

Sampling framework

OTB Team



07/06/2016



The context

Learning framework :

Current sample selection :

Learning framework :

- ▶ Focused on the training step
- ▶ Missing tools for the samples preparation

Current sample selection :

Learning framework :

- ▶ Focused on the training step
- ▶ Missing tools for the samples preparation

Current sample selection :

- ▶ Obscure mechanism, difficult to predict how many samples will be selected and where
- ▶ Allows to set maximum number of samples per class, training/validation ratio, bound by minimum class
- ▶ Filters not scalable

The plan

- ▶ Add new applications to allow more control on sample selection

The plan

- ▶ Add new applications to allow more control on sample selection
- ▶ Integration in the existing application TrainImagesClassifier

The plan

- ▶ Add new applications to allow more control on sample selection
- ▶ Integration in the existing application TrainImagesClassifier
- ▶ Foresee the use of large training sets



The plan

- ▶ Add new applications to allow more control on sample selection
- ▶ Integration in the existing application TrainImagesClassifier
- ▶ Foresee the use of large training sets
- ▶ Foresee the use of stratified learning



Processing steps

- ▶ Step 1 : analyse input geometries (number of candidate samples per class, and per polygon)

Processing steps

- ▶ Step 1 : analyse input geometries (number of candidate samples per class, and per polygon)
- ▶ Step 2 : select sample positions based on sampling rates per class, and sampler type

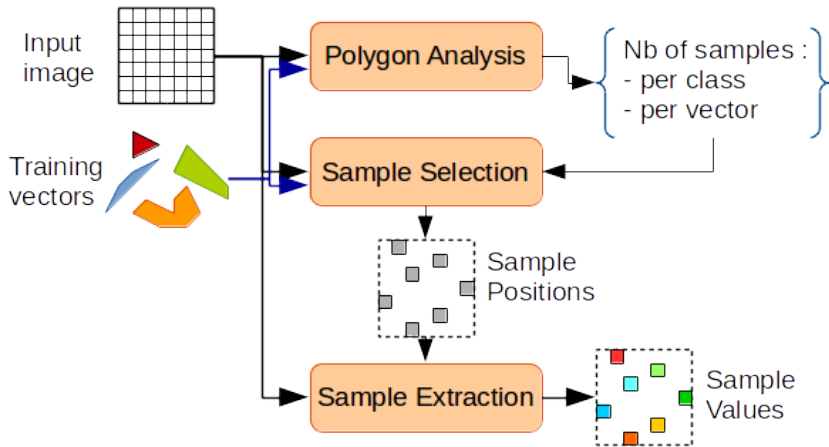
Processing steps

- ▶ Step 1 : analyse input geometries (number of candidate samples per class, and per polygon)
- ▶ Step 2 : select sample positions based on sampling rates per class, and sampler type
- ▶ Step 3 : extract sample values

Processing steps

- ▶ Step 1 : analyse input geometries (number of candidate samples per class, and per polygon)
- ▶ Step 2 : select sample positions based on sampling rates per class, and sampler type
- ▶ Step 3 : extract sample values
- ▶ Step 4 : ML training from the set of samples

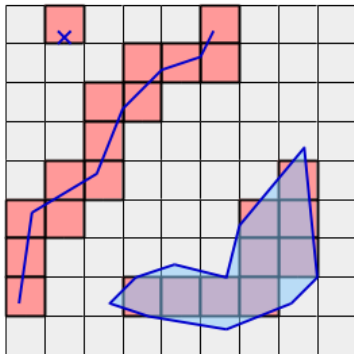
The big picture



Handling of input geometries

Depending on the input training geometry :

- ▶ Point : select closest pixel
- ▶ Line : select pixels intersecting the line
- ▶ Polygon : select pixel centers inside the polygon



Sampling strategies

- ▶ Take all available samples
- ▶ Set a common number of sample for each class
- ▶ Fit the number of samples based on the smallest class
- ▶ Select different sample counts for each class



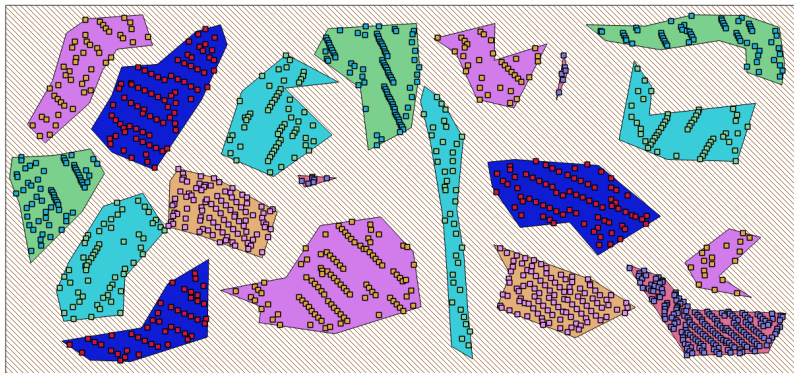
Sampler types

How to select N samples among T ?

- ▶ Periodic : an improvement of the decimation method
- ▶ Pattern : uses a repeated custom selection pattern (X__X__X__...)
- ▶ Random : uses a random test at each candidate with a given probability



Example



Implementation aspects

How it is done :

Implementation aspects

How it is done :

- ▶ Uses streaming on the support image

Implementation aspects

How it is done :

- ▶ Uses streaming on the support image
- ▶ Uses multi-threading over the input geometries

Implementation aspects

How it is done :

- ▶ Uses streaming on the support image
- ▶ Uses multi-threading over the input geometries
- ▶ Supports input raster mask

Implementation aspects

How it is done :

- ▶ Uses streaming on the support image
- ▶ Uses multi-threading over the input geometries
- ▶ Supports input raster mask
- ▶ Handles geometry collections

Current state

- ▶ Step 1 : PolygonAnalysis : first implementation available in 5.4
- ▶ Step 2 : SampleSelection : in RFC process
- ▶ Step 3 : SampleExtraction : work in progress
- ▶ Step 4 : TrainVectorsClassifier : work in progress
- ▶ Integration into TrainImagesClassifier : to do

Future evolutions

- ▶ Better multi-image support
- ▶ Stratified learning
- ▶ Sample values computed during extraction