Center for Spatial Information Science and Systems
George Mason University

CropScape for Nepal Technical Document

Date Revised: 1 May 2023
Revised by: Gil Heo

Table of Contents

Contents

1. Introduction

The “CropScape for Nepal” is a web service-based and rich internet application that handles large volumes of data, intensive computation, concurrent access and spatial–temporal intensity. The portal supports online visualization, geospatial navigation and querying, reformatting and transformation, delineation of area of interest, on-the-fly data analysis, data processing, and dissemination, into an NDVI and VCI application for Nepal area based on distributed geospatial services and components. The portal uses intuitive building of the end user’s skills and experiences to design a user interface that is effective for both beginners and advanced users. The scalability, reusability, interoperability, performance, and efficiency are balanced in the application of a Service-Oriented Architecture (SOA), and in the implementation of Web services.

2. Design

2.1 Overview

An overview of the CropScape foe Nepal system design is shown Figure 1. It is a composition of multiple sub-module components and the connections among them, which are introduced in detail below.

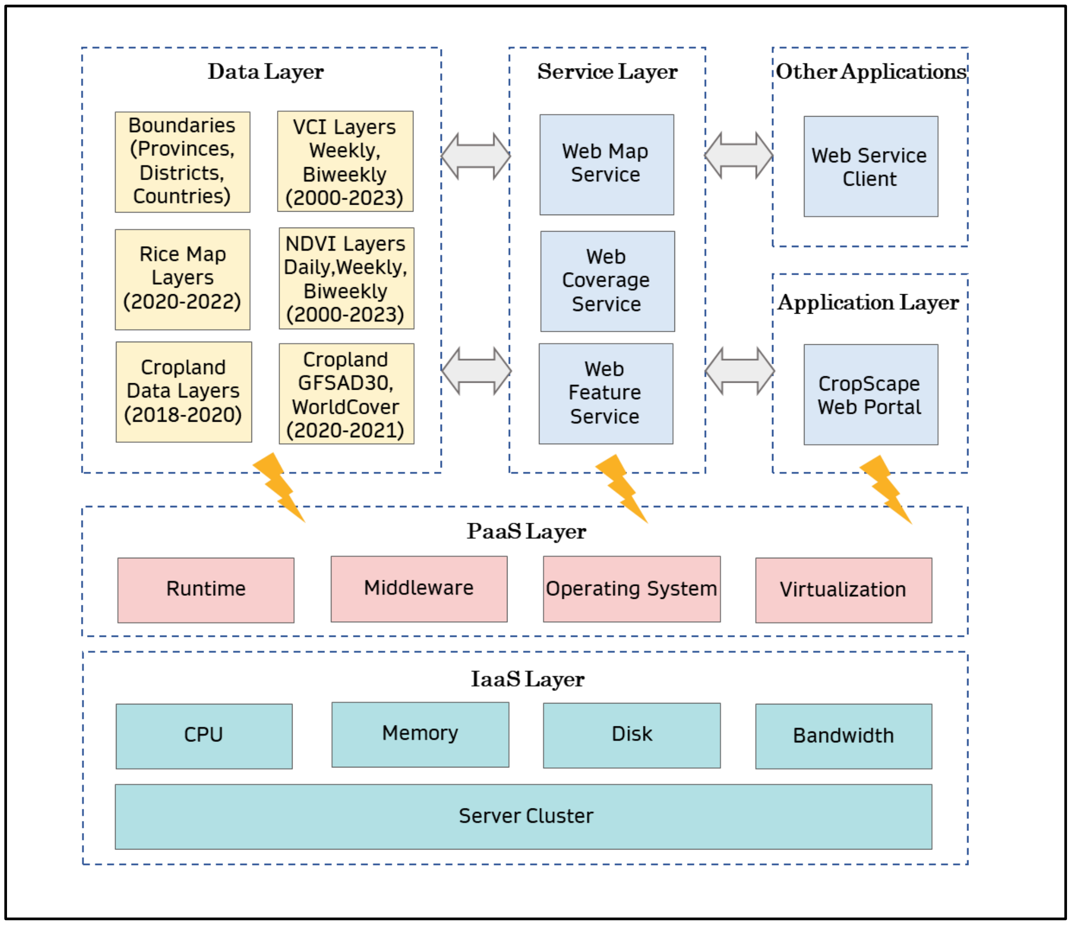

Figure 1. High-level CropScape for Nepal in cloud environment

The original system architecture of CropScape was described in papers titled "*CropScape: A Web service based application for exploring and disseminating US conterminous geospatial cropland data products for decision support*" and "*Enhancing Agricultural Geospatial Data Dissemination and Applications Using Geospatial Web Services*", which contains application layer, service layer, and data layer. The application layer refers to GIS applications and clients that support OGC Web services. The service layer includes standard geospatial Web services such as Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), and Web Processing Service (WPS). All raster data and vector data are stored in the data layer. The Infrastructure-as-a-Service (IaaS) layer is the foundational layer of the cloud environment. It consists of the physical cloud infrastructures, which is built with the server cluster, and offers the computing resources (e.g., CPU, memory, disk, bandwidth). The Platform-as-a-Service (PaaS) layer mainly manages virtualization, operating system, middleware, and runtime. Figure 1 shows the high-level architecture of the framework.

The CropScape for Nepal is based on the original system architecture but implemented as a web application which can run on a server or any platforms including virtual machines.

2.2 Model-View-Controller (MVC) design

The CropScape for Nepal is a web application, and the architecture is based on Model-View-Controller (MVC) software design pattern. *Model* manages data and business logic, *View* handles layout and display, and *Controller* routes commands to the model and view parts.

2.2.1 Model

In the CropScape for Nepal, *Model* has backend processing logic as following:

- Generating cropped NDVI or VCI data and image file with its mapfile
- Creating boundary for Area of Interest (AOI) defined by end-user
- Downloading generated GeoTIFF image file
- Acquiring pixel value in NDVI or VCI image
- Building statistical information
