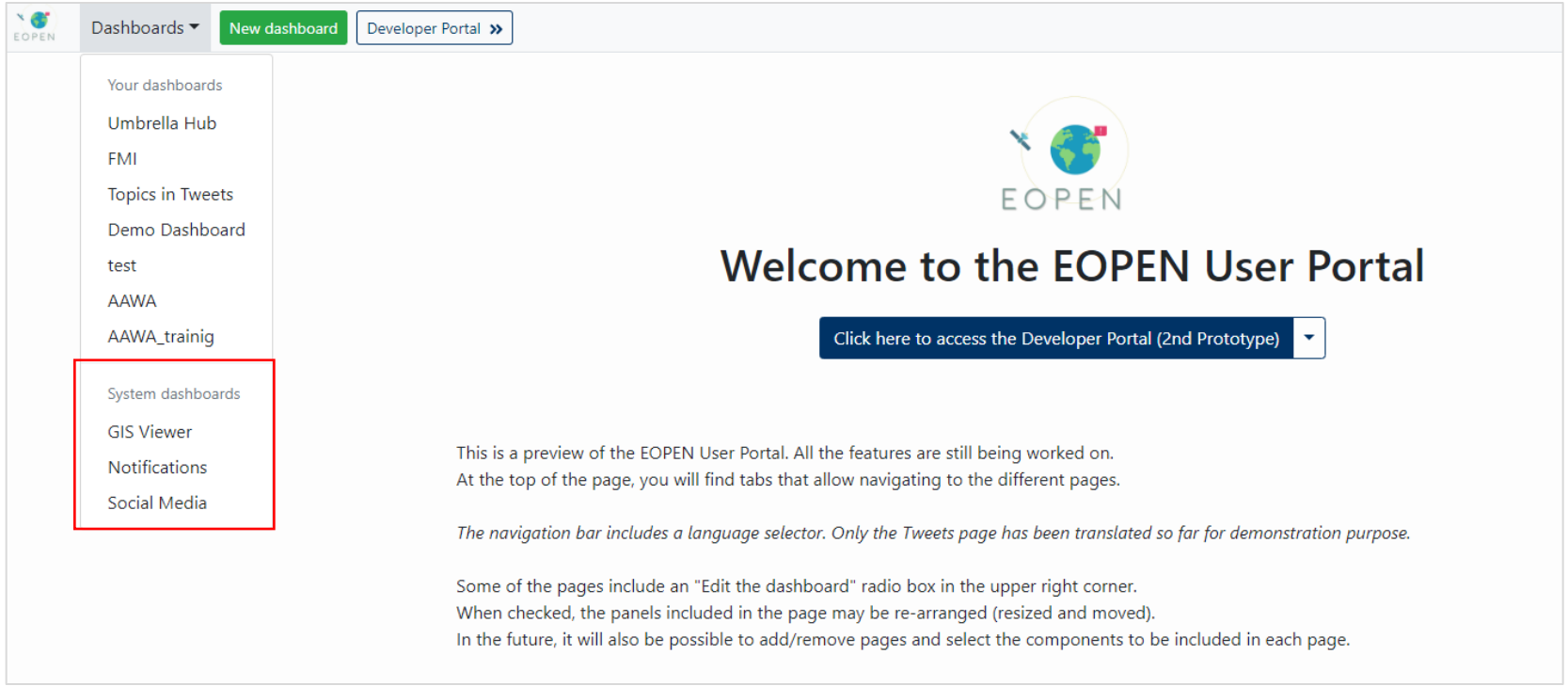
(f) EOPEN's business model and exploitation plans and

(g) the evaluation report of the 1st prototype

- A new instance of the EOPEN Platform (2nd prototype) has been deployed at https://proto2.eopen.spaceapplications.com

