ETH Zurich
Image Interpretation, HS 2021

Lab Report 3

Time Series

Marie-Josianne Fandré, Valérie Hellmüller, Pascal Imhof, Yatao Zhang

December 15^{th}, 2021

Professorship .... Prof. Dr. Konrad Schindler
Supervision .... Mehmet Özgür Türkoglu

1 Introduction

For the third lab within the course Image Interpretation the goal is to classify crop types with help of satellite pictures taken in a time series. The task of our group is to do the classification with conventional classifiers, such as random forest and support vector machine.

2 Methodology

2.1 Pre-Prossessing

The received data consists of satellite images which are showing fields in Zürich at 71 different days in the time period from January to December 2019. In total there are over 50 different crop types available on these images. According to the task sheet only 13 of them are used. Depending on the type of grain, a different number of pixels is available. The range is from $3'574'346$ for meadow to $47'277$ for spelt. To ensure that the classifiers train and validate the same number of pixels for each grain, $40'000$ pixels are randomly sampled from each crop type. The resulting dataset is split into 75% training data and 25% validation data. To split the data, every fourth pixel was assigned to the validation dataset. For each pixel there is RGB (red, blue, green) and NIR (near infrared) available. To expand the information content for the classifiers the following indexes are calculated:

- Normalized Difference Vegetation Index (NDVI)
- Atmospherically Resistant Vegetation Index (ARVI)
- Green Chlorophyll Index (GCI)

The NDVI is classically used in time series of vegetation because it indicates the "greenness" of the picture by giving a ratio over the Red and the NIR band.
The ARVI is derived from the NDVI, it has the same information content but it can be used in regions with a high aerosol. By taking the Blue Band into the equation it makes it more robust to air particles and topographic effects
The GCI is used, as the name suggests, to evaluate the content of chlorophyll. The chlorophyll content leads to the health of the plant or shows its seasonality. [2]

Further analysis of the data showed that day 12 in the dataset does not contain a lot of information. A lot of the values are equals to zero or NaN. Due to that this day is not used for the following computations.

2.2 Classification methods

This lab aims to utilize the traditional classification methods to classify time-series crop data. Therefore, in this lab, we experimented with three kinds of classification methods introduced by Professor Konrad Schindler in the course. They have different feature engineering methods to handle the time-series classification task, including stacking pixel-wise features, aggregating features across time, and aggregating predictions across time.

2.2.1 Stacking pixels

This method directly input raw time-series data and derived pixel-wise features into the classifier without considering the time-dependent features. In detail, given the original data set, we reordered per-pixel features from all frames into a vector in a fixed order, and then fed it to the selected classifier. In this lab, we concatenated the RGB band, NIR band, NDVI, ARVI and GCI features in 71 days by order into a long sequence to train the classifier.
The advantages are that there is no loss of information and it's simple to operate by using this method. However,

it may lead to a very large context or long sequence that occupies a huge feature space which is inefficient. Also, this method could fail to model the relevant information and temporally changing features, which may affect the performance.

2.2.2 Aggregating features across time

Aggregating features across time refers to summarizing time-series features along the time dimension. These time-series features can be obtained through calculating statistical indicators and exploiting local or global order. In the lab, we have the following time-series features based on the temporal variations of the above per-pixel features (RGB band, NIR band, NDVI, ARVI and GCI):

- Statistical indicators: max / min / mean / variance per pixel.
- Features of local order: moving average with a sliding window of 5 to ignore the local fluctuation.
- Features of global order: the slope of linear trend per pixel, which is calculated based on the least-square method on linear fitting.

By aggregating features across time, some temporal features of time-series data can be extracted to model its time-dependent characteristics. But some information is lost because the temporal sequence is reduced to a less-dimensional feature vector.

2.2.3 Aggregating predictions across time

The method consists of two steps: first, in every time step, we train a classifier separately based on the above per-pixel features (RGB band, NIR band, NDVI, ARVI and GCI) to predict the labels of test data; second, the method aggregates all the predictions in 71 days and then produces the final output for each pixel through voting. The label that has the maximal votes will be regarded as the final label.
Utilizing this method to handle the time-series classification task can guarantee that the data in every step is less, but most of the temporal information will be lost and cannot model temporally changing outputs. Meanwhile, if the time dimension is too large, training multiple classifiers one time may waste lots of time.

