**Automation of In-Season Crop Acreage Forecasting using SAR Sentinel-1A and EOS-4 Data**

Radhika Aggarwal1, Akash Goyal2

Department of CSE, The NorthCap University, Gurugram, Haryana, India1

Regional Remote Sensing Centre-North, New Delhi, India2

aggarwalradhika2905@gmail.com1, akash\_g@nrsc.gov.in2

**ABSTRACT**

This paper proposes a GUI-based automation system for in-season multi-class crop classification and crop acreage estimation on SAR data using ML algorithms. The system provides options for two satellites - Sentinel-1A and RISAT-1, enabling users to download, preprocess, and classify the data quickly and efficiently. The preprocessing steps include radiometric calibration, speckle filtering, subset, geometric/terrain correction, mosaic, district masking, crop masking, and layer stacking. A backscatter curve is plotted using the ground truth points to validate the accuracy of the preprocessed image. The classification is performed using six ML algorithms, and hyperparameter tuning is done to optimize the model training parameters. After the classification, the output is sieved, and the area of each crop class is calculated using Python CSVs. The proposed automation system generates a single PDF file as a summarizing document, which includes the classification report, confusion matrix, backscatter curve, and classified image. The calculated multi-crop areas can also be visualized using pie charts, bar charts, etc. The system is highly efficient and accurate, available to be used by any layman, and it can provide valuable insights for stakeholders in the agricultural sector. The methodology presented in this research paper can be replicated and extended for different datasets and applications.

**KEYWORDS:** Machine Learning, Automation, Artificial Intelligence, Hyperparameter Tuning, Geoprocessing, GDAL, SNAP, Sentinel-1A, EOS-4, Digital Elevation Module, Remote Sensing

1. **INTRODUCTION**

Crop classification using remote sensing data has been an area of research interest for many years. Traditionally, it was done using conventional approaches like a visual interpretation of satellite imagery, which is a time-consuming and labor-intensive process. However, with the advancements in Machine Learning (ML) algorithms, automated crop classification using Synthetic Aperture Radar (SAR) data has gained popularity in recent times. SAR data has unique features that make it ideal for crop classification, one of them being it is not affected by cloud cover or atmospheric conditions, and has a high spatial resolution that enables the identification of small agricultural plots. ML algorithms, on the other hand, have proven to be highly effective in processing and analyzing large amounts of data quickly, efficiently, and accurately.

In this research paper, we explore the use of technologies such as machine learning algorithms, the geospatial data abstraction library, the SNAP python interface, the ASF-Alaska data acquisition python library, and the PyQT5 technology of Python for the GUI-based automation of multi-class crop classification, and in-season crop acreage forecasting using SAR (Sentinel-1A, EOS-4) data. The data acquisition is either done using asf-alaska API or locally available resources, preprocessing using SNAP python interface, numerical analysis i.e. backscatter curve using data analytical tools of Python, and classification and acreage forecasting using the scikit-learn library of python. The output of the classification is used to calculate the area of each crop class in the ROI which can be helpful for farmers, policymakers, and other stakeholders in the agricultural sector, as it can help in making informed decisions related to crop management and resource allocation. The primary objective of this research is to automate and perform a numerical analysis of the given data.

An additional feature to automate the process of downloading the Shuttle Radar Technology Mission Height (SRTM) module has also been introduced in the proposed system, which extracts the latitudinal and longitudinal credentials of the data file during preprocessing to bring about the geometric terrain correction for Sentinel-1A data.

In conclusion, the use of AI and ML algorithms for crop classification is a promising field of research that has the potential to revolutionize the agricultural sector. The research presented in this paper contributes to this field by demonstrating the effectiveness of the proposed flow of processes in automating the crop classification process and providing valuable insights for stakeholders in the agricultural sector.

1. **MATERIALS AND METHODOLOGY**

**Types of data used**

In the proposed automation tool, SAR data has been utilized to bring about information over crop area for an ROI. Table 1. depicts the specifications of the respective satellites and the data used.

| S. No. | Satellite | Year Launched | Sensor | Organization | Spatial Resolution | Temporal Resolution | Polarization |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | Sentinel-1A | 2014 | SAR-C | ESA | 10m | 12 days | VV, VH |
| 2. | RISAT-1 / EOS-4 | 2012 | SAR-C | ISRO | 1m | 17 days | HH, HV |

Table 1. Tabular description of types of data used: Sentinel-1A, and EOS-4.

**Sentinel-1A**

Sentinel-1A is a Synthetic Aperture Radar (SAR) satellite mission operated by the European Space Agency (ESA) as a part of the Copernicus program. It provides high-resolution SAR data over a wide range of applications including land, ocean, and cryosphere monitoring. The Sentinel-1A mission carries a C-band SAR sensor, which operates in two polarizations: Vertical Vertical (VV) and Vertical Horizontal (VH). The VV polarization is transmitted and received in the vertical direction. This polarization is sensitive to the vertical structure of the target, such as vegetation canopy and roughness of the surface. It is commonly used for land applications such as vegetation mapping, soil moisture estimation, and urban area monitoring. In addition, it is sensitive to oil spills and can be used for marine applications. The VH polarization is transmitted in the vertical direction and received in the horizontal direction. This polarization is sensitive to the orientation of the target, such as man-made objects and surface roughness. It is commonly used for land applications such as detecting man-made objects and monitoring infrastructure. It is also used for sea-ice monitoring and ship detection in marine applications.