3. Visualization of Data

▷ Click Dashboards to enter into GIS Viewer, Notification, and Social Media

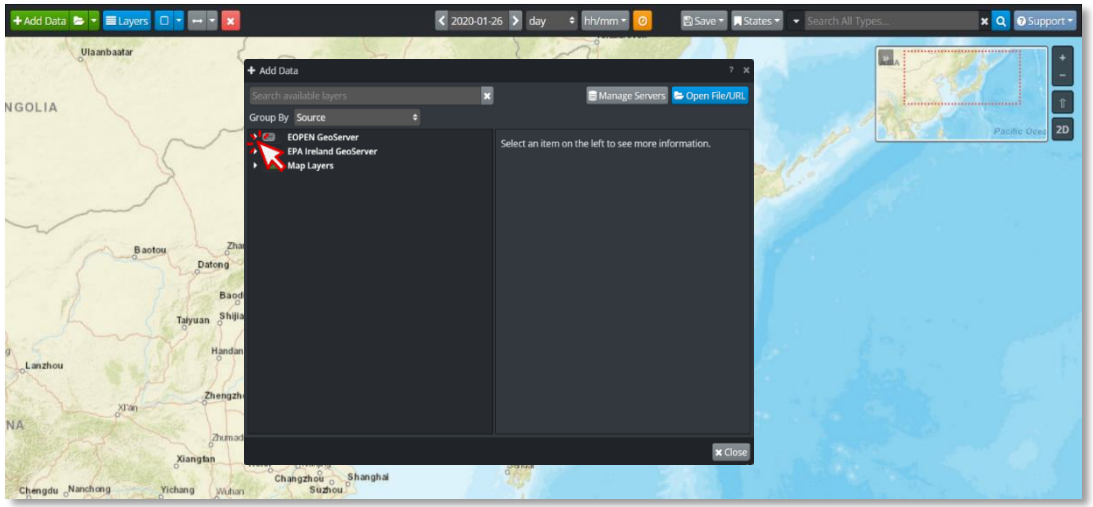


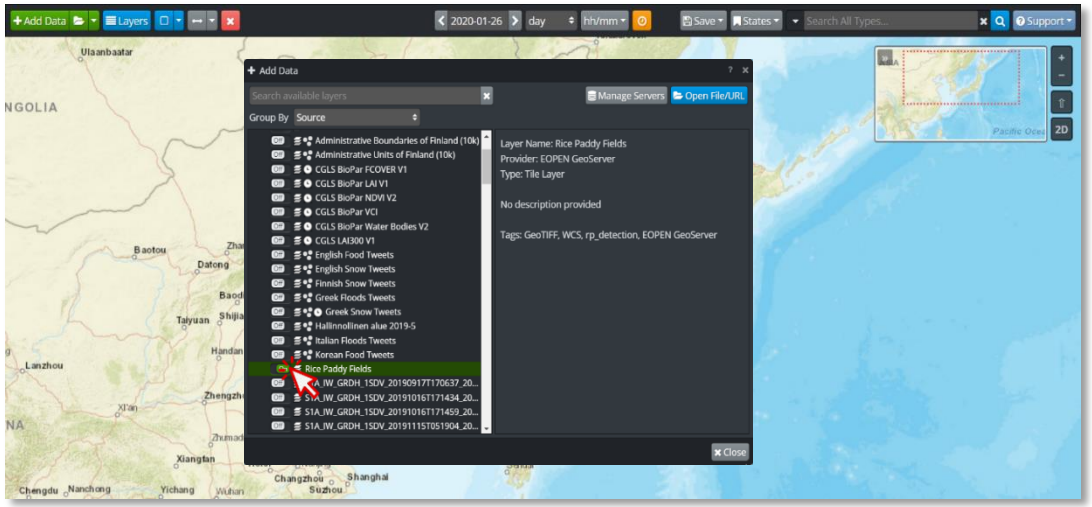
3. Visualization of Data

▷ GIS Viewer

- Click Add data button → Click EOPEN GeoServer(You can group available layers by tag, type, source etc.) → Click “Rice paddy Field tiles”

*Available Data : Water body, Rice paddy, Collected tweets, Weather etc.