2.2.4 Classifier Choice

Based on above three classification methods, this lab selected three different classifiers to evaluate their performance, including random forest (RF)[4], support vector machine (SVM) [3] and k-nearest neighbors (KNN) [1]. The number of trees in RF was set as 100, and the number of neighbors in KNN was set as 3. In addition, accuracy, precision, recall and F1 score were utilized to assess for these classification methods.

3 Results and Discussion

This lab experimented with a Linux system with 340GB RAM, 96 threads and 1 TESLA V100 16GB GPU.

3.1 Running time and memory

Table 1: Running time and required memory for the training and validation: time (s) / memory (GB)

Classifier	Stacking pixels	Aggregate features	Aggregate predictions
RF	1109.71 / 3.68	1378.53 / 5.17	6526.30 / 6.03
SVM	5613.29 / 3.68	4992.55 / 5.18	14541.99 / 6.05
KNN	527.67 / 3.73	507.91 / 5.19	313.00 / 6.14

In table 1, the running time and memory required for training the classifiers presented in Chapter 2.2.4. The KNN classifier has the highest memory requirements while having the lowest training time. This is caused by the algorithm itself which only stores the training data in a more optimised way but it does not try to generalize the data as presented by [5]*. Only later in the prediction part, the real classification will happen when the algorithm searches the nearest neighbours within the training data for each validation or test pixel. The SVM is the slowest algorithm while it still needs a relatively high amount of memory. This is due to the bad scaling of this method concerning memory and computation time as shown by [5]†. The memory consumption of the RF is the lowest while the training time is tremendously lower than for the SVM classifier. But the training time as well as the memory consumption depends heavily on the chosen hyper-parameters as this algorithm generalises the classification. These parameters are especially on the tree depth, the maximum number of random trees used and how many features are used for a decision.

3.2 Comparisons of different methods

The performance assessment with three different classification methods based on three classifiers for the validation data is shown in Table 2 and Figure 1.

Table 2: Overall accuracy for the training and validation dataset

Classifier	Stacking pixels	Aggregate features	Aggregate predictions
RF	0.9314	0.9249	0.7967
SVM	0.8329	0.8154	0.4924
KNN	0.9334	0.9243	0.5638

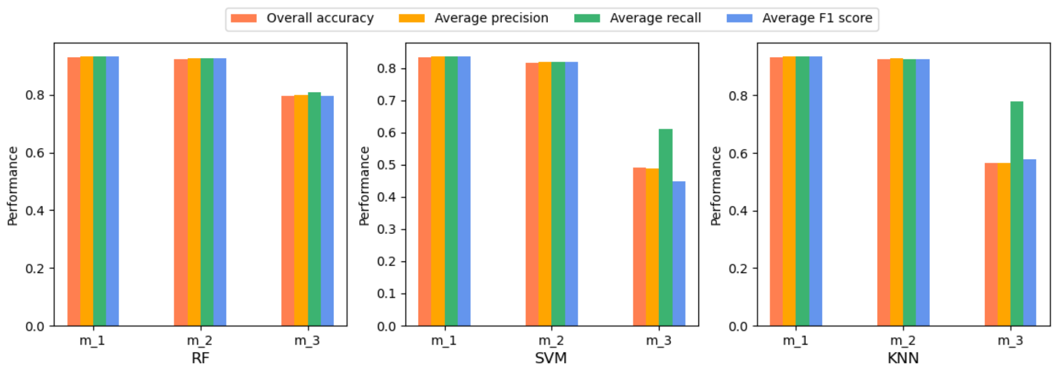

Figure 1: Performance evaluation of three classification methods based on three classifiers. m_1: stacking pixels, m_2: aggregate features, m_3: aggregate predictions

It becomes evident that the method of stacking pixels yields the best results for all three classifiers, and aggregating features across time provides similar performance to stacking pixels but its accuracy is a little lower. For these two methods, RF and KNN hold almost equivalent accuracy, while the SVM shows a bad performance. To this point, the tree-based classifier and the neighbor-based classifier can extract useful features to distinguish different types of crops, while the SVM fails to effectively project existing features to a high-dimensional feature space. In addition, aggregating predictions across time presents the worst performance for all three classifiers, especially for SVM and KNN. One assumption is that the recognition of crops is highly dependent on the time-series features and it can not be accurately classified by only considering the features in one-time epoch.