Sentinel-1A provides SAR data with a spatial resolution of up to 10 meters and a swath width of up to 400 km. It operates in different modes including Stripmap, Interferometric Wide Swath, and Extra-Wide Swath modes, each with different spatial resolutions and swath widths. The data can be acquired in different polarization modes, including single polarization and dual polarization modes.

**RISAT-1/EOS-4**

RISAT-1 is a C-band Synthetic Aperture Radar (SAR) satellite launched by the Indian Space Research Organisation (ISRO) in 2012. It provides a very high spatial resolution of 1m and the data used in the study has a temporal resolution of over 17 days. The SAR sensor on RISAT-1 has two polarization modes: Horizontal-Horizontal (HH) and Horizontal-Vertical (HV). In HH polarization, the radar transmits a horizontal electromagnetic wave that is polarized parallel to the ground. The radar receives backscattered signals that are also polarized parallel to the ground. HH polarization is sensitive to roughness, volume scattering, and surface features perpendicular to the radar line of sight. This polarization mode is useful for land cover classification, soil moisture estimation, and urban area mapping applications. In HV polarization, the radar transmits a horizontal electromagnetic wave that is polarized perpendicular to the ground. The radar receives backscattered signals that are polarized parallel to the ground. HV polarization is sensitive to double-bounce scattering and can distinguish between different types of scattering surfaces. This polarization mode is useful for vegetation mapping, biomass estimation, and wetland mapping applications.

RISAT-1 data is available in Level-1 Ground Range Detected (GRD) format, which is the basic processing level for SAR data. The data is provided in single look complex (SLC) format, which means that the raw data has been focused to a high resolution using a SAR processing algorithm. The SLC data is then processed to generate the GRD product, which is a geocoded image with radiometric and geometric corrections applied. The RISAT-1/EOS-4 GRD product provides information on the backscattered power of the radar signal, which can be used to extract information about the surface properties of the Earth. The backscattered power is measured in decibels (dB) and is a function of the radar incidence angle, polarization, and surface roughness. Overall, the RISAT-1/EOS-4 SAR data in HH and HV polarizations provide valuable information for a range of applications in Earth observation and remote sensing, especially for studies related to land cover, vegetation, and wetlands.

**Technology Stack:**

To develop the automation tool, a set of technical requirements were considered, and using them only the system has been programmed.(Table 2.)

| **S. No.** | **Technology** | **Use Case** |
| --- | --- | --- |
| 1. | Python Programming | v3.6, v3.11 |
| 2. | GDAL | Data engineering |
| 3. | SNAP Python Interface | Data preprocessing |
| 4. | ASF-Search | Data acquisition |
| 5. | Scikit-learn | Data classification |
| 6. | PyQt5 | GUI |

Table 2. Technology stack for the development of the automation tool.

1. **Python (Python 3.6.0 and Python 3.11.0)**

Python is a popular open-source programming language that provides improved process control features. It can create sophisticated multi-protocol network applications while simultaneously preserving a clear and simple syntax.

Two versions of Python are required for the project because of the compatibility issues with supporting Python interfaces like “snappy” (SNAP-Py interface) and “sklearn” for machine learning classifications. Python 3.6 is used for preprocessing geospatial images. Preprocessing of the images follows the methodology as discussed further in the report. Python 3.11 is used for training the model and predicting the classified images using geospatial data abstraction techniques.

## **SNAP Python Interface**

The primary programming language for the implementation of SNAP (Sentinel Application Platform) is Java, and so is its API a Java API. But somehow, Python has been used here for the SNAP interface because its architecture describes that one can use both SNAP Engine and Desktop with Python too. Using SNAP with Python elevates the overall data preprocessing and accuracy of process outputs, such as terrain correction and speckle filtering. It supports a very limited number of Python versions, i.e., Python 2.7, 3.3, and 3.6. In the proposed automation tool, Python 3.6 has been used to preprocess the satellite data.

For implementation through Python, there should be configurations set up by the user in the SNAP Command-Line. It can be done by following the below steps(Figure 1.):

* Install SNAP using the setup file available online.
* Keep clicking Next, since no customization is needed. Click on the Install button and close after installation.
* Open ‘SNAP Command Line’ from Start.
* Type cd C:\Program Files\snap\bin
* Type snappy-conf C:\Python36\python.exe
* Copy 'C:\Users\\*username\.snap\snap-python\snappy' to 'C:\Python36\Lib\site-packages\\'
* Users can set up a memory limit for SNAP to consume since its major drawback is memory and space management.

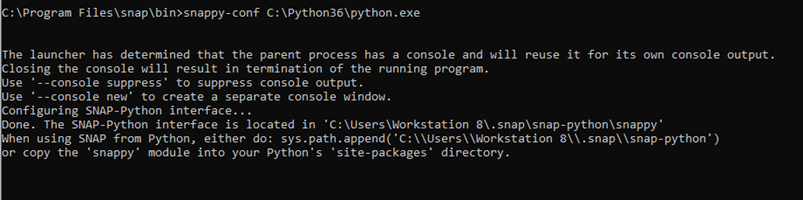


Figure 1. SNAP Command-Line.