- Generating GML and KML file

The *Model* modules are written in Java codes called by Servlet directly or invoking external programs by creating a process using system call. The external programs consist of gdal tools, shell scripts, C/C++ programs, executable JAR, etc.

2.2.2 View

In the system, *View* is running on the web engine in a web browser. After initial loading on a web browser from web application server, the view side is working independently, and communicates with the server via Asynchronous JavaScript and XML (AJAX) technology.

The view side is focused on the following functions:

- Managing map view area with base map
- Showing toolbars with invoking related functions
- Selecting NDVI or VCI product with periodic and masking options

- Showing rice map layers
- Showing area boundaries by provinces, districts, and counties
- Managing NDVI and VCI product layers added by end-user

The *View* is written in HTML and JavaScript codes based on the Ext JS application framework. All GUI controls, such as list box, text field, tree, toolbars, popup windows, charts, etc. are designed to use Ext JS components. All map layer manipulation, drawing polygon, working as a WMS client, etc. are designed to use OpenLayers library.

2.2.3 Controller

In the system, *Controller* consists of a combination of event handlers invoked by user interactions from web pages and their corresponding Web APIs implemented as servlets fetched by the event handlers.

The controller side is focused on the following topics:

- Requesting NDVI or VCI data and image with AOI boundary
- Requesting pixel information
- Requesting statistical information of the focused NDVI or VCI image
- Acquiring AOI boundary WFS
- Displaying map layer images from WMS
- Requesting static resource content in a preset directory

The *Controller* of user interaction part is event handlers written in JavaScript. Each event hander invokes corresponding Web APIs in the server-side by using Ajax technology which creates asynchronous web applications.

The other part which is in server-side consists of Web APIs which are written in Java. Each API invokes proper logic module defined in the *Model* side.

2.3 Rice Map Layers

The system has a pre-generated rice map images and cropland map images.

Layer Name	**Description**
Rice Map 2022 (Prediction)	Rice Map prediction image for year 2022
Rice Map 2021	Rice Map image for year 2021
Rice Map 2020	Rice Map image for year 2020
CDL 2020	Cropland image for year 2020 from Cropland Data Layer

CDL 2019	Cropland image for year 2019 from Cropland Data Layer
CDL 2018	Cropland image for year 2018 from Cropland Data Layer
Cropland (GFSAD30)	Cropland image from USGS Global Food Security-Support Analysis Data at 30 m (GFSAD30)
Cropland (WorldCover 2021)	Cropland image for year 2021 from ESA WorldCover at 10 m
Cropland (WorldCover 2020)	Cropland image for year 2020 from ESA WorldCover at 10 m

Each image is stored in GeoTIFF format, and it is accessible via local WMS as a layer.

2.4 Product builder

The system has daily, weekly, and biweekly NDVI and VCI products from year 2000 to current (year 2023). The products are generated automatically by executing scheduled jobs which are daily job for daily NDVI, and weekly job for weekly and biweekly NDVI and VCI from MODIS data supported by EarthData, NASA.