According to the above analysis, the RF has the best accuracy in classifying time-series crop data. Among all three classification methods, stacking pixels performed best, aggregating features across time followed but the gap was very small, and aggregating predictions across time had the least performance. Thus, we suggested utilizing stacking pixels based on RF as the final classification method for the test dataset. Also, to compare

*https://scikit-learn.org/stable/modules/neighbors.html
†https://scikit-learn.org/stable/modules/svm.html

their difference further, another two classification methods based on the RF were implemented in the test dataset.

3.3 Results for the test dataset and map generation

Table 3 shows the final result of three classification methods by using the RF classifier. Same as the result of Chapter 3.2, stacking pixels holds the best performance among three classification methods, and the running time and required memory also shows its superiority. However, the performance in the test dataset is not ideal compared to the validation results. Although these three traditional classification methods can be used to classify time-series data, their generalization ability and robustness are limited. Besides, the visualization result of ground truth and prediction with the best classification method is illustrated in Figure 2

Table 3: Accuracy, time and memory for the test dataset by using the RF classifier

Item	Stacking pixels	Aggregate features	Aggregate predictions
Accuracy	0.7240	0.6739	0.5339
Time (s)	1135.6375	1174.2282	8279.9842
Memory (GB)	10.5265	13.2250	12.1864

4 Conclusion and Outlook

In this lab, we explored three classification methods to handle time-series crop classification based on traditional classifiers, including stacking per-pixel features, aggregating features across time and aggregating predictions across time. The method of stacking pixels based on the RF classifier performed best but failed to generalize in the test dataset. Thus, extracting time-series features and selecting suitable models are of great significance to handle similar tasks in the future, such as the hidden Markov model (HMM) and long short-term memory (LSTM).

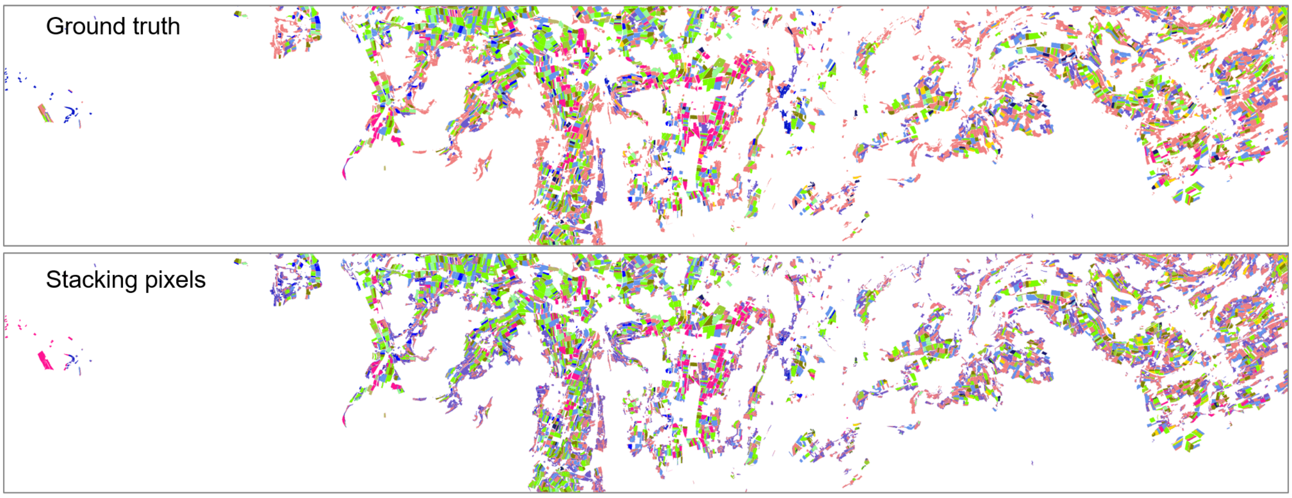

Figure 2: Map generation for ground truth and prediction results by using stacking pixels based on the RF classifier

References

[1] Zhenyun Deng et al. "Efficient kNN classification algorithm for big data". In: *Neurocomputing* 195 (2016), pp. 143–148.

[2] Vinod Kumar et al. "Comparison of different reflectance indices for vegetation analysis using Landsat-TM data". In: *Remote Sensing Applications: Society and Environment* 12 (2018), pp. 70–77.

[3] Giorgos Mountrakis, Jungho Im, and Caesar Ogole. "Support vector machines in remote sensing: A review". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.3 (2011), pp. 247–259.

[4] Mahesh Pal. "Random forest classifier for remote sensing classification". In: *International journal of remote sensing* 26.1 (2005), pp. 217–222.

[5] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.