## **GDAL (Geospatial Data Abstraction Library)**

## It is a Python library, released in 2000, used to manipulate and study geospatial data in raster and vector formats, e.g., ‘.tif’ for raster, ‘.shp’ for vector. It is a free, open-source library widely provided in the form of wheel files for almost all Python versions. A good number of geospatial/remote-sensing-based software use this library in their backend programs, such as ArcGIS, QGIS, ERDAS, Google Earth, etc. GDAL wheel files are usually available [here](https://www.lfd.uci.edu/~gohlke/pythonlibs/).

## **ASF-ALASKA Python Interface**

ASF-Alaska is a website managed by NASA to download SAR spatial imagery. [It](https://search.asf.alaska.edu/#/?dataset=SENTINEL-1) has provided a Python interface namely “**asf\_search**” which allows users to download required SAR satellite images.

It is simple to use and provides a bridge between NASA’s Alaska Satellite Facility Distribution Active Archive Center (ASF DAAC), which allows the user to download satellite data by specifying parameters such as start and end dates, flight direction, platform, well-known text, processing level, and a relative orbit. To access it through Python, one needs to have a login username and password too for authentication purposes.

## **SCIKIT-LEARN Machine Learning Library**

It is a Python library used for the implementation of machine learning algorithms and techniques on a set of training and testing data. It consists of various regression, classification, and clustering algorithms such as linear regression, logistic regression, SVMs, decision trees, random forest, gradient boosting, k-means, etc to manipulate and apply the property of prediction and estimation on a given set of data.

In the proposed automation software, it is used for working with random forests, decision trees, k-nearest neighbors, support vector machines, naïve bayes, and multi-layer perceptron.

## **Py-QT5 Technology**

It is one of the options available in Python used to develop GUI-based applications to provide an easy-to-interact-with, user-friendly frontend so that the communication between processes and the end goal receiver becomes faster and clearer. It comes with a set of tools, according to the user requirements; one of them being QtDesigner which has an interface very easy to use and design ‘.ui’ files.

**System Configuration:**

Table 3. describes the ideal system configuration for the efficient and smooth functioning of the automation tool.

| **S. No.** | **Requirement Type** | **Configuration** |
| --- | --- | --- |
| 1. | GPU | NVidia |
| 2. | Workstation | High-end |
| 3. | Installed RAM | 16.0 GB or more |
| 4. | System Type | 64-bit operating system, x64-based processor |

Table 3. System configuration for smooth functioning of the automation tool.

**Study Gap**

In earlier development studies for this type of tool, there were a few loopholes that made it difficult for the user to understand the workflow and output mechanism of the system. A few of them are:

* Poor memory management
* Time complexity
* Error in downloading DEM (SRTM 1 Sec HGT)
* Poor frontend and GUI
* Inaccurate area estimation
* Too many dependencies

In this proposed system, all above listed issues were solved, and sorted. Shuttle Radar Technology Mission Height (SRTM 1 Sec HGT) is a digital elevation model (DEM) sampled at one-arc second and intervals in latitude and longitude. In a geographic projection, it is divided into one by one-degree latitude and longitude.

The nomenclature convention of the file names goes by the latitude and longitude of the lower left corner of the tile, which are basically the geometric center of the lower left pixel, and an extension ‘.hgt’ for them being height files. In the terrain correction process, these elevation models are used to remove the misleading influence of topology on backscatter values in SAR images. A function to design nomenclature for naming SRTM data files over an area was created to help the tool auto-download the .hgt files from the internet since there could be proxy restrictions that don’t allow SNAP terrain correction to auto-download SRTM data.

A new Graphical User Interface has been designed for the system for better interaction between the user and itself. The user has multiple features to give in.

Preprocessing for SAR EOS-4 data has also been introduced here which has different temporal and spatial resolutions than Sentinel-1A. It has polarizations such as HH, HV, and HH+HV.

**The flow of processes**

After multiple hits, trials, and tests, a specific approach was devised to bring about the desired outcomes of the data. Figure 2. shows the flow of steps followed to estimate the spread of each crop in the satellite data.

