

H2I Workshop

Spatial Classification of Hyper-spectral Images

16/09/2021

unibz
Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

efre·fesr
Südtirol · Alto Adige

Europäischer Fonds für regionale Entwicklung
Fondo europeo di sviluppo regionale



EUROPEAN UNION



AUTONOME
PROVINZ
BOZEN
SÜDTIROL

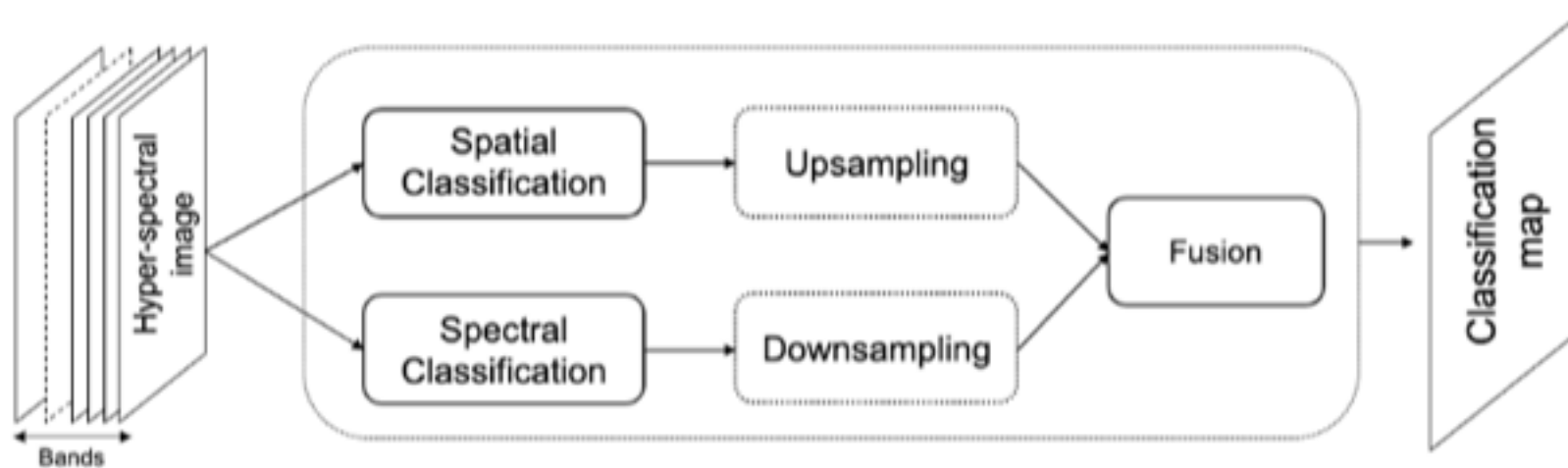


PROVINCIA
AUTONOMA
DI BOLZANO
ALTO ADIGE

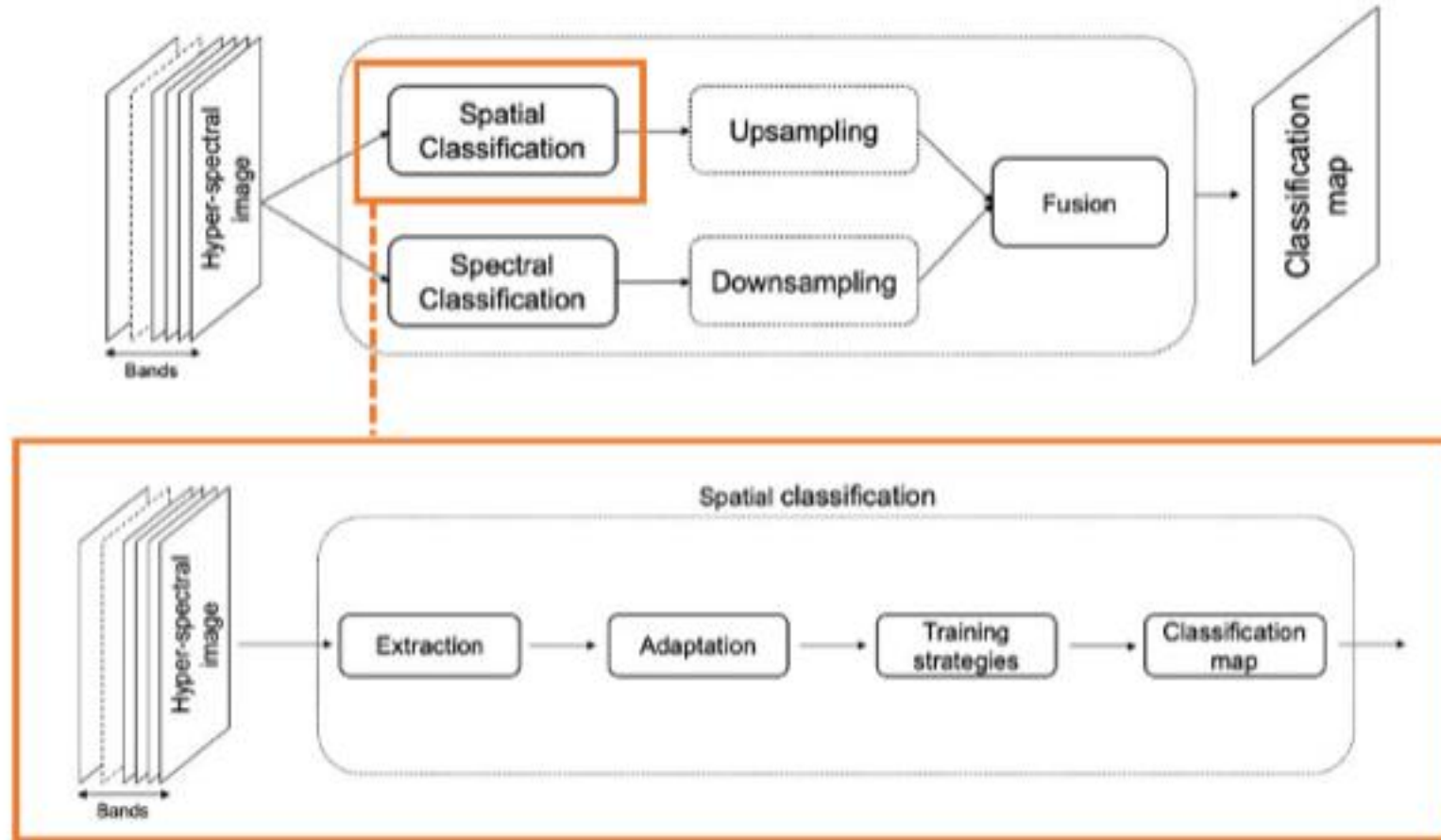
Outline

