Map rice area using extracted Sentinel-1 pixel-based time-series report

Student name: Tran Quoc Khanh

Student ID: 20190085

1. Introduction

Last decade witnessed a rapid development of Computer Science. Computers are more affordable and accessing to internet is available for many people. The amount of data we are generating nowadays is tremendous. Combining with high computing capability, computer can do many tasks which were impossible in the past. The ability to understand information and extract insights is important.

Applying machine learning and deep learning technique, we can gain a lot of information. Many deep learning models, such as Convolutional Neural Network has powerful ability to extract information, without feature engineering, like traditional machine learning model. They are used popularly.

In this project, we will use satellite imageries of Sentinel-1 to classify pixels to land cover types, such as forest or rice. Our model proposals are Convolutional Neural Network and Recurrent Neural Network. The second section describes dataset and discusses why pattern recognition is needed. Following sections, the third and fourth section talk about model implementations, Convolutional Neural Network and Recurrent Neural Network. Overall performance as well as classification report of each model will be addressed.

2. Data

The dataset we used in this project is captured by Sentinel-1 Satellites in the year 2020. There are 27 sites in this data, the number of pixels in each site ranges from hundreds to over fifty thousand pixels. Each point is determined by site no and pixel no in this site. There are 190130, nearly two hundred thousand points in total.

The Sentinel-1 has 12-day repeat cycle. This means that it takes the satellite 12 days to recapture a position in Earth. As a result, time step of each time series is 12 days and length of them are 30. The satellite capture information by transmitting wave and receive reflected wave. Type of received data is determined by technique. In this dataset, for each time point we have two types, vertical transmit and vertical receive, VV and vertical transmit and horizontal receive, VH.

We have five types of land, Crop, Forest, Rice, Urban and Water. There are over one hundred thousand point of Forest class. The number of Rice point is also large, about seventy thousand. On the other hand, Crop, Urban Water have a small number of points. In case of Water, we only have 355 points, which is tiny comparing to Forest. Our dataset is imbalanced, which can lead to potential issues. To make sure training set contains all classes, we split train and test set with ratio of train set are 0.5, 0.4, 04, 0.7 and 0.8, corresponding to class Crop, Forest, Rice, Urban and Water.

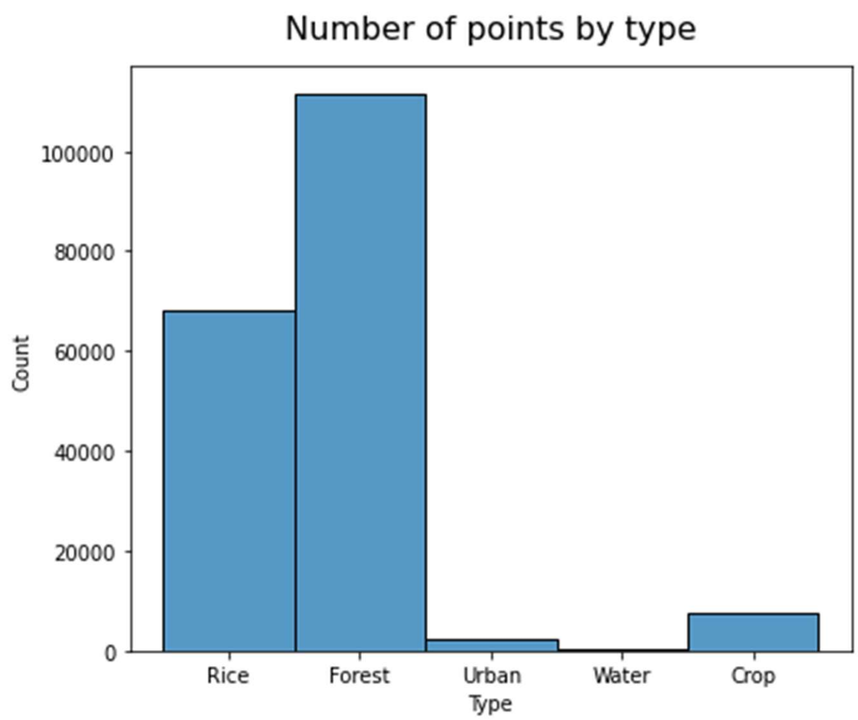

Figure 1.1 Number of points in each class.

Following figure plot some examples of each land type. Left axis is VV polarization and the right is VH of the same point. One thing can be observed from this example is that the value of land type is not stable, it alters slightly, even greatly in the example of rice point. Hence, classification by using threshold for one time step might not work. Image segmentation can be an approach. However, satellite imageries are not captured at one time like RGB images. The relationship between pixels tends to be lower. Therefore, image segmentation might not be a good choice. Because one point is captured multiple times, we propose using time series classification. Each point will be classified by its represented time series independently.