The MODIS data are in the following URL:

```
https://e4ftl01.cr.usgs.gov/MOLT/MOD09GQ.006/    (Version 6, discontinued on Feb 17, 2023)
https://e4ftl01.cr.usgs.gov/MOLT/MOD09GQ.061/    (Version 6.1)
```

2.4.1 Daily NDVI

Each daily NDIV product is generated by a daily builder tool once a day.

The first step of the generating is downloading corresponding MODIS data files from the EarthData website.

The following MODIS data file are downloaded:

Country	MODIS data
Nepal	• MOD09GQ.A*yyyydoi*.h24v05.061.*xxxxxxxxxxxxx*.hdf • MOD09GQ.A*yyyydoi*.h24v06.061.xxxxxxxxxxxxx.hdf • MOD09GQ.A*yyyydoi*.h25v05.061.*xxxxxxxxxxxxx*.hdf • MOD09GQ.A*yyyydoi*.h25v06.061.*xxxxxxxxxxxxx*.hdf
Pakistan	• MOD09GQ.A*yyyydoi*.h23v05.061.*xxxxxxxxxxxxx*.hdf • MOD09GQ.A*yyyydoi*.h23v06.061.xxxxxxxxxxxxx.hdf • MOD09GQ.Ayyyydoi.h24v05.061.xxxxxxxxxxxxx.hdf • MOD09GQ.Ayyyydoi.h24v06.061.xxxxxxxxxxxxx.hdf
Bangladesh	• MOD09GQ.A*yyyydoi*.h25v06.061.*xxxxxxxxxxxxx*.hdf • MOD09GQ.A*yyyydoi*.h26v06.061.xxxxxxxxxxxxx.hdf

After the downloading is finished successfully, the tool generates a single NDVI data file in HDF format from the tiled MODIS data files. The last step of the tool is that it builds a GeoTIFF image file with its mapfile from the NDVI HDF file. Now the system is ready to serve as the image accessible via WMS.

2.4.2 Weekly/Biweekly NDVI

Each weekly and biweekly NDIV product is generated by weekly builder tools once a week. The tool calculates a maximum NDVI value of each pixel of a given period (weekly for 7 days or biweekly for 14 days interval, from the first day of year), and stores them as a single HDF format file. After creating a weekly or biweekly NDVI HDF file, the tool creates a GeoTIFF file for serving the image as a WMS layer.

2.4.3 Weekly/Biweekly VCI

Each weekly and biweekly VCI product is generated by weekly builder tools once a week. The tool calculates VCI value of each pixel from given formula:

The MODIS data are in the following URL:

$$\boxed{\mathrm{VCI} = 100 * (\mathrm{NDVI} - \mathrm{NDVI\ min}) / (\mathrm{NDVI\ max} - \mathrm{NDVI\ min})}$$

Before calculating VCI, Min/Max NDVI data should be pre-calculated. In the CropScape for Nepal system, weekly and biweekly Min/Max NDVI data files from year 2000 to the last year are prepared for rapid VCI calculation. Like other products, the tool creates weekly and biweekly GeoTIFF files for serving the images as WMS layers.

The Min/Max NDVI preparation job is a yearly scheduled job. When the end of every year, Min/Max NDVI will be calculated for preparing the next year. The job is scheduled for an early day every year.

2.4.4 Mask layer

All products can be added as a WMS layer with a mask which is one of rice map in the portal. Each creating a mapfile of its product for WMS, the tool also creates additional mapfile which includes all masking layer information. Layer masking functions have been supported by the Map Server since version 6.2 in 2011. Internally, end-user can add NDVI or VCI layer with mask option by using masked layer name, instead of the original layer name.

3. Implementation

3.1 Top-level Software System Architecture

The CropScape for Nepal is a web application executable on an Apache Tomcat web application server. The following Figure 2 shows the software system architecture of the system.