- Segmentation Framework for Hyper-spectral Images Classification
- Segmentation Framework – Spatial Classifier
- Segmentation Framework
- Architecture of classifier
- Apply methodologies
- Training the output unit
- Training the input unit
- Single block classifier into multi block classifier – called segmentor
- Segmentation result
- Expected Benefits

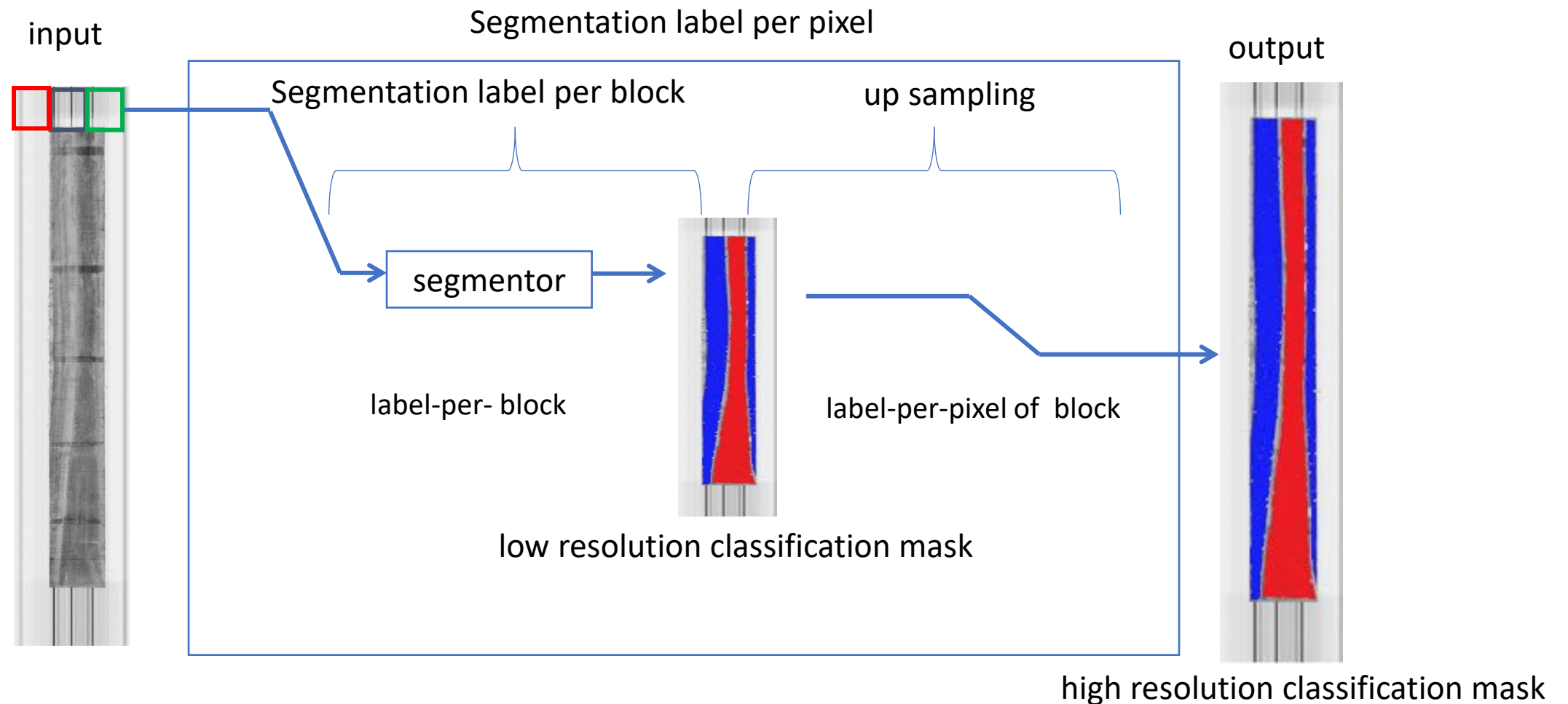
Segmentation Framework for Hyperspectral Images Classification