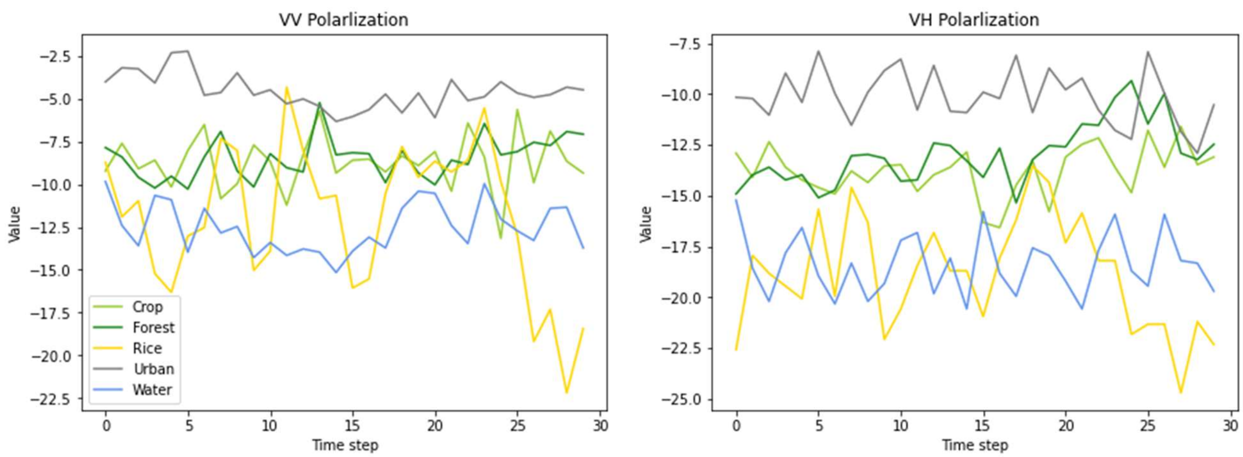

Figure 1.2 Plot of some time series.

3. Convolutional Neural Network

In time series classification, we cannot apply straightly machine learning algorithms, because these algorithms do not take into account pattern in series, each feature is treated separately. For example, if we change order of features, the model might not change, but order of feature is important in time series. Hence, it often requires some specific feature engineering techniques to preprocess the data, Dynamic Time Wrapping for instance. Fortunately, Artificial Neural Networks can address this problem. In practice, Convolutional Neural Networks and Recurrent Neural Networks show high performance in time series classification task. We will talk about CNN in this section, and RNN in the next one.

3.1 Model architecture

As we know, Convolutional Neural Network is widely used in computer vision tasks, such as image classification, objects detection. It used a 2D kernel, which moves over the input data, to extract features. The same technique can be applied to time series, 1D kernel is applied instead of the 2D. We applied a simple 1D CNN model for our problem. Following is the summary of our architecture.

Layer	Detail
Conv1d-1	Number of kernels: 4 Kernel size: 3

	Stride: 1
Conv1d-2	Number of kernels: 8 Kernel size: 3 Stride: 1
Conv1d-3	Number of kernels: 8 Kernel size: 3 Stride: 1
Conv1d-4	Number of kernels: 10 Kernel size: 3 Stride: 1
Linear-5	Input size: 220 Output size: 64
Linear-6	Input size: 64 Output size: 16
Linear-7	Input size: 16 Output size: 5

Table 3.1 CNN model architecture.

The first four layer are 1D convolutional layer, which has kernel of size 3 and stride equals to 1. Next, we use three fully connected layer to output a tensor of size 5, that represents the probability of each class. Leaky ReLU activation function is used. Total number of parameters are over fifteen thousand.

3.2 Results

We trained our model on VV, VH and both VV and VH data. In both VV and VH implementation, input of model is two channels, each channel is one polarization type. The number of channels can be extended when we have more bands. Training loss of these model is plotted in the following figure. Model with only VH polarization converts slowly and has the highest loss almost all the time, whereas VV model is slightly better. The model which has lowest loss is both polarization, which convert more quickly than two others.