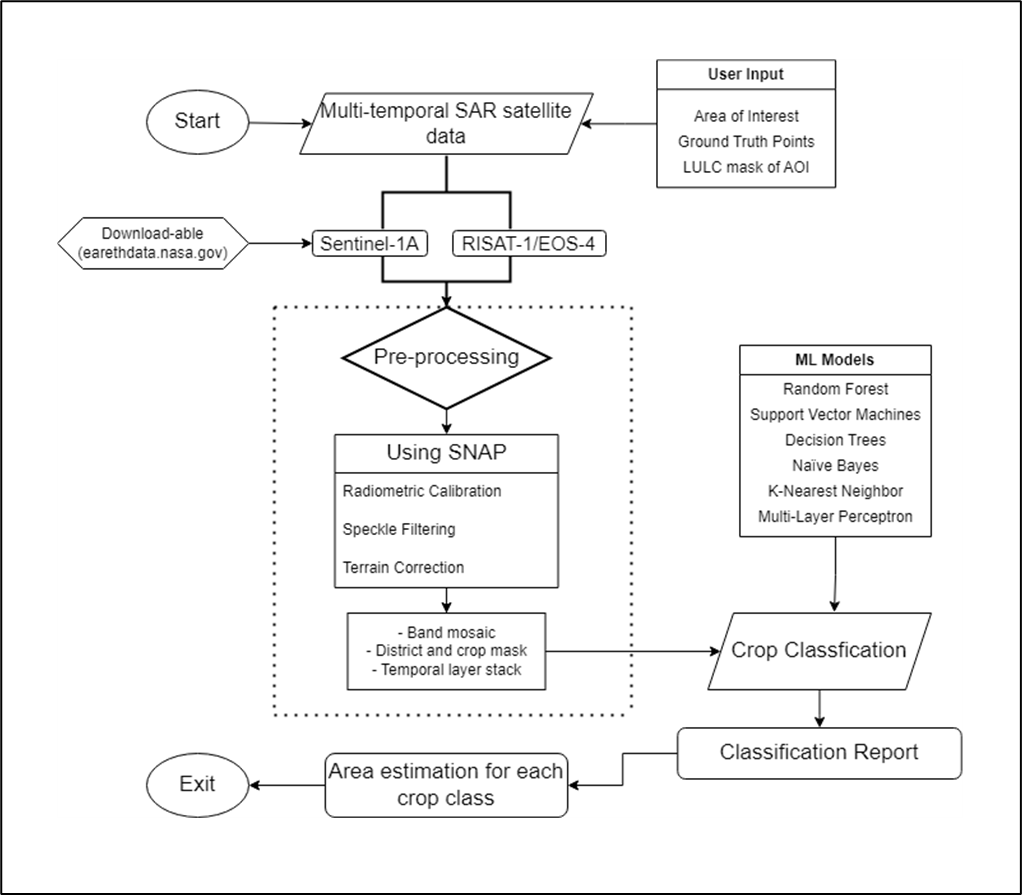


Figure 2. Detailed flowchart of the methodology followed by the system.

The primary task for the data is to get preprocessed and classified, and then the calculation for each classified crop can be done. The automation flow is

1. **Data download using ASF-Search**
2. **Preprocessing using SNAP-Py**

* **Calibration**
* **Speckle Filter**
* **Subset**
* **Geometric Terrain Correction**
* **Mosaic**
* **Layer Stack**
* **LULC Mask**

1. **Classification using scikit-learn**
2. **Area Estimation**

## **Data Download using ASF-Search**

**SENTINEL-1A**

Due to high temporal resolution, the phenological information of the target crops can be easily captured by the satellite through the multi-date analysis. The satellite captures imagery at a high spatial resolution of 10m and a temporal resolution of 12 days. The high temporal and spatial resolution of the Sentinel-1 data makes it perfect for the study of the monitoring of crops. The satellite operates at a central wavelength of 5.04 GHz, at a spatial resolution of 10m, and a swath of 250 Km.

In this study, the data download is automated using a Python module called asf-search which can be used to fetch the orbit number from the date of pass input by the user and calculate the relative orbit number using the formula:

Relative Orbit Number = mod (Absolute Orbit Number orbit - 73, 175) + 1

It is observed that Relative orbit number for the images with same paths (12 days’ time gap) are same. Using the calculated relative orbit number, all the tiles from the input range of dates are downloaded to the setup path.

The user is provided with the choice to either download the data with the input of the area of interest shapefile or browse the data files from the system.

Data download privileges are not available online for EOS-4 data at the moment.

## **Preprocessing using SNAP-Py**

Preprocessing for both SENTINEL-1A and EOS-4 has been approached in different ways. The Sentinel-1A data is available as Single Look Complex (SLC), and Ground Range Detected (GRD). For the current study, the GRD product of Sentinel-1A was employed and further pre-processed. The GRD data products have already been focused, multi-looked, and projected in the ground range. Further preprocessing of the data was carried out to obtain the final backscatter image and values. The data were pre-processed using the following steps:

### **Speckle Filter using Lee Filter**

The speckle in the SAR imagery is the granular noise, which is mainly because of the interference of waves reflected from many elementary scatters (Lee et al., 1994). Speckle filtering is the process to reduce the granular noise present in the imagery to increase image quality. This step is performed before radiometric calibration and terrain correction, so the speckle present in the image does not get propagated further. In the current study, the Lee filter (Lee, 1980) was used to perform speckle filtering using a 3X3 window size. The large window size of 5X5 and 7X7 was avoided as at these window sizes small information is lost which can further affect the accuracy of classification.

### **Radiometric Calibration**

### The radiometric calibration process is done to convert the digital pixel values to radiometrically calibrated backscatter values.

To calibrate EOS-4 data, we don’t use SNAP features, instead, use a formula to transform the pixel values:

Where

DN = backscatter value in the raw satellite image

i = incidence angle

k = calibration constant (HH or HV)

DN’ = modified value of the pixel

Unit conversion of pixel values to decibels is also done here in both types of data using the formula:

Where

DN’’ = pixel value in decibel

### **Range Doppler Terrain Correction using SRTM 1 Sec (30m) HGT DEM Data**

The SAR data are majorly captured at a slanting angle, i.e. greater than 0°, therefore, there can be few distortions corresponding to side-looking angle. Terrain correction is therefore carried out to eliminate these distortions and match the imagery to the actual effective world. These distortions are majorly caused by foreshortening and shadow using Digital Elevation Models (DEM). The publicly available SRTM DEM data of 30m resolution was used to carry out Range Doppler terrain correction (Filipponi, 2019).

1. **Feature Engineering**

The corrected images were then:

* Mosaic-ed according to the similar dates of the pass,
* Extracted by agricultural mask associated with the region of interest,
* Layer-stacked for all the bands into a single layer and raster file.

## **Training data**

Ground truth and land use categories were collected so that ground truth information is widely spread and represents all land use categories within the region of interest or district.