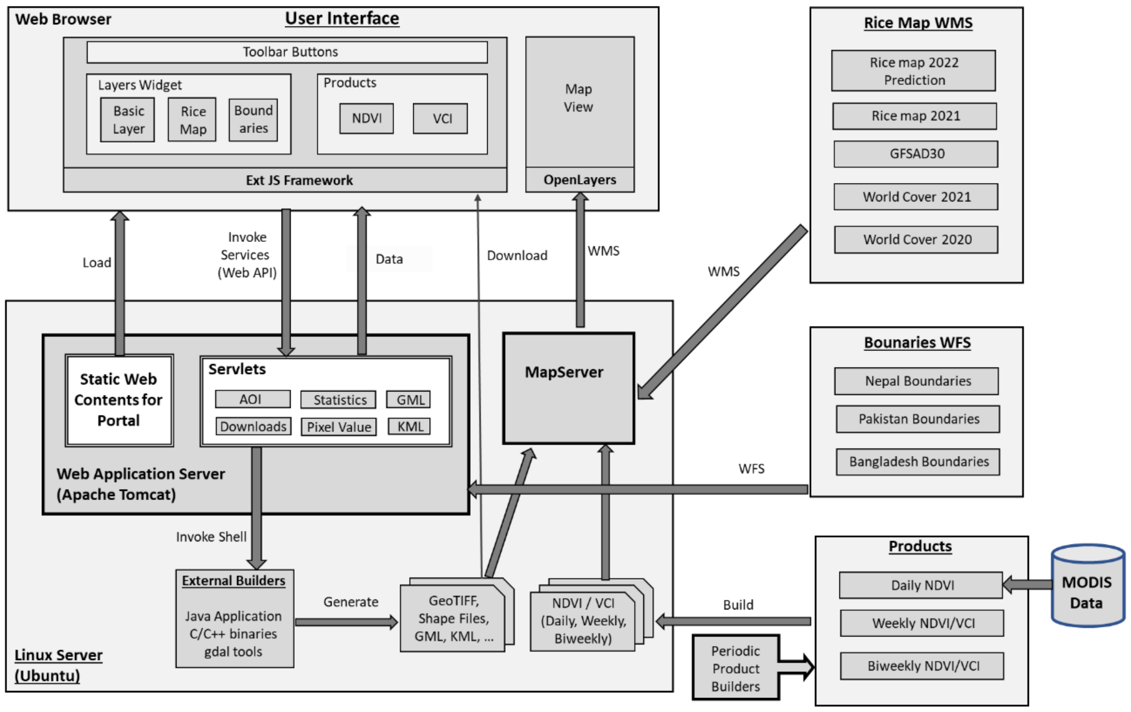

Figure 2. Software Architecture of the CropScape for Nepal System

The system consists of two main parts: server-side components which run on a web application server, and client-side components which interact with end-user. The system has also one of another important part, product builders, which is preparing NDVI and VCI products.

3.2 Client-side Components

Client-side is implemented as HTML and JavaScript codes for running on a web browser. It is loaded by a web browser at initial time, and then runs independently of the server.

Communication between client-side and server-side is established asynchronously by using Ajax. Client-side request services to server-side by calling Web APIs which are implemented by servlets.

3.2.1 Ext JS framework

The client-side initializes its base framework, Ext JS, by calling a method, *Ext.onReady()*. At the initial time the following modules are prepared:

- Initializing OpenLayers, including essential layers, and map control handlers
- Loading initial configurations and data from the server
- Setting toolbars, panels, including their event handlers
- Assigning initial values (product, period, date, etc.), of each component in the panels

The Ext JS framework involves layout of all visible components, including buttons, trees, text areas, selectors, etc., and their event handling. All codes in the client-side satisfies Ext JS coding requirements.

3.2.2 Layers Panel

At the initial time, the WMS configuration of the rice map layers is loaded from the resource directory of the server. The configuration is a JSON format, and the following is an example of the configuration of "Rice Map 2022 (Prediction)":

```
{
  "Rice Map 2022 (Prediction)": {
    "url": "/cgi-bin/wms_ricemap_nepal",
    "layers": "ricemap_2022_prediction_rice",
    "data": "/CROPSCAPE/NEPAL/data/other/ricemap/epsg32645/ricemap_2022_prediction_clipped.tif",
    "legend": "/CROPSCAPE/NEPAL/legend/Legend_ricemap_rice.map",
    "visibility": true
  },
   ...
}
```

Each top-level name key is a layer name which will show at a rice map tree node in the Rice Map tree in the *Layers* panel. The key and value pair of each property means:

Name	Value
url	The URL of WMS service endpoint
layers	Layer name in the GetCapabilities information of the WMS
data	Actual the image GeoTIFF file location in the server. It is required for generating mask layer information in each mapfile for the NDVI and VCI images
legend	A mapfile location which describes legend value range
visibility	Default visibility setting of the layer on the map view area

The rice map layers in a map object instance created by OpenLayers are initialized with the configured values.