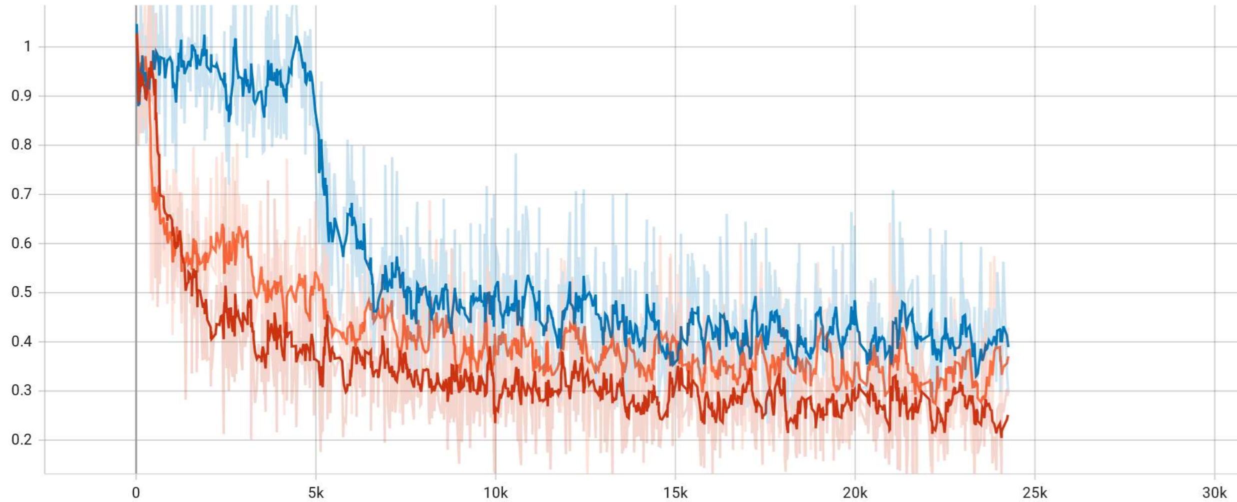

Figure 3.1 Training loss of CNN with different polarization. The blue and orange curve are loss of models which have input is only VH and VV, correspondingly. Red curve is both polarizations model.

Both polarization model performs better than only one is obvious, because it contains more information. Its performance is the highest in all measurement, nearly 0.9 precision, recall, F1 and accuracy score. Following is VV polarization model and the last one is VH model.

	Weighted precision	Weighted recall	Weighted F1-score	Accuracy
VV polarization	0.86	0.857	0.854	0.857
VH polarization	0.848	0.842	0.838	0.842
Both VV and VH polarization	**0.892**	**0.889**	**0.887**	**0.889**

Table 3.2 Performance of CNN model, using VV, VH and both VV and VH data.

However, our dataset is imbalanced. We have much more Forest and Rice point than the others. The number of Water points is only 355, much smaller than Forest, over one hundred thousand points. Consequently, the model might perform badly on small classes. Classification report of both polarization model is described in the heatmap bellow. Forest class has the highest recall score. The highest precision score is Rice class, but recall score of this class is quite lower, nearly 0.79. Performance in small classes like Urban and Water is bad, only around 0.6 in all measurement.