The ground truth information for both target and non-target crops was collected using MapPad software. The Polygon shapefiles for each of the fields for target and non-target crops were collected. To perform supervised classification, good quality Ground Truth (GT) points or training data are required. GT points are the polygon or point features using which supervised classification classifies the images into different categories. The spread and number of training points play a significant role in supervised classification. As a thumb rule, the number of points for the optimum number of training points should preferably be 10 times the number of variables used in the classification. The larger the size of the training sample, the more is the spread of training points, and so the higher accuracy. Therefore, theoretically, the total training points to be collected should be equal to 10 or 100 times the no. of variables. For the current study, the number of variables differs with the region of interest. A common understanding is minimum of 10 variables will be present in each district therefore, the total points which can be collected are equal to approximately 100 points for each crop type. The higher number and spread over the district will capture the geographical variability in the crop. All the non-target classes in training samples should be present otherwise there is a chance of misclassification. The Ground truth points collected are split into training and validation points. Out of the total points, 75% of the samples were used for training the model and 25% of the samples were taken to validate the final classification result.

To achieve a higher classification accuracy, the spatial distribution of training points also matters. In different parts of the AOI, the classes or the crop in this case may have some different variations and complications. Therefore, to better capture the dynamics of crops with varying sowing and harvesting time across the district, the spatial distribution of the training points should be well distributed across the region of interest.

## **Machine Learning-based Classification**

The pre-processed image obtained from the above steps was further used for classification. In the proposed study, ML algorithms have been used for image classification. Many of them have been developed over the past decade to carry out the land use land cover classification. Out of all the available machine learning, Random Forest, Support Vector Machine, and K-Nearest Neighbor have gained much popularity as these algorithms are insensitive to noise data which makes them convenient to use in unbalanced data (Breiman, 2001). Here, the following algorithms have been used:

- Random Forests

- K-Nearest Neighbors

- Support Vector Machines

- Decision Trees

- Naïve Bayes

- Multi-Layer Perceptron

These classifiers are the most utilized classifiers in Machine learning-based classification studies. Hyperparameter tuning has been done to enhance the performance of mentioned algorithms. (Table 4.)

| **S. No.** | **Algorithm/Classifier** | **Tuned Hyperparameters** | **Used Values** |
| --- | --- | --- | --- |
| 1. | Random Forests | Number of features to consider at every split  Maximum number of levels in the tree  Minimum number of samples required to split a node  Minimum number of samples required at each leaf node  Method of selecting samples for training each tree | ['sqrt']  [1000,2000]  [3, 5]  [1, 2]  [True, False] |
| 2. | K-Nearest Neighbors | Leaf size  Number of neighbors  Number of candidates | [1,50]  [1,30]  [1,2] |
| 3. | Support Vector Machines | Cost  Gamma  Kernel | [0.1, 1, 5, 10]  [1,0.1,0.01,0.001]  ['rbf', 'poly', 'sigmoid'] |
| 4. | Decision Trees | Splitter  Maximum depth  Minimum sample leaf  Minimum weight fraction leaf  Maximum features  Maximum leaf nodes | ["best", "random"]  [1,3,5,7,9,11,12]  [1,2,3,4,5,6,7,8,9,10]  [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]  ["log2", "sqrt", None]  [None,10, 20, 30, 40, 50, 60, 70, 80, 90] |
| 5. | Naïve Bayes | Priors  Var smoothing | [None, [0.1,] \* len(number\_of\_class)]  [1e-9, 1e-6, 1e-12] |
| 6. | Multi-Layer Perceptron | Hidden Layer Size  Activation function  Solver  Alpha  Learning Rate | [(10,30,10),(20,)]  ['tanh', 'relu']  ['sgd', 'adam']  [0.0001, 0.05]  ['constant', 'adaptive'] |

Table 4. Hyperparameter tuning variables and their values of the used ML classifiers

1. **GUI and Testing**

**Graphical User Interface**

A user-friendly graphical user interface was designed to ensure effective communication between the user and the system for clearer inputs to the software. To set up the inputs, the main window i.e. Figure 3. is displayed.

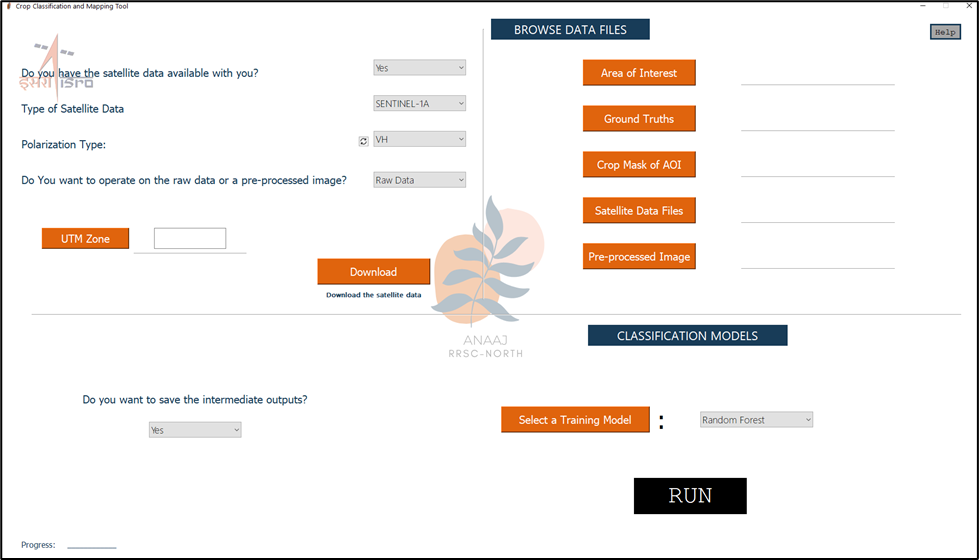


Figure 3. GUI window to enter user inputs.