Segmentation Framework – Spatial Classifier

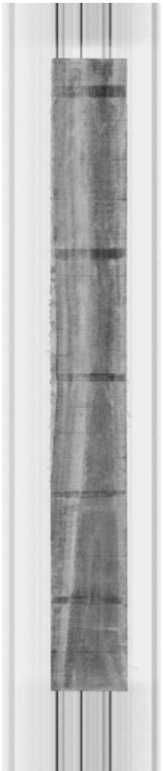


Proposed Framework

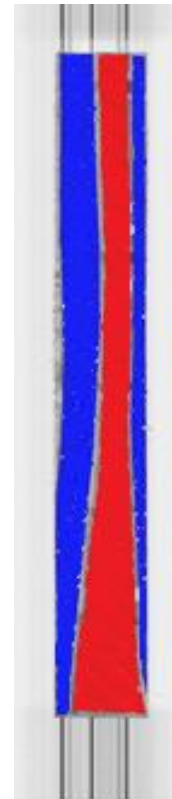


Proposed Framework

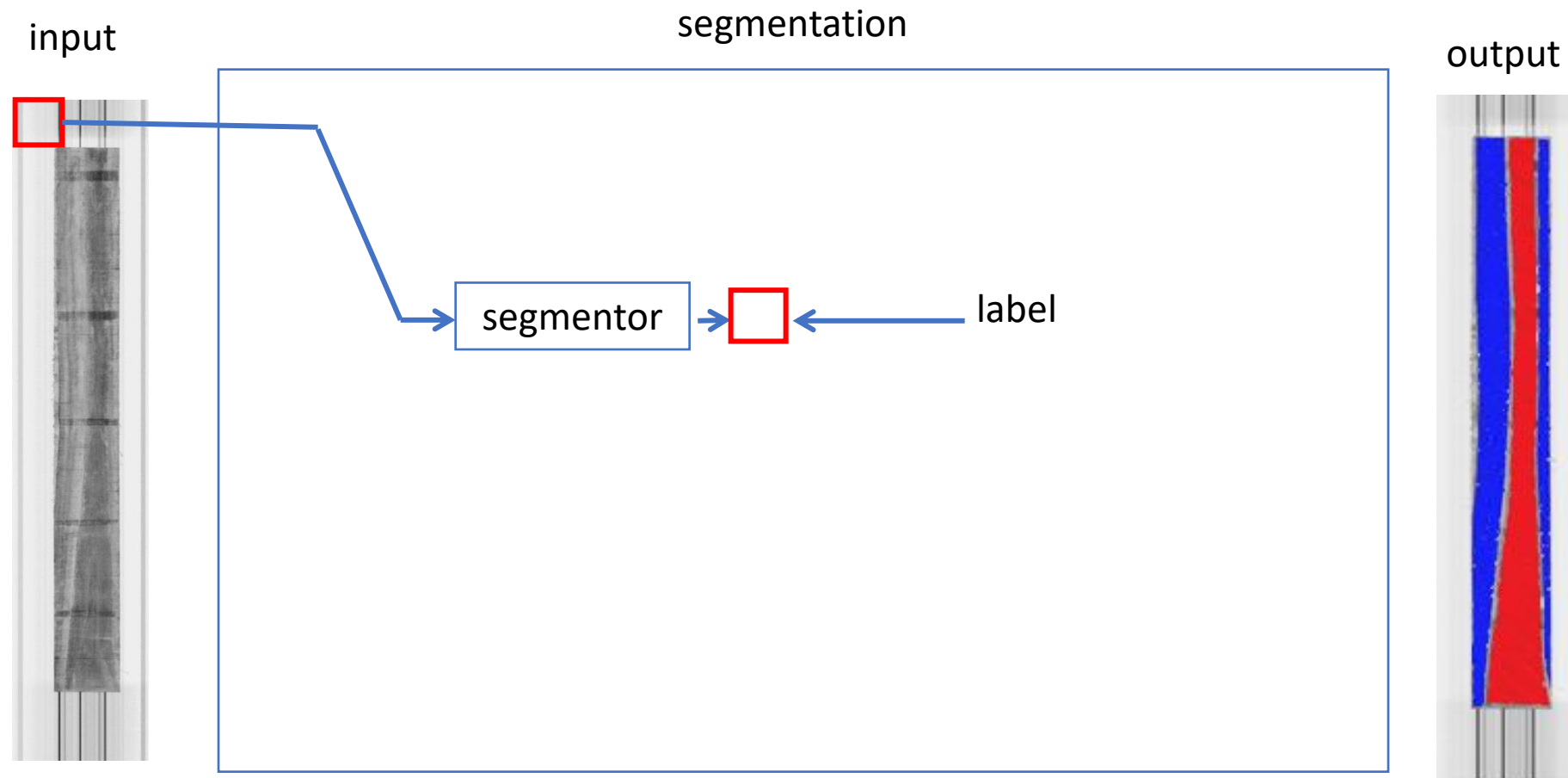
input