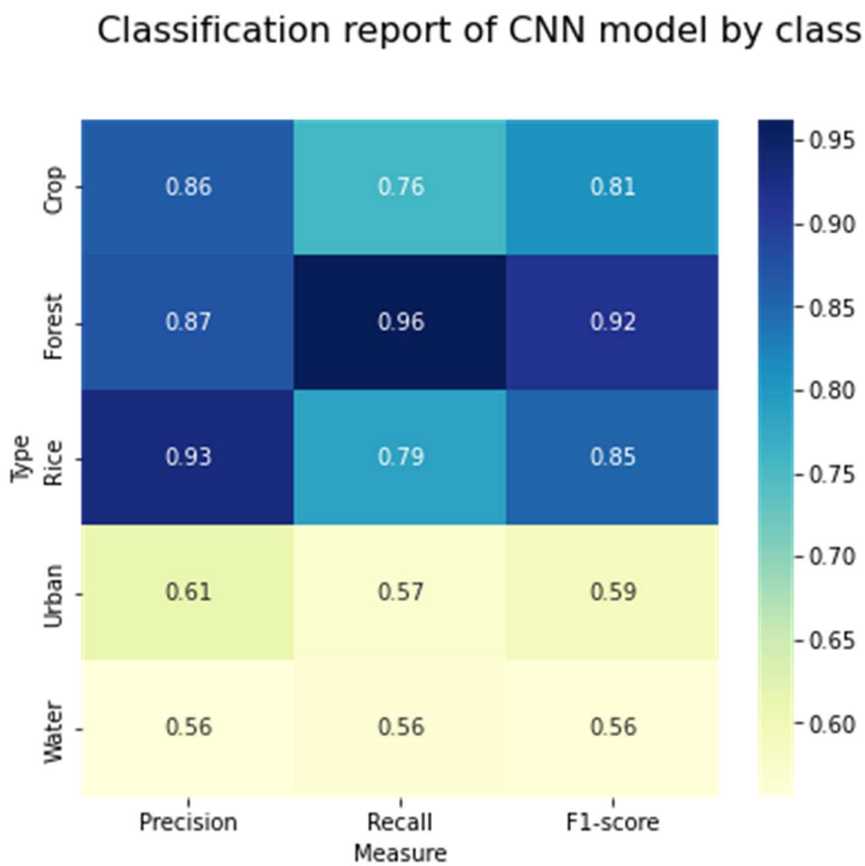

Figure 3.2 Classification report of CNN model, using both VV and VH data.

4. Recurrent Neural network

4.1 Model architecture

Recurrent neural networks can be applied to time series classification task, as well as CNN. When CNN model requires time series have the same length, RNN does not. It is an advantage of RNN. Because the task is not too difficult, we propose to implement Vanilla RNN, without any memory or attention mechanism.

Layer	Detail
RNN-1	Input size: 2 Hidden size: 60
Linear-1	Input size: 60 Output size: 32
Linear-2	Input size: 32 Output size: 5

Table 4.1 RNN model architecture.

The main layer of our model is RNN layer. It takes input as a time point, so the input size is two. It will output a tensor of size 60. Number of stacked layers in RNN part is consider as a hyperparameter, which we need to initialize. The output of last time point is input to fully connected layer to classify the time series. It is many-to-one architecture. The activation we use is also Leaky ReLU with 0.1 slope.

4.2 Results

As described in previous section, the number of stacked layers in RNN part is a hyperparameter. We will train some model with different values to choose the most suitable one.

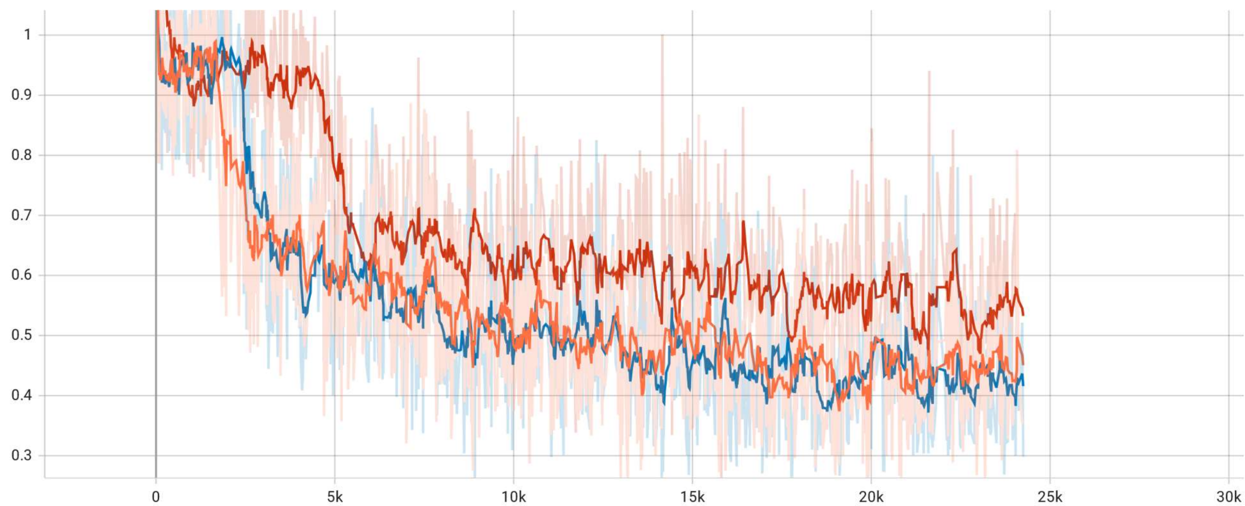

Figure 4.1 Training loss of RNN with different number of stacked layers. The orange, blue and red curve are training loss of two, three and four stacked layer RNN, respectively.

The figure above plot training loss of RNN model. Orange curve is two stacked layers RNN's loss curve, the blue and red are three and four layers RNN, respectively. We see that loss of four stacked layers RNN is higher than two and three layers, which are nearly identical. The overall performance of three layers RNN is highest in all measure, slightly better than two layers model. Whereas four layers model has the worse score.