It has the following user inputs:

1. If the data is available with the user?
2. Yes
3. No, Download
4. Type of satellite data
5. Sentinel-1A
6. EOS-4/RISAT-1
7. Polarization type
8. For Sentinel-1A: VV, VH, VV+VH
9. For EOS-4: HH, HV, HH+HV
10. Operation on raw data or pre-processed image?
11. UTM Zone of the AOI
12. DOWNLOAD Button to set up the start and end dates of the data in case it is not available locally
13. Area of Interest (.shp file)
14. Ground Truth Points (.shp file)
15. Crop Mask of the AOI (.shp file)
16. Satellite data files in case available
17. Pre-processed image tif file in case of operation on it
18. Training ML model type
19. RUN button to run the automation process

On clicking the DOWNLOAD button, Figure 4. is displayed where the user can select the start and end date of Sentinel-1A data to be downloaded.

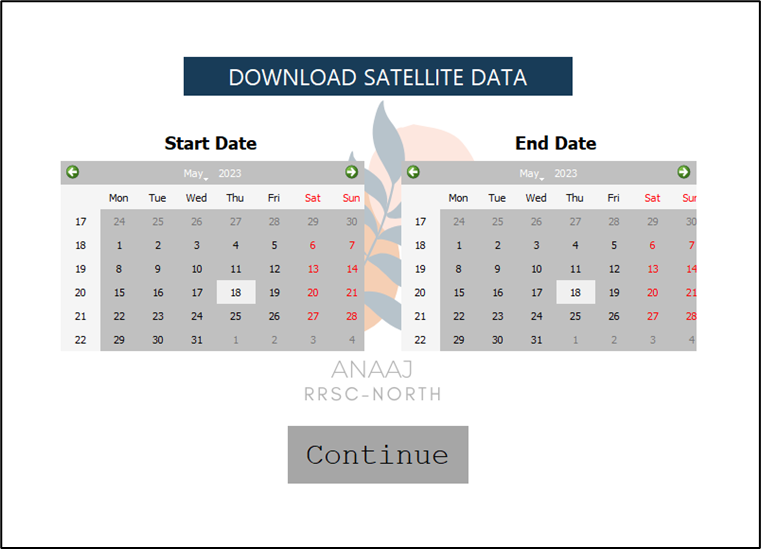


Figure 4. GUI window to select the temporal range of the data to be downloaded.

**Experimentation and Testing on sample data**

The current test has been carried out with data of the Aligarh district of Uttar Pradesh, located between latitudes 27°57'N and 28°18'N and longitudes 77°48'E and 78°61'E, in the southeastern part of the state. E. The district is 193 meters above the mean sea level. The great rivers Ganga and Yamuna, which come from the northeast and northwest sides, respectively, border the district. From the northwest, the Palwal district of Haryana, from the northeast, Badaun, from the north, Bulandshahar, from the north, Mathura, from the west to the southeast, Hathars, and from the south and east, Etah. The district's median annual rainfall is 662.8 mm (Department of Agriculture Cooperation & Farmers Welfare, 2019). The district's typical temperature is 25.2°C. The 2011 census revealed that Aligarh has a population of roughly 3.67 million (Census of India, 2011). The district's reported total area is 371300 ha, of which around 304000 ha are net sown (Agriculture, Cooperation and Farmers Welfare, 2012). (Figure 5.)