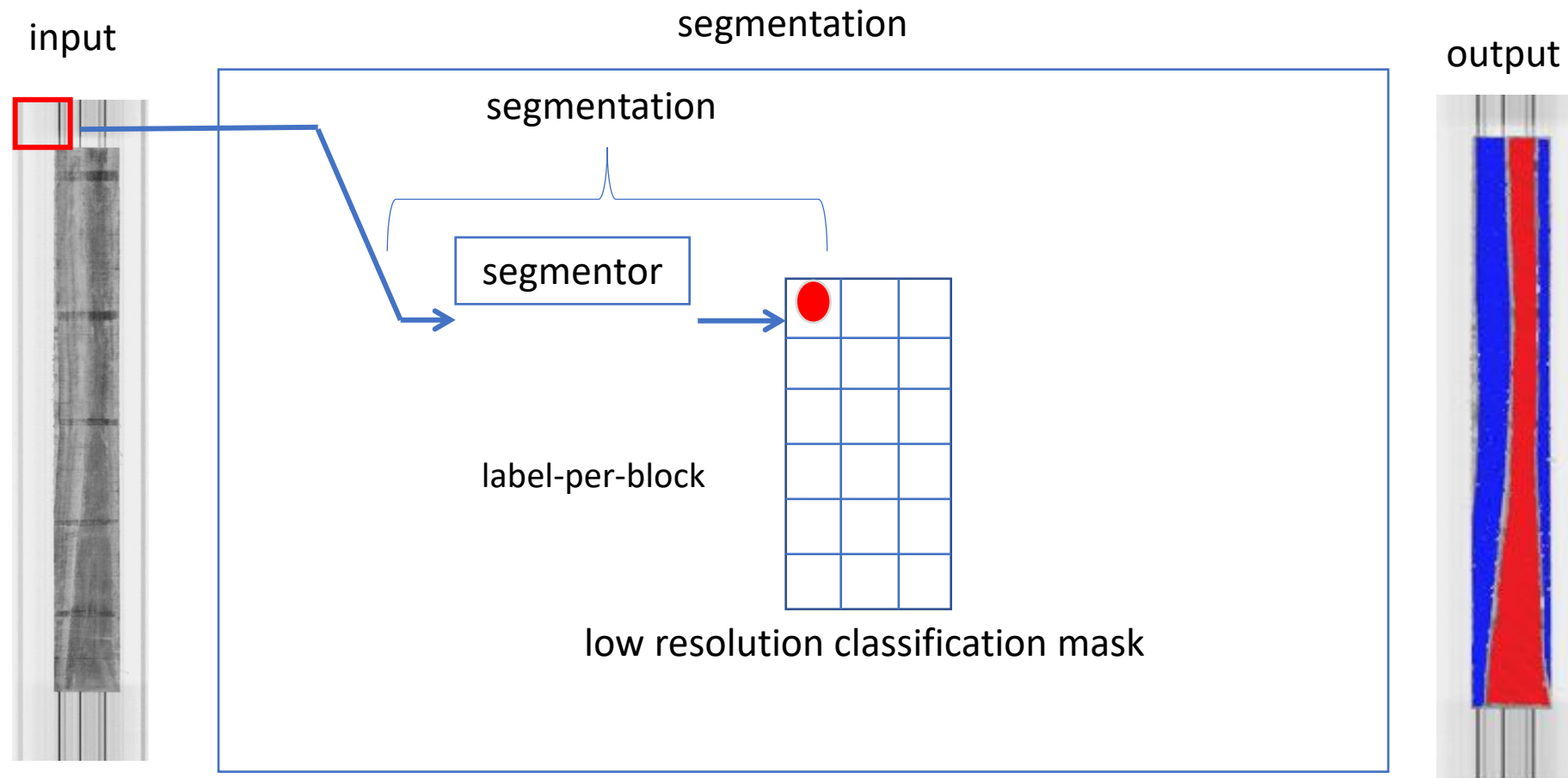
output



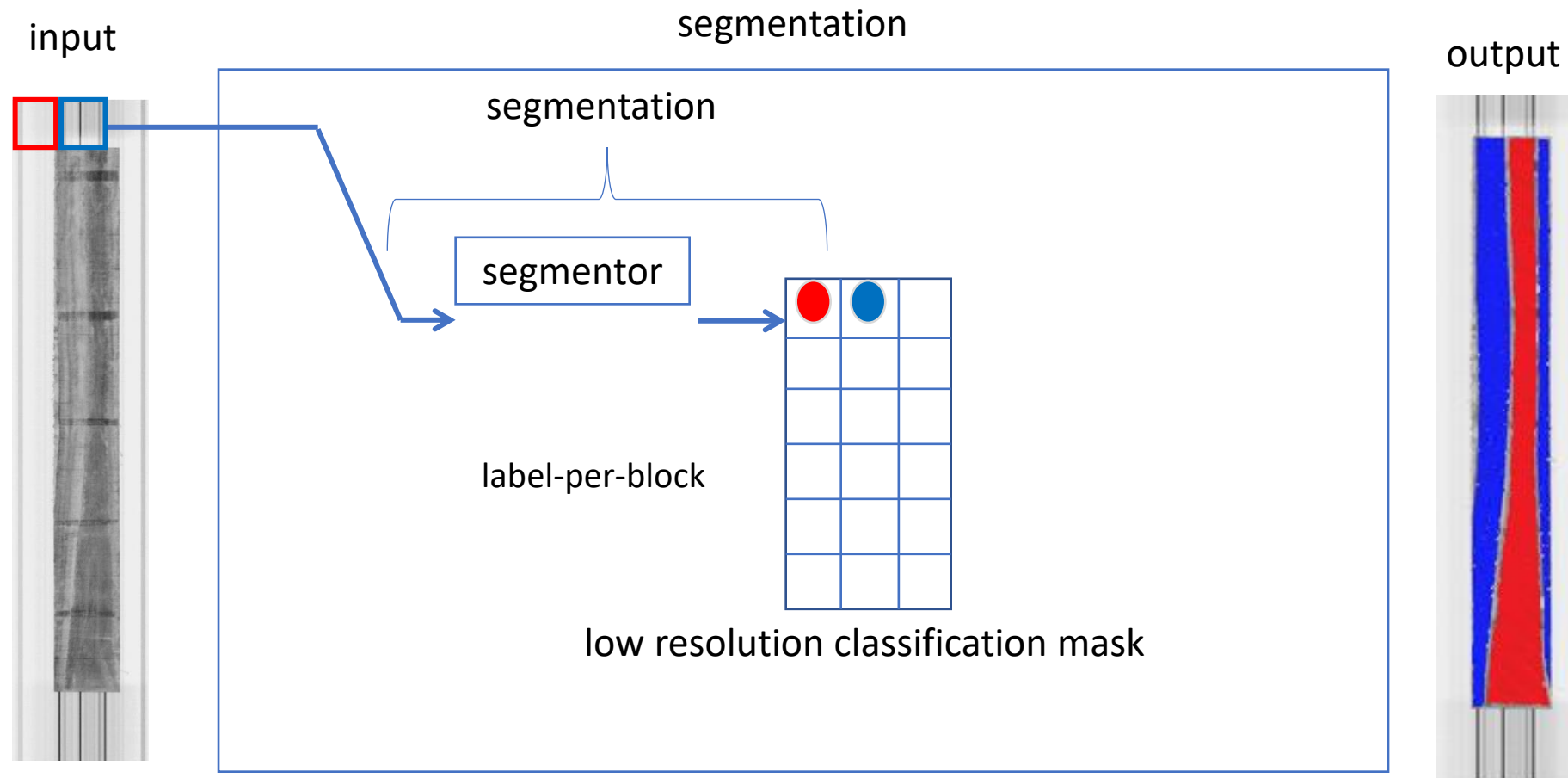
Proposed Framework



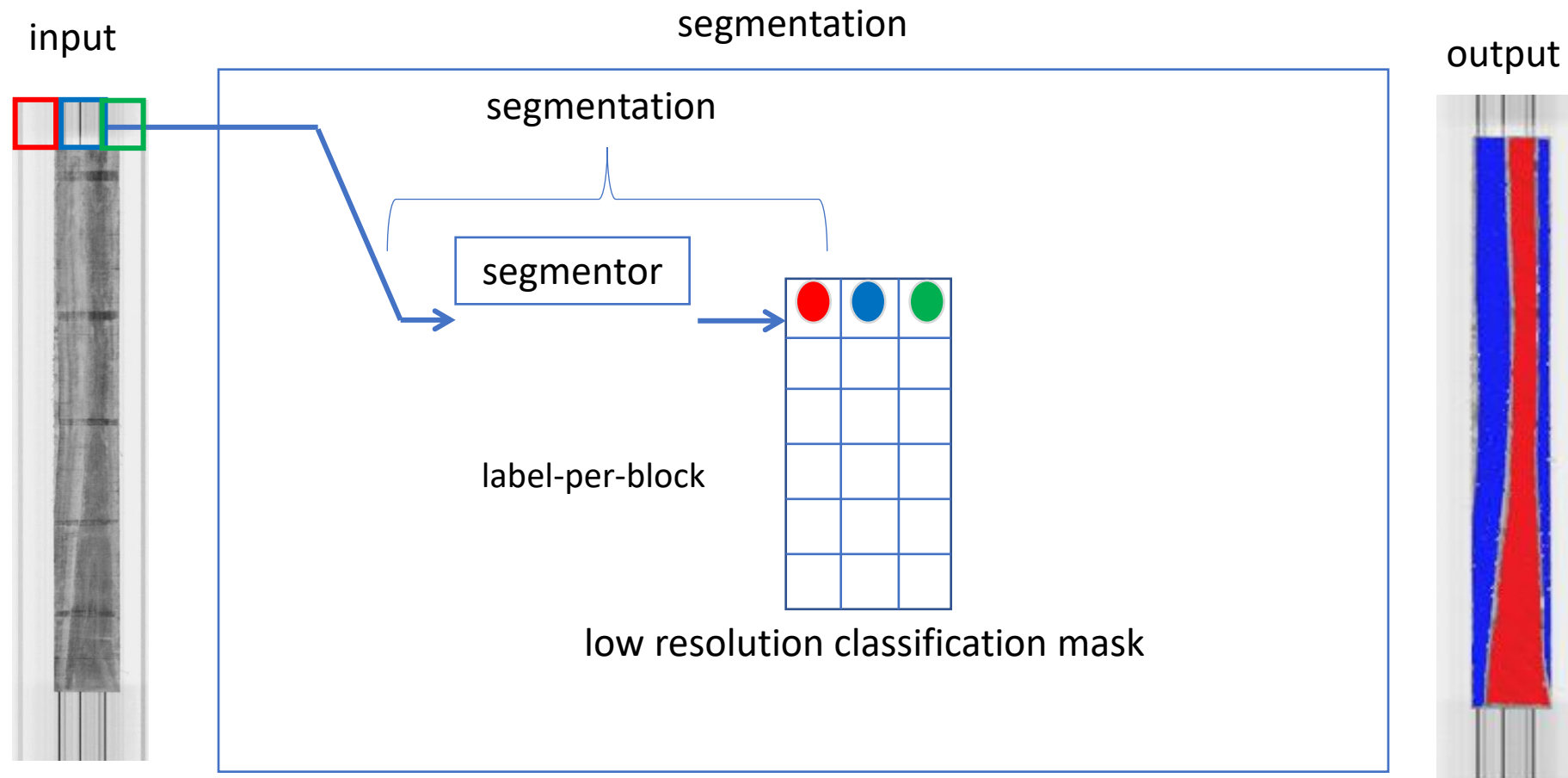
Proposed Framework



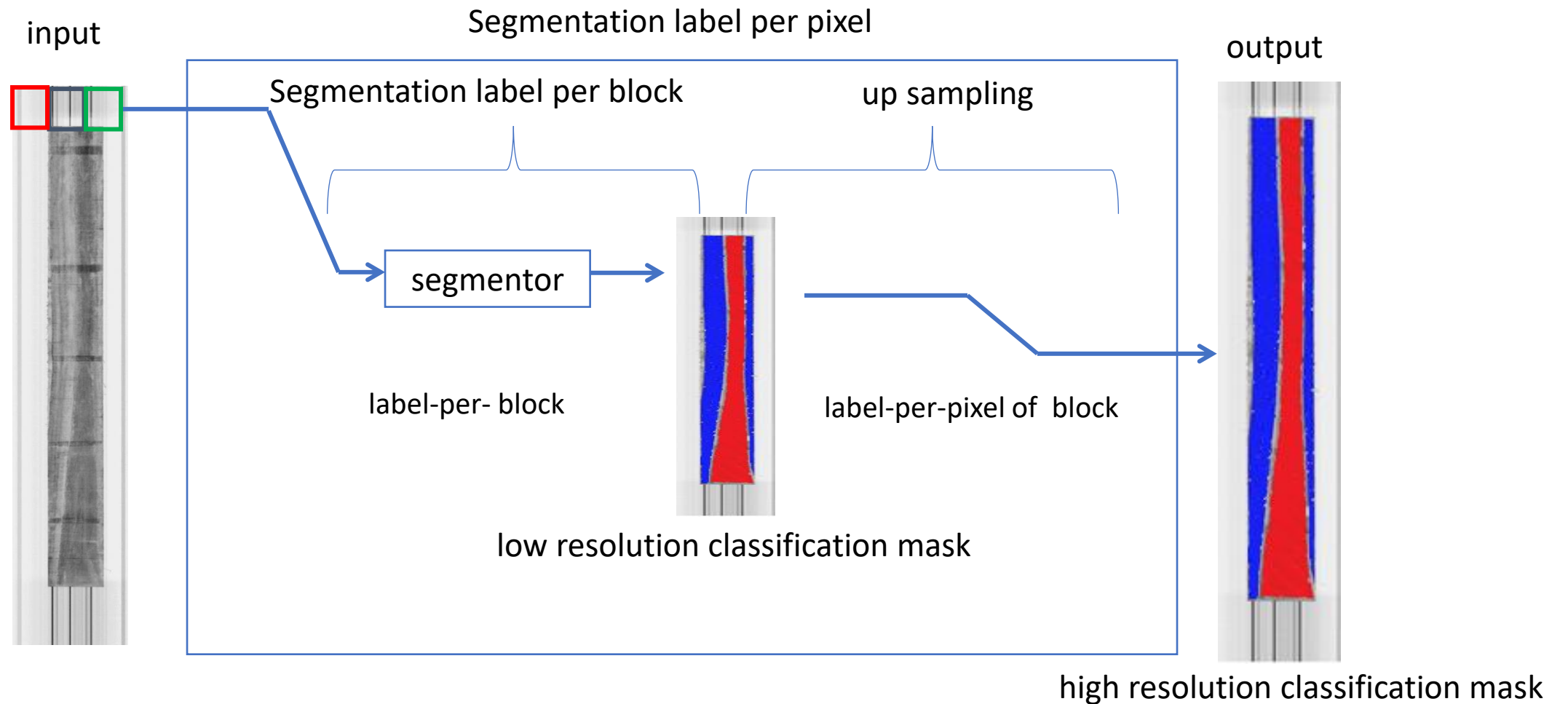
Proposed Framework