	Weighted precision	Weighted recall	Weighted F1-score	Accuracy
Two layers RNN	0.859	0.856	0.853	0.856
Three layers RNN	**0.866**	**0.859**	**0.854**	**0.859**
Four layers RNN	0.829	0.826	0.815	0.826

Table 4.2 Performance of RNN model with different number of stacked layers.

Similar to CNN model, RNN tends to perform poorly on small classes. The following heatmap is classification report of three stacked layers RNN. In this figure, the measurements in Forest and Rice class are highest, linearly reduce to Crop, Urban and Water. Performance in Water class is very poor, with precision and recall scores are only 0.24 and 0.07, correspondingly.

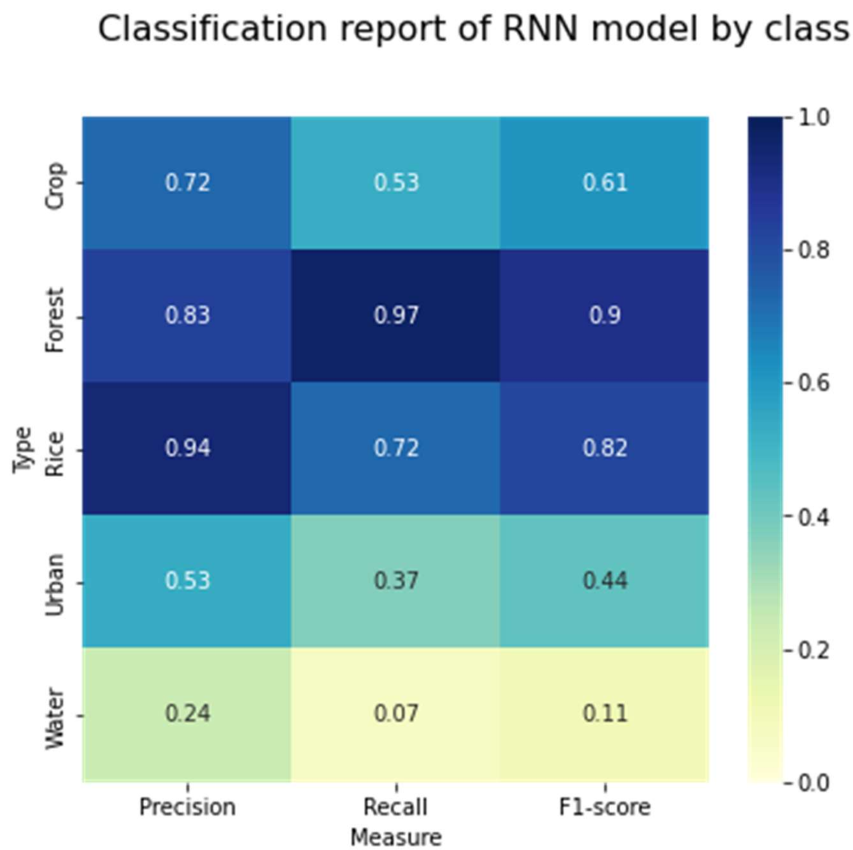

Figure 4.2 Classification report of RNN model with two stacked layers.

5. Conclusion and future development

In this project, we have to classify point to land type, Crop, Forest, Rice, Urban and Water, using time series captured by Sentinel-1 Satellite. The implemented model are Convolutional Neural Network and Vanilla Recurrent Neural Network. The performance of CNN model using both VV and VH data is higher than using only one VV or VH, and higher than Vanilla RNN, too. However, RNN model is more light weight than the CNN. It also does not require input time series have fixed length, which is needed in CNN model.

Both implementations have poor performance in small classes, Urban and Water. Because our dataset is imbalanced, we have much more Forest and Rice points, bad score in small classes is necessary. It is natural, water and urban area is much smaller than other. In practice, if accurate classification of these land type is needed, we can implement some techniques to handle imbalanced dataset. Over-sampling small and under-sampling large classes can be a simple solution. Besides that, we can consider weighting more in loss of small classes. If these techniques do not work, collecting more data is essential.

6. References

Jason Brownlee. 1D Convolutional Neural Network Models for Human Activity Recognition. Machine Learning Mastery (Sep 21, 2018) https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/

Piero Paialunga. Hands-On Climate Time Series Classification with Deep Learning, using Python. Towards Data Science. (Aug 6, 2021) https://towardsdatascience.com/hands-on-climate-time-series-classification-with-deep-learning-using-python-6d5de81004c9

Marco Del Pra. Time Series Classification with Deep Learning. Towards Data Science (Sep 8, 2020) https://towardsdatascience.com/time-series-classification-with-deep-learning-d238f0147d6f