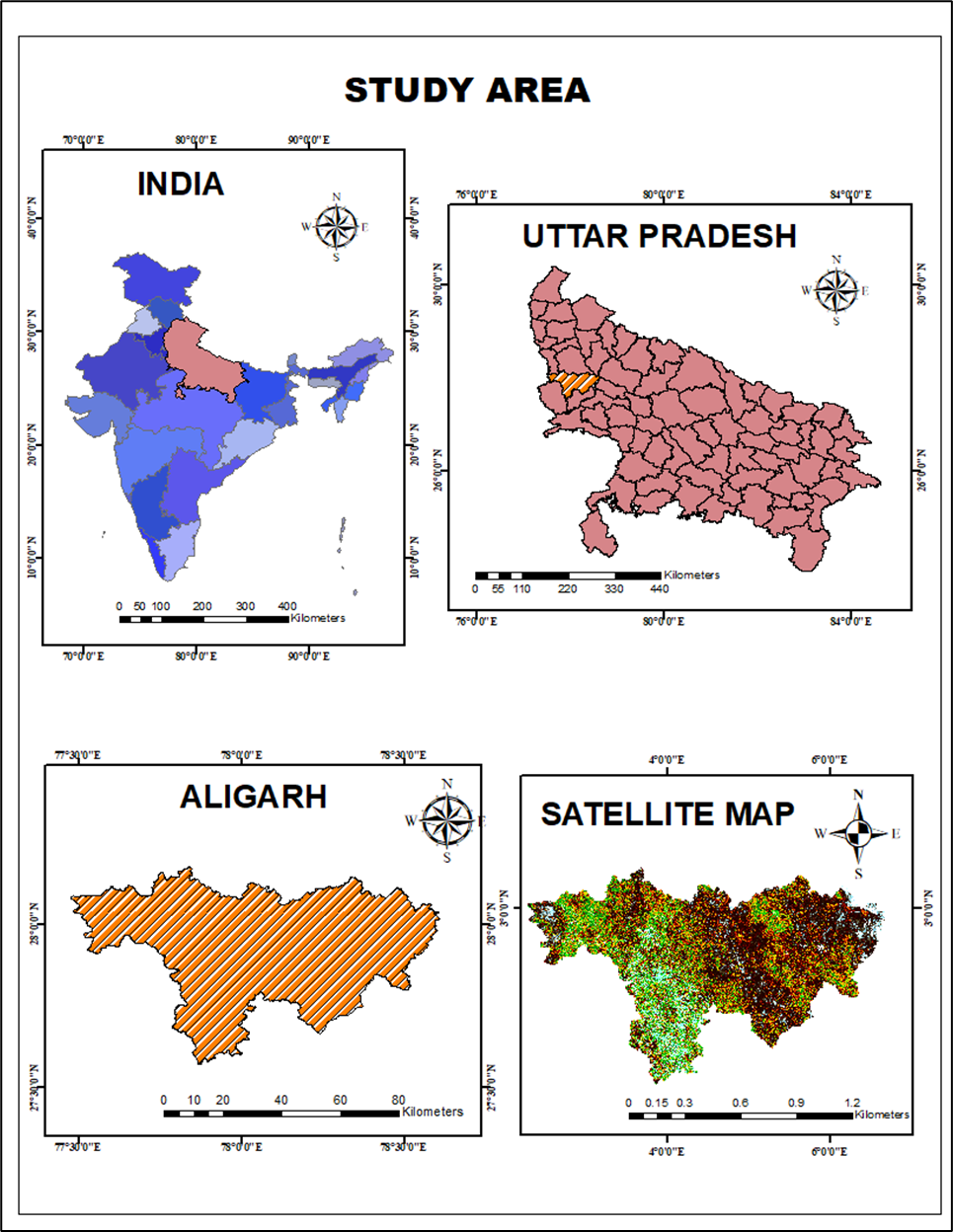


Figure 5. The study area for testing.

**Data Download**

While downloading, a complete list of the data to be collected is displayed in the command line, along with the relative orbit of all the data files. (Figure 6.)

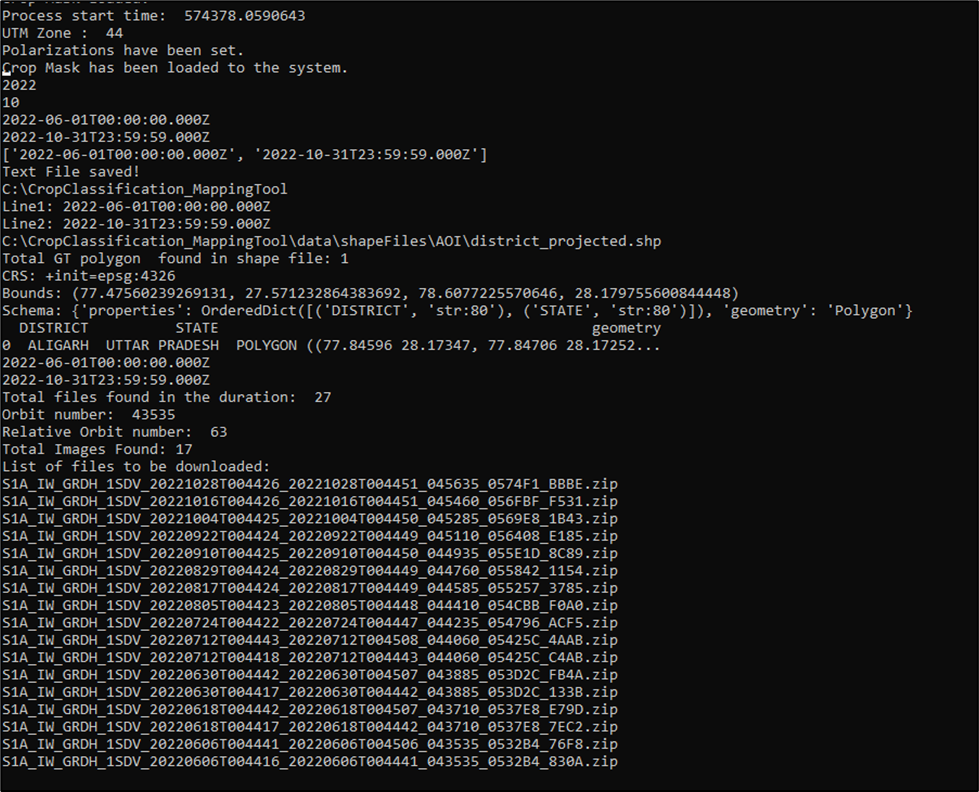
****

Figure 6. Command line output for data download.

**Backscatter Curve**

After preprocessing, a backscatter curve is generated using the layer-stacked preprocessed image and the ground truth points for each crop class over the temporal data collected. Figure 7. shows the curve generated for the data from June 2022 to October 2022, and crop classes Arhar, Bajra, Cotton, Fallow\_Land, Jowar, Maize, Paddy, and Sugarcane. The more the backscatter coefficient value more is the probability of the crop class being present at the field during that time.

