TENSOR-BASED NONLINEAR CLASSIFIER FOR HIGH-ORDER DATA ANALYSIS

International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018)

K.Makantasis[1], A.Doulamis[2], N.Doulamis[2], A.Nikitakis[3], A.Voulodimos[2,4]

[1]KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus
[2]National Technical University of Athens, Athens, Greece
[3]Althexis Solutions Ltd, Nicosia, Cyprus
[4]Department of Informatics, Technological Educational Institute of Athens, Athens, Greece

Outline

High-order data

Typical ML approach

Linear tensor-based ML

Rank-1-FNonlinear tensor-based ML

Results

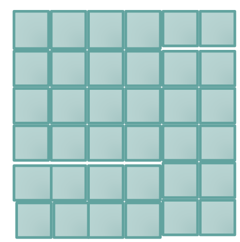
High-order data

The order of the data

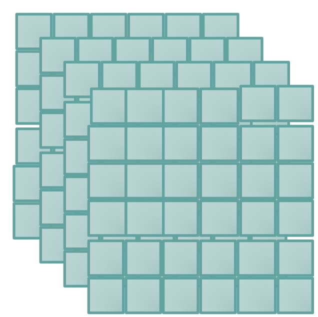
Zero order → scalars

First-order → vectors

Second-order → matrices



Third-order → 3d cubes

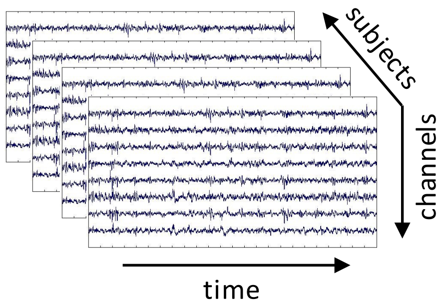


High-order data

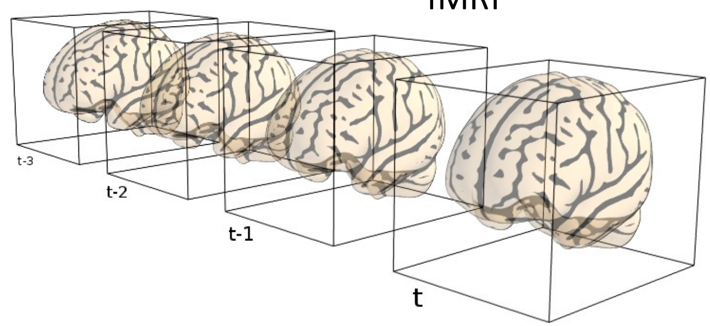
Advances in sensing technologies

Generation of large amounts of high-order data

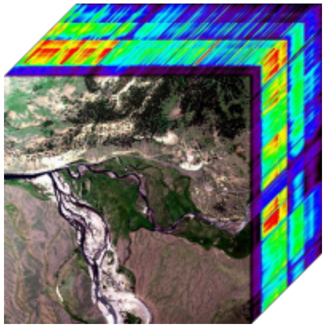
EEG



fMRI



Hyperspectral



Typical ML approach

Step

Vectorization of data

Features extraction

Learning

Typical ML approach

Step

Vectorization of data

Features extraction

Learning

Problems

Destroys the structural information

Information Loss

Overfitting

Typical ML approach

Step

Vectorization of data

Features extraction

Learning

Problems

Destroys the structural information

Information Loss

Overfitting

Example
Hyperspectral image
$(256 \times 256 \times 100)$

Linear regression

$$f(\boldsymbol{x}) = \boldsymbol{w}^{T}\boldsymbol{x}$$

parameters to estimate

- **w**$\in \Re^{100 \cdot 256^2}$
- $6{,}553{,}600$ parameters

Typical ML approach

Exceptions

Multi-linear regression models
- Exploit tensor algebra
- Low capacity
- Linear decision boundaries

Linear tensor-based ML

Tensor regression

$\boldsymbol{X} \in \Re^{p_1 \times p_2 \times \cdots \times p_d}$

$f(\boldsymbol{X}) = (\boldsymbol{w}_d \otimes \boldsymbol{w}_{d-1} \otimes \cdots \otimes \boldsymbol{w}_1)^T \, vec(\boldsymbol{X})$

The weights are in the form of a tensor

- Rank-1 canonical decomposition

$$\boldsymbol{W} = \boldsymbol{w_1} \circ \boldsymbol{w_1} \circ \cdots \boldsymbol{w_d}$$