3.2.3 Products Panel

Products panel supports product type selector with its period, year, and date. The selected values by end-user are used to build a desired WMS query parameter string as follows:

Mapfile Path:

```
/CROPSCAPE/NEPAL/data/products/{period}/{type}/{year}/{type}_{date}_rice.map

ex) Type: VCI, Period: Weekly, Year: 2023, Date: 16_2023.04.16_2023.04.22
⇨ /CROPSCAPE/NEPAL/data/products/weekly/vci/2023/vci_16_2023.04.16_2023.04.22_rice.map
```

Layer name:

```
{period}_{type}_{date}

ex) Type: VCI, Period: Weekly, Year: 2023, Date: 16_2023.04.16_2023.04.22
⇨ Weekly_VCI_16_2023.04.16_2023.04.22
```

If an end-user added optional mask layer option with the mask name "Rice Map 2022 (Prediction)" then, the layer name will be changed with appending the mask layer name, "ricemap_2022_prediction_rice". Note, the mapfile of the query string is same as non-mask layer query.

Layer name:

```
{period}_{type}_{date}_{mask}

ex) Type: VCI, Period: Weekly, Year: 2023, Date: 16_2023.04.16_2023.04.22
⇨ Weekly_VCI_16_2023.04.16_2023.04.22_ricemap_2022_prediction_rice
```

If an end-user clicks "add" button, the add button handler in the client-side will create a WMS layer instance of OpenLayers and add the layer instance into the product layer tree as an Ext JS tree element.

3.2.4 Toolbar

Most communication between client-side and server-side is invoked by button event handlers in the toolbar. Some buttons invoke corresponding Web API in the server-side.

The following are the relationship between toolbar buttons and their corresponding Web APIs:

Toolbar Button	Web API (servlet) in server-side
Identify Pixel Value	(GET) /GetVCIPixelValue
Define Area of Interest	(GET) /GetGML

Define an Area of Interest	(POST) /ImportAOI
Import/Export a Vector File	(POST) /ExportAOI (GET) /DownloadTIFFFile
Download	(POST) /DownloadData
Statistics	(POST) /CheckDataAvailability (POST) /GetStatistics
Generate Profile	(POST) /GenerateVCIProfile (POST) /GenerateVCIProfileforPolygon

3.3 Server-side Components

The CropScape for Nepal is a web application, and it runs on a web application server. The web application also invokes external tools. The generated temporal resources in PNG, KML, GML, or GeoTIFF by web application or external tools are served to client-side by fetching directly via HTTP GET request, or by displaying as a layer on base map area via WMS.

Deployed server-side components consist of the following parts:

- CropScape for Nepal web application
- Apache Tomcat Web Application Server for running CropScape for Nepal web application
- External tools for generating temporal resources
- Map Server for serving WMS and WFS
- Rice map images
- Cropland map images (CDL, NASA GFSAD30, ESA WorldCover 2020 and 2021)
- Weekly and biweekly VCI in format HDF and GeoTIFF with its mapfiles
- Daily, weekly, and biweekly NDVI in format HDF and GeoTIFF with its mapfiles

3.3.1 CropScape for Nepal Web Application

The CropScape for Nepal web application serves initial static web resources for a client-side which are running on a web browser and supports Web APIs which are requested by the client-side.

The following are Web APIs of the server-side:

Web API	**Description**
/DownloadData	Generate an AOI cropped GeoTIFF image from given parameter’s HDF file

/GetImage	Generate a PNG and KML files from given GeoTIFF image
/GetGML	Generate boundary GML from defined AOI by province/district/county
/ExportAOI	Generate GML or zipped ESRI shapefile from end-user's drawing polygon
/DownloadTIFFFile	Download a resource file from given path
/ImportAOI	Upload GML or zipped ESRI shapefile
/RetrieveGML	Convert a feature collection GML from uploaded zipped ESRI shapefile
/GetVCIPixelValue	Acquire a pixel value from generated AOI cropped GeoTIFF file
/proxy (RequestProxy)	Accessing cross-origin resource for avoiding HTTP cross-origin restriction
/CheckDataAvailability	Check existence of the NDVI or VCI GeoTIFF image for the given date defined by end-user
/GetStatistics	Create periodical statistic information from end-user defined AOI boundary and given layer image
/GenerateVCIProfile	Generate time series profile (begin-end) of NDVI for given pixel or lat/long
/GenerateVCIProfile forPolygon	Generate time series profile (begin-end) of NDVI for given polygon
/GetResource	Fetch a resource file located in resource directory of the server

3.3.2 External Tools

Some of server-side generation are done by invoking external tools from web application. The following are the tools invoked by server-side:

The following are Web APIs of the server-side:

External Tool	Description
gdal tools	ogrinfo, ogr2ogr, gdalwarp, gdal_rasterizer, gdal_translate, gdaldem, gdallocationinfo
bin utils	tar, unzip
NDVIChartGeneartor.jar	Chart image generator for making time series profile (location: ./javaprogs/)
gdalstat.x	Acquire statistical information from given GeoTIFF file (location: ./programs/cpp/gdalstat/release/)

3.4 Product Builder

NDVI and VCI products in the CropScape for Nepal are generated by daily and weekly scheduled jobs.