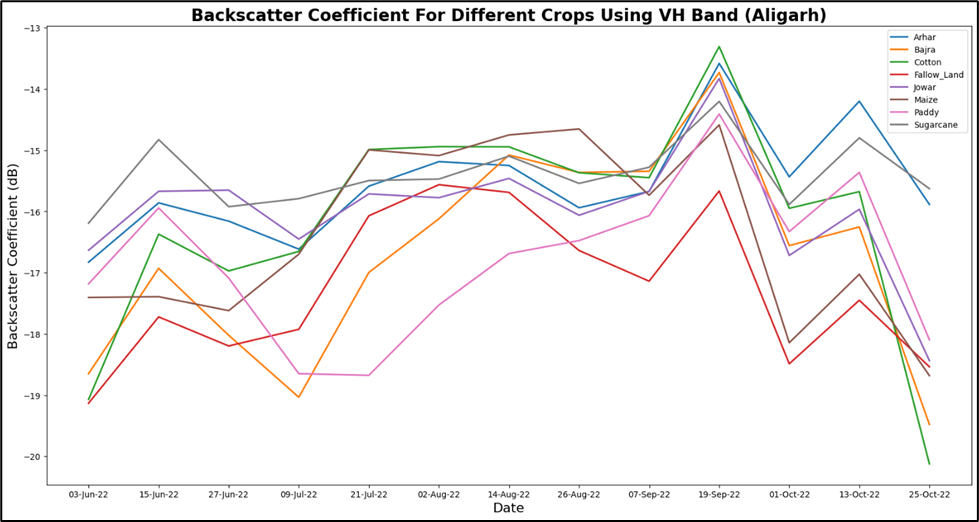


Figure 7. Backscatter curve generated by the automation tool.

**Classified Image**

After preprocessing, the image is classified using the selected machine learning algorithm. Figure 8. shows the classified image for a set of crop classes over the area of the Aligarh district. An additional automated 4-page report is also generated by the system at the end to summarize all the outputs of the flow of processes.

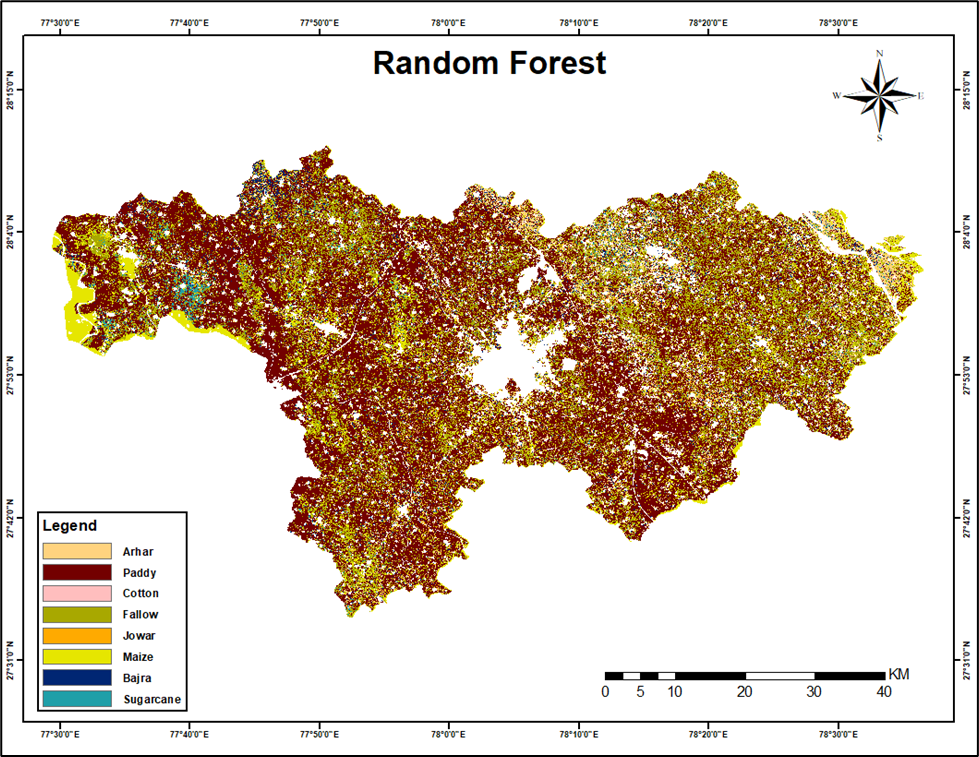


Figure 8. Classified image generated by the automation tool.

1. **CONCLUSION**

This paper aims at observing the automation process for area estimation of a group of crops over an area. Various tests were carried out with some areas of interest and the automation tool registered an accuracy of over 90% for the classification of the crops. The study concludes that by automating the process of data acquisition using ASF’s Python search module, processing using SNAP, data engineering using GDAL and rasterio, classification with fine-tuning of the machine learning algorithms hyperparameters for their sensitive range, and estimation of the crop acreage by transforming the resultant classified data, there is an improvement of almost 20 to 30 percent of accuracy in the performance of classification process, time consumption has reduced from weeks or even months to days, the source code is optimized, the GUI is user-friendly and easy to understand, space complexity has been optimized according to the user input, and data is made readily available.

With the availability of more ground truth points, we can implement deep learning techniques to improve the predictability and performance of the models. Processing optical data can also be done through programming and automated for further estimation.

1. **REFERENCES**