Linear tensor-based ML

Tensor regression

$\boldsymbol{X} \in \Re^{p_1 \times p_2 \times \cdots \times p_d}$

$f(\boldsymbol{X}) = (\boldsymbol{w}_d \otimes \boldsymbol{w}_{d-1} \otimes \cdots \otimes \boldsymbol{w}_1)^T \, vec(\boldsymbol{X})$

The weights are in the form of a tensor

- Rank-1 canonical decomposition

$$\boldsymbol{W} = \boldsymbol{w_1} \circ \boldsymbol{w_1} \circ \cdots \boldsymbol{w_d}$$

Reduces the number of parameters - smaller number of labelled examples

Linear tensor-based ML

Tensor regression

$\boldsymbol{X} \in \Re^{p_1 \times p_2 \times \cdots \times p_d}$

$f(\boldsymbol{X}) = (\boldsymbol{w}_d \otimes \boldsymbol{w}_{d-1} \otimes \cdots \otimes \boldsymbol{w}_1)^T \, vec(\boldsymbol{X})$

The weights are in the form of a tensor

- Rank-1 canonical decomposition

$$\boldsymbol{W} = \boldsymbol{w_1} \circ \boldsymbol{w_1} \circ \cdots \boldsymbol{w_d}$$

Reduces the number of parameters - smaller number of labelled examples

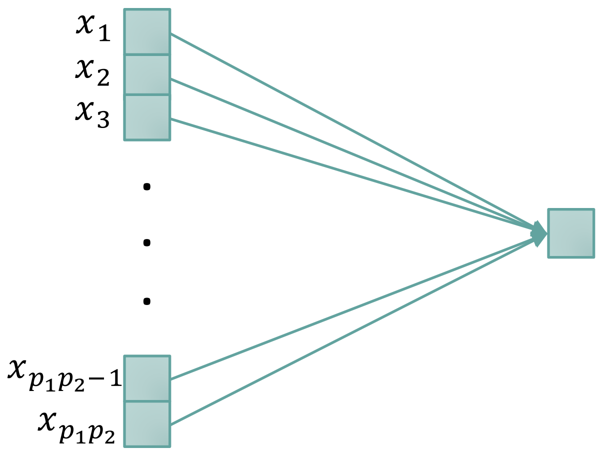
Drawback

- Linear decision boundaries in input space

Linear tensor-based ML

Consider $\boldsymbol{X} \in \Re^{p_1 \times p_2}$

Linear regression
$\boldsymbol{w} \in \Re^{p_1 \cdot p_2}$

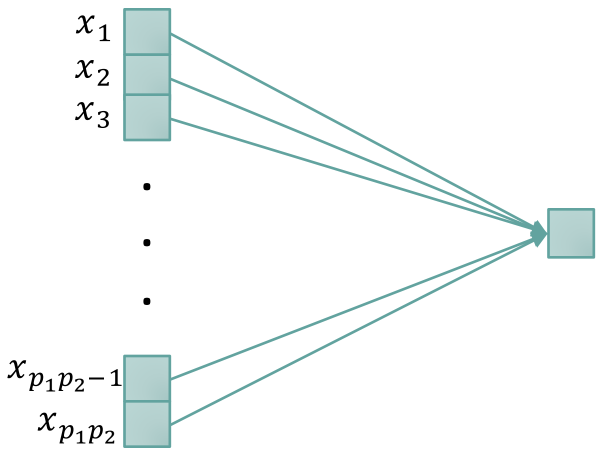


Linear tensor-based ML

Consider $\boldsymbol{X} \in \Re^{p_1 \times p_2}$

Bilinear regression

$\boldsymbol{w} \in \Re^{p_1 \cdot p_2} : \boldsymbol{w} = vec(\boldsymbol{w}_1^T \cdot \boldsymbol{w}_2)$