Proposed Framework

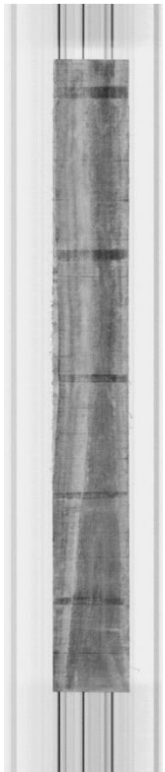


Proposed Framework



Proposed Framework

input



Which classifier used?

- Small size block classifier
- Used Convolutional Neural Network (CNN)
- cifar10Net available in MATLAB
- Simple classifier
- doesn't use a lot of resources

output



Architecture of classifier

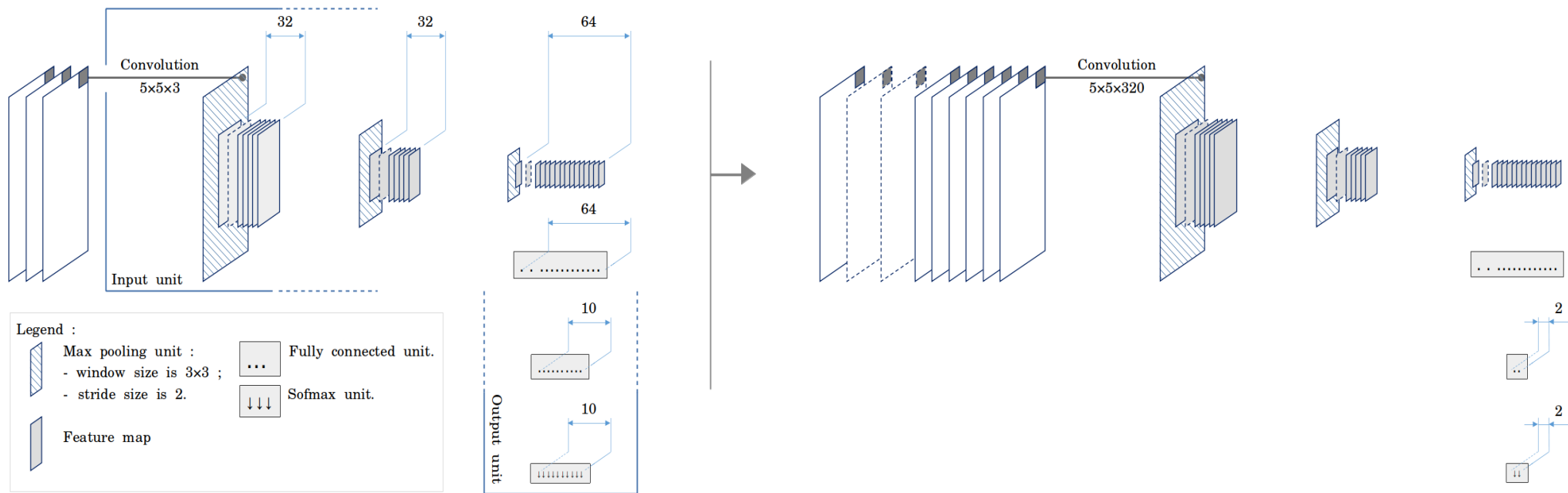


Fig. The architecture of cifar10Net (left) and the developed hyperspectral images classifier (right)