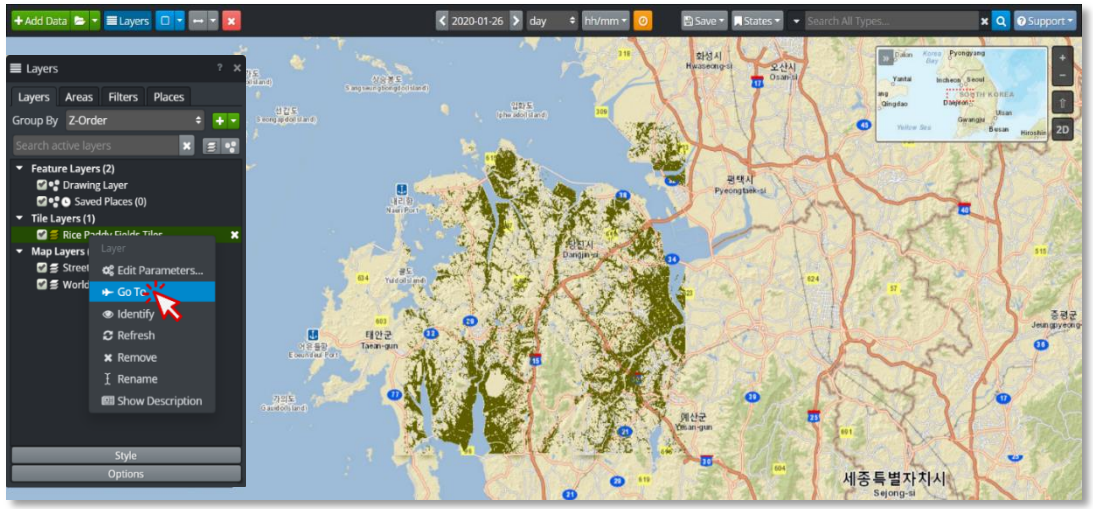


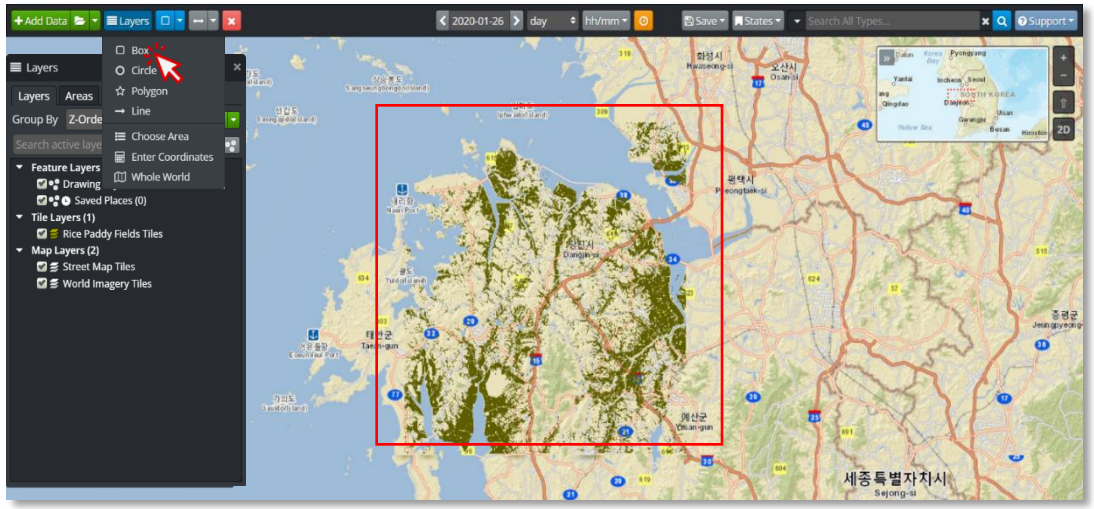
3. Visualization of Data

▷ GIS Viewer

- Click Go To (You can find your result.) → Select the appropriate layer

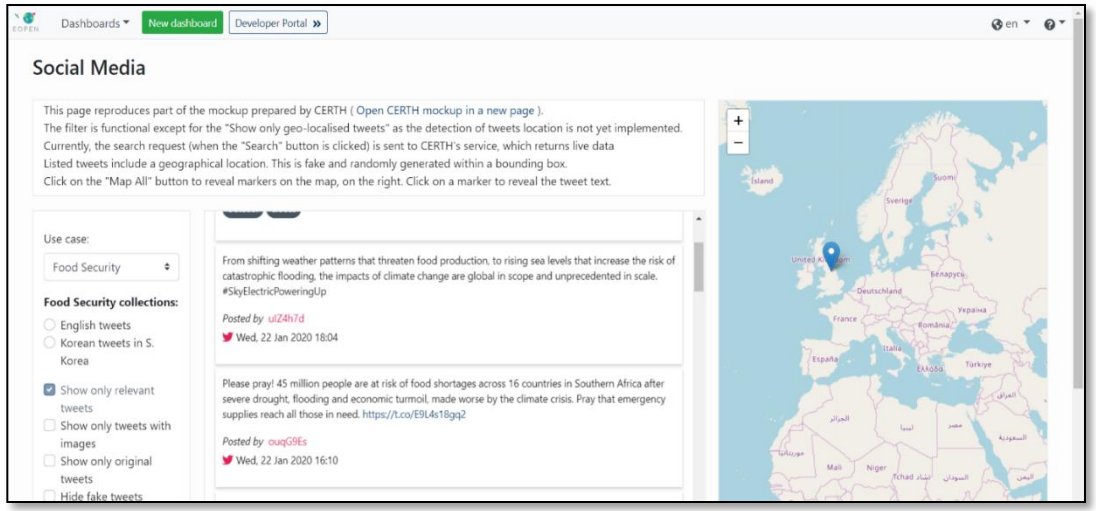
* You can change the style for your result; opacity, brightness, contrast, color etc.