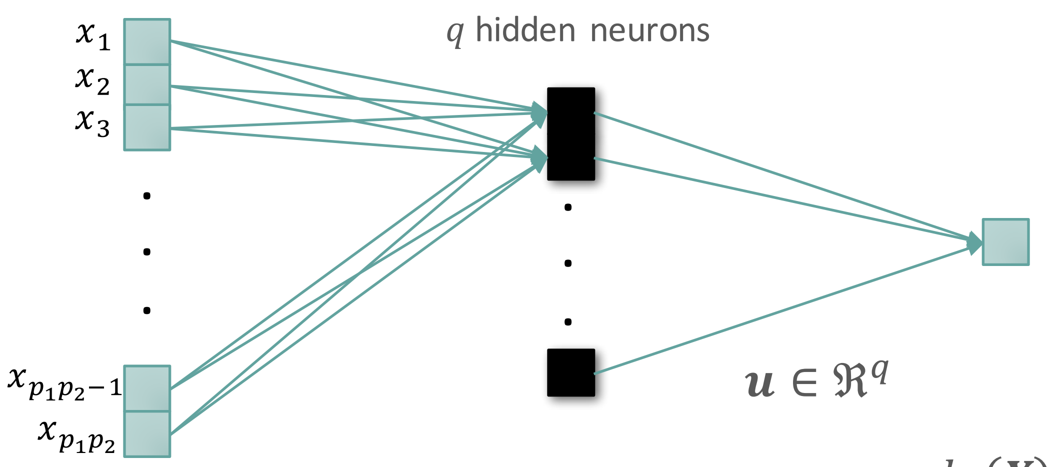


Rank-1-FNN

Consider $\boldsymbol{X} \in \Re^{p_1 \times p_2}$

Rank-1 FNN

$$\boldsymbol{w}_i \in \Re^{p_1 \cdot p_2} : \boldsymbol{w}_i = vec(\boldsymbol{w}_{1,i}^T \cdot \boldsymbol{w}_{2,i})$$



$$h_i(\boldsymbol{X}) = \sigma\left(\left(\boldsymbol{w}_{2,i} \otimes \boldsymbol{w}_{1,i}\right)^T vec(\boldsymbol{X})\right)$$

$$f(X) = \boldsymbol{u}^{\mathrm{T}} g\left(\boldsymbol{h}(\boldsymbol{X})\right)$$

Rank-1-FNN

1) Observation

$$\left\langle \boldsymbol{w}_{d,i} \otimes \boldsymbol{w}_{d-1,i} \otimes \cdots \otimes \boldsymbol{w}_{2,i} \otimes \boldsymbol{w}_{1,i}, \boldsymbol{X} \right\rangle =$$

$$\left\langle \boldsymbol{w}_{l,i}, \boldsymbol{X}_{(l)}(\boldsymbol{w}_{d,i} \otimes \cdots \otimes \boldsymbol{w}_{l+1,i} \otimes \boldsymbol{w}_{l-1,i} \otimes \cdots \otimes \boldsymbol{w}_{2,i} \otimes \boldsymbol{w}_{1,i}) \right\rangle$$

2) Denote as

$$\tau_{\neq l,i} = \boldsymbol{X}_{(l)}(\boldsymbol{w}_{d,i} \otimes \cdots \otimes \boldsymbol{w}_{l+1,i} \otimes \boldsymbol{w}_{l-1,i} \otimes \cdots \otimes \boldsymbol{w}_{2,i} \otimes \boldsymbol{w}_{1,i})$$

then the estimation of $\boldsymbol{w}_{l,i}$ corresponds to training a typical FNN with input $\tau_{\neq l,i}$.

Rank-1-FNN

Training algorithm

1. Set iteration index $n \rightarrow 0$
2. Initialize randomly the weights $\boldsymbol{w}_{l,i}(n) \in \Re^{p_l}$ and $\boldsymbol{u}(n) \in \Re^{q}$
3. Repeat

   for $l = 1, \dots, d$

   Compute $\tau_{\neq l,i}$ using $\boldsymbol{w}_{d,i}(n)$ for $d > l$ and $\boldsymbol{w}_{d,i}(n+1)$ for $d < l$

   Estimate weights $\boldsymbol{w}_{l,i}(n+1)$

   Estimate weights $\boldsymbol{u}(n+1)$

   Set $n \rightarrow n+1$

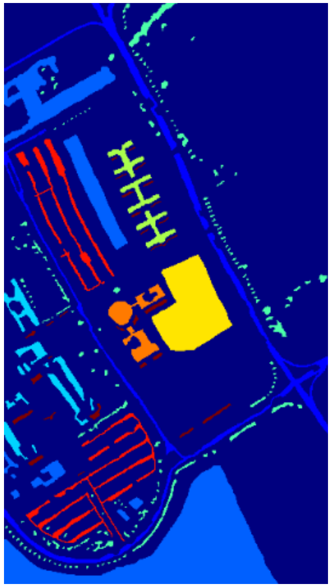
   Until termination