Architecture of classifier

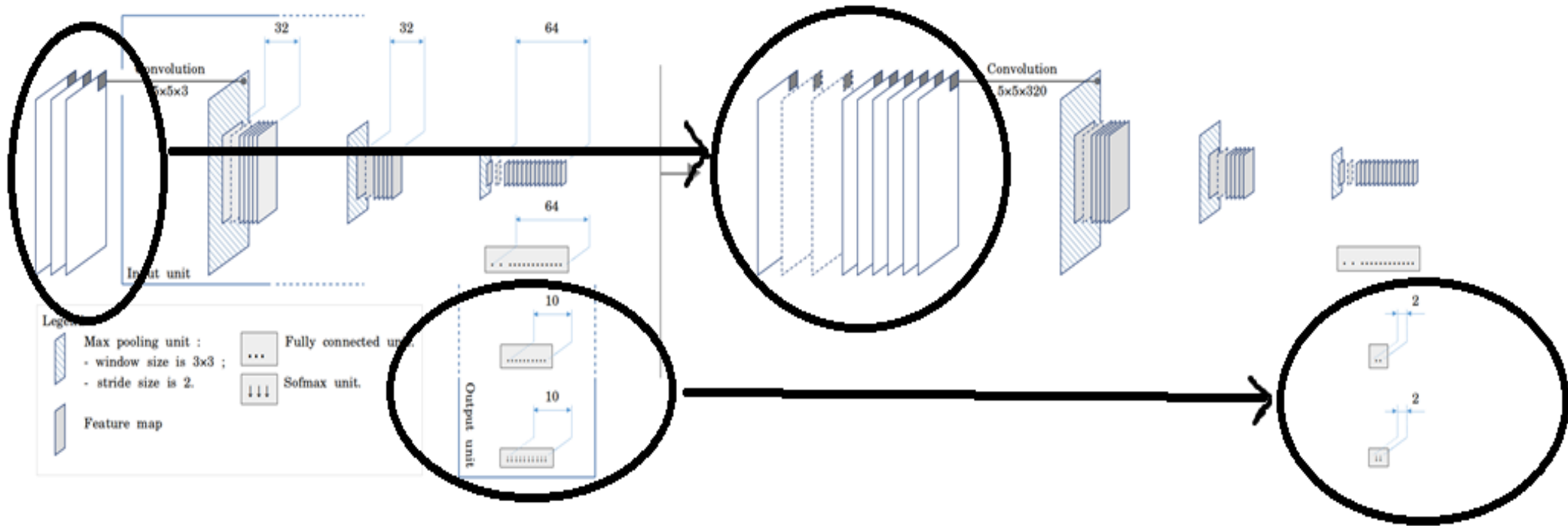


Fig. The architecture of cifar10Net (left) and the developed hyperspectral images classifier (right)

Architecture of classifier

- How to do?

Adapting Methodologies

- Modifying the output units of general image classifier
- Modifying the input units of general image classifier
- Tuning the hyper parameters of the modified layers
- Training

Training the output unit

The image shows the MATLAB R2019b environment. The Editor window displays a script for training a neural network. The Command Window shows the execution progress, including training on a single CPU and initializing input data normalization. A table displays the training results for the first two epochs.

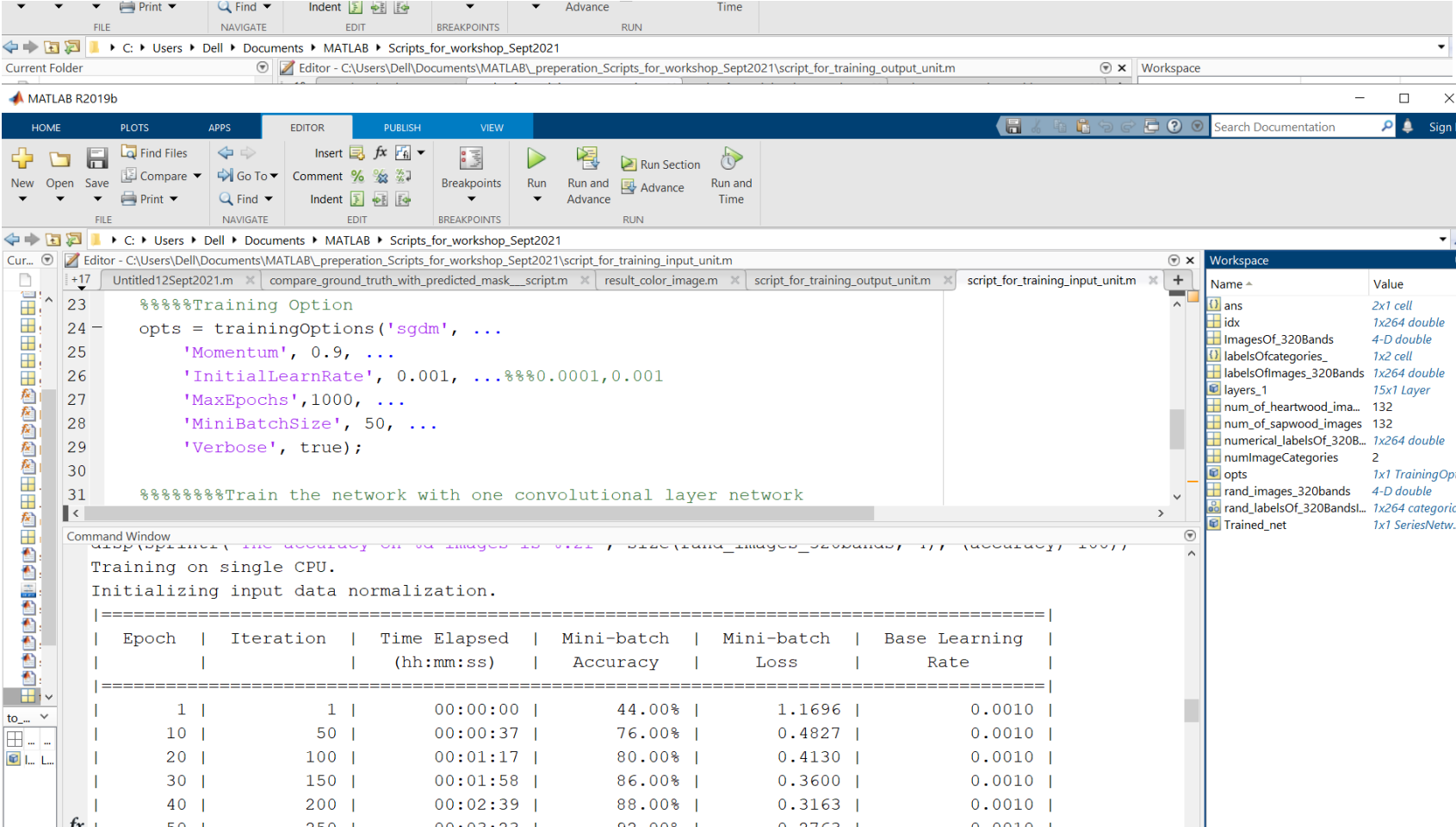
```
'Momentum', 0.9, ...
'InitialLearnRate', 0.0001, ...%%0.0001,0.001, two learning rate
'MaxEpochs', 1000, ...%%
'MiniBatchSize', 50, ...
'Verbose', true);

%%%%%% Train the network with two fullyconnected
```

Training on single CPU.
Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:00	56.00%	0.6869	1.0000e-04
1	50	00:00:02	58.00%	0.6822	1.0000e-04
1	100	00:00:04	52.00%	0.6955	1.0000e-04
1	150	00:00:07	46.00%	0.7092	1.0000e-04
1	200	00:00:08	60.00%	0.6764	1.0000e-04
1	250	00:00:10	42.00%	0.7178	1.0000e-04
1	300	00:00:12	48.00%	0.7028	1.0000e-04
2	350	00:00:15	40.00%	0.7208	1.0000e-04
2	400	00:00:16	40.00%	0.7192	1.0000e-04

Training the input unit



The MATLAB R2019b interface displays a script for training a convolutional layer network. The script is located in the Editor window, showing the following code:

```
23 %%%%Training Option
24 opts = trainingOptions('sgdm', ...
25     'Momentum', 0.9, ...
26     'InitialLearnRate', 0.001, ... %0.0001, 0.001
27     'MaxEpochs', 1000, ...
28     'MiniBatchSize', 50, ...
29     'Verbose', true);
30
31 %%%%%%Train the network with one convolutional layer network
```

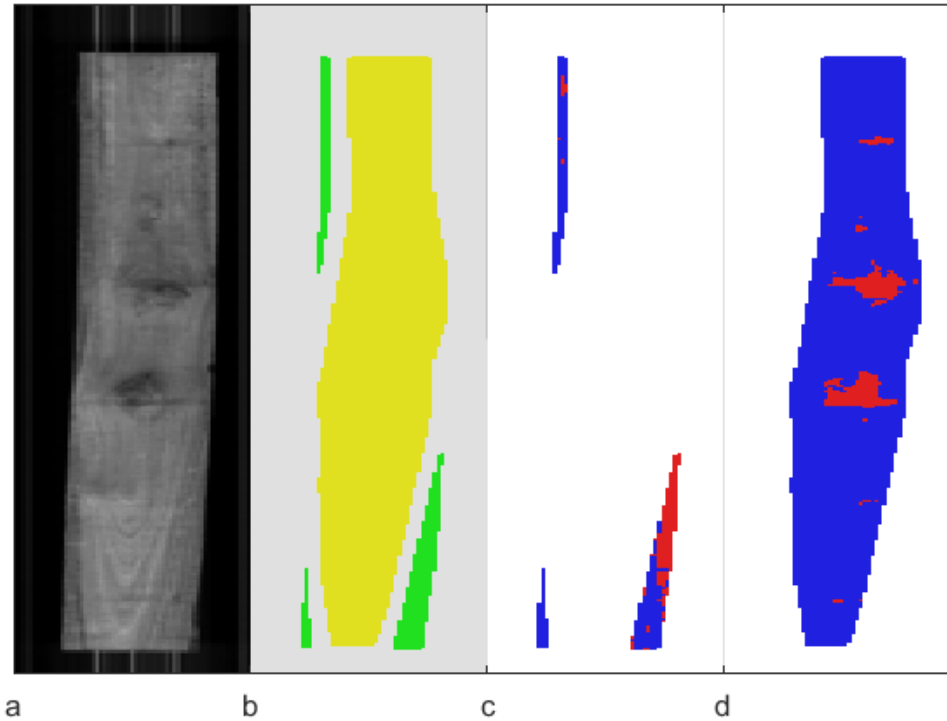
The Command Window shows the training progress table:

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:00	44.00%	1.1696	0.0010
10	50	00:00:37	76.00%	0.4827	0.0010
20	100	00:01:17	80.00%	0.4130	0.0010
30	150	00:01:58	86.00%	0.3600	0.0010
40	200	00:02:39	88.00%	0.3163	0.0010
50	250	00:03:23	89.00%	0.2763	0.0010

Single block classifier into multi block classifier –called segmentor

- Modifying the output units of single block classifier
- Modifying the input units of single block classifier
- Tuning the parameters of the modified layers
- No Training

Segmentation result



a to b: input image, b to c: ground truth, c to d: sapwood, d to e: heartwood
green is sapwood, yellow is heartwood
blue is correctly classified, red is uncorrectly classified

Expected Benefits

- Classification process is maturely than segmentation process in image processing area therefore, searching a classifier to adapt for using as an engine in segmentation framework is more easier than searching a segmentor.
- Training with pure data to a classifier is more simple than training with the whole image into a segmentor.

Thanks!

🏠 <https://h2i.inf.unibz.it/>
✉ h2i-fesr@googlegroups.com