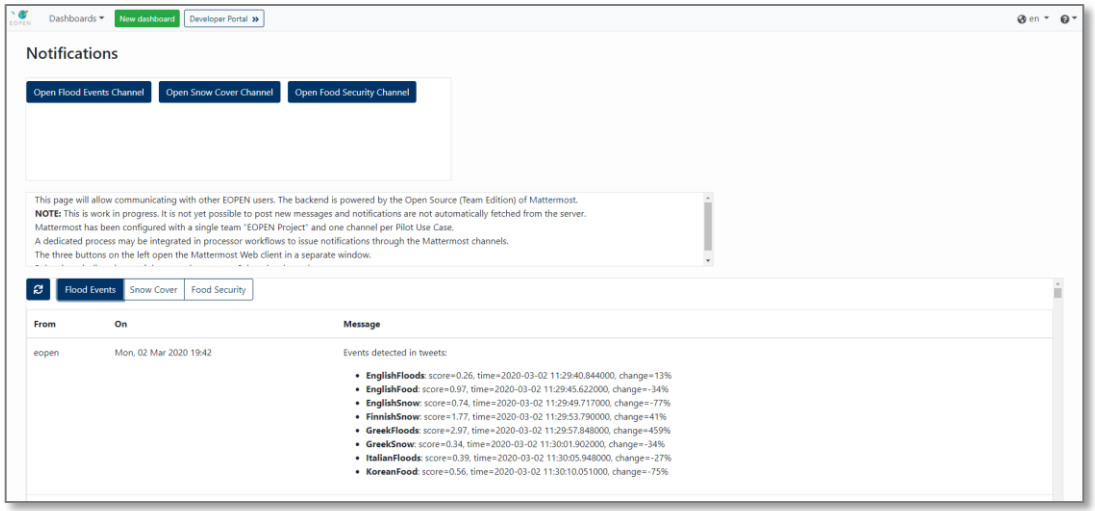




3. Visualization of Data

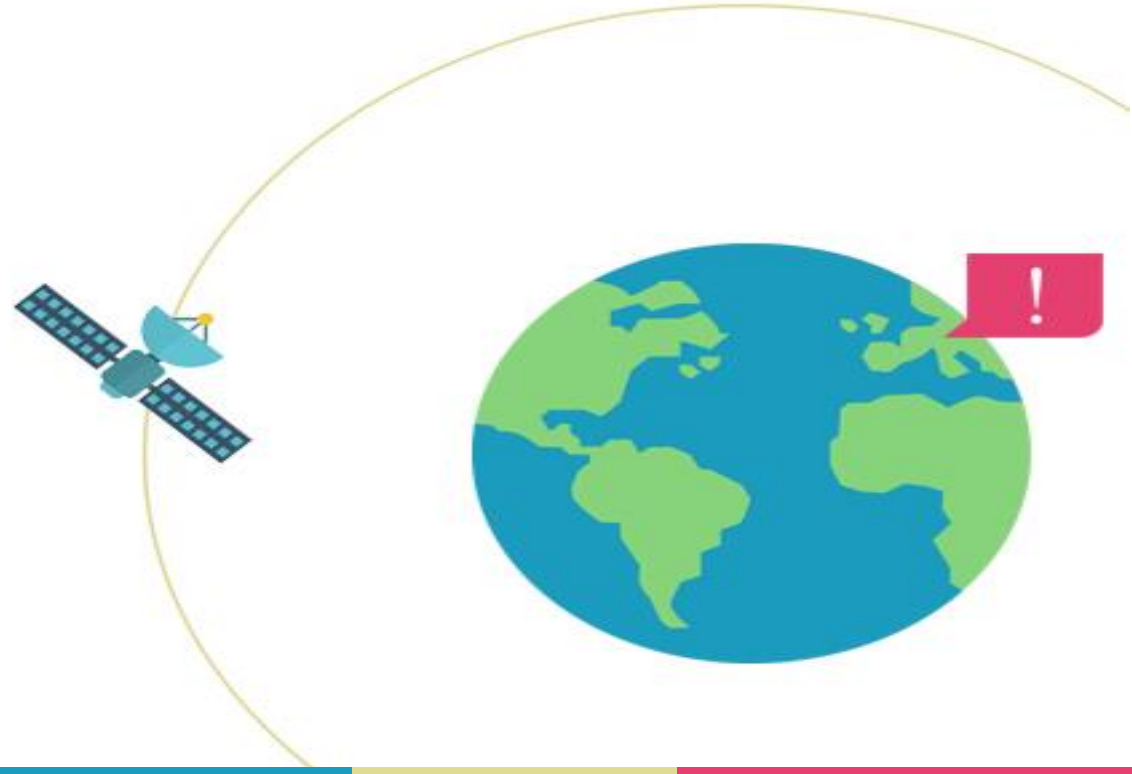
- ▷ Social Media and Notification
- Select the relevant issues and options (You can search the available tweets.)
- the notifications issued by the applications running in the EOPEN Platform





Thanks!

Any questions?



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 776019