3.4.1 Daily NDVI

Daily NDVI is generated once in a day. The following are main programs for each generating step:

(location: ./bin/updatedailyndvi_dynamic/)

Program	Description
processall.sh	Top-level batch program for periodic daily NDVI
processdaily.sh	Top-level batch program for each single daily NDVI
updatedaily_workflow _modis.sh	Download MODIS HDF files from NASA data website and then generate a single daily NDVI HDF file, GeoTIFF file, and mapfiles
hdfndvi_dynamic.sh	Generate a daily NDVI HDF file from MODIS HDF files (external tool: ./programs/ndviall.x)
togeotiffutm45n_dynamic.sh	Generate a GeoTIFF file from the NDVI HDF file
createmapfile	Generate two mapfiles: one for the original reference, the other for including mask layers
build_rice_mapfile_v2.py	Generate a mapfile which includes masking layers

3.4.2 Weekly/Biweekly NDVI

Weekly and biweekly NDVIs are generated once a week. The following are main programs for each generating step:

(location: ./bin/updatedweeklyndvi_dynamic/, ./bin/updatedbiweeklyndvi_dynamic/)

Program	Description
processall.sh	Top-level batch program for all possible weekly/biweekly NDVI given periodic years
processweeks.sh	Top-level batch program for all weekly/biweekly NDVI for 1 year period
process1periodndvi.sh	Generate a single weekly/biweekly NDVI HDF file, GeoTIFF file, and mapfiles
periodndvi.sh	Generate a weekly/biweekly NDVI HDF file from daily NDVI HDF files (acquiring max value) (external tool: ./programs/cndvi.x)

togeotiffutm_dynamic.sh	Generate a GeoTIFF file from the weekly/biweekly NDVI HDF file
createmapfile	Generate two mapfiles: one for the original reference, the other for including mask layers
build_rice_mapfile_v2.py	Generate a mapfile which includes masking layers

3.4.3 Weekly/Biweekly VCI

Weekly and biweekly VCIs are generated once a week. The following are main programs for each generating step:

(location: ./bin/updatedweeklyvci_dynamic/, ./bin/updatedbiweeklyvci_dynamic/)

Program	Description
processall.sh	Top-level batch program for all possible weekly/biweekly VCI given periodic years
processweeks.sh	Top-level batch program for all weekly/biweekly VCI for 1 year period
process1periodvci.sh	Generate a single weekly/biweekly VCI HDF file, GeoTIFF file, and mapfiles
periodvci.sh	Generate a weekly/biweekly VCI HDF file from weekly NDVI HDF files and Min/Max weekly NDVI HDF files (external tool: ./programs/vci_improved.x)
togeotiffutm_dynamic.sh	Generate a GeoTIFF file from the weekly/biweekly VCI HDF file
createmapfile	Generate two mapfiles: one for the original reference, the other for including mask layers
build_rice_mapfile_v2.py	Generate a mapfile which includes masking layers

3.4.4 Yearly Min/Max NDVI

Each generation of a weekly or biweekly VCI is required Min/Max NDVI. For maximize generation efficiency, all Min/Max NDVIs from 2000 to the last year are precalculated. This calculation job is executed once a year. The following are main programs for the step:

(location: ./bin/updatedyearlyminmaxweekly_dynamic/, ./bin/updatedyearlyminmaxbiweekly_dynamic/)

Program	Description
processall.sh	Top-level batch program for generating weekly/biweekly Min/Max NDVI for periodic years

mmcalc.sh	Top-level batch program for all weekly/biweekly NDVI for all weekly/biweekly periods of single year
calc1periodicminmax.sh	Generate a weekly/biweekly Min/Max NDVI HDF file from year 2000 to given year for a single weekly/biweekly period (external tool: ./programs/vciperiodminmax.x)

4. Revisions

Date	Author	Description
1 May 2023	Gil Heo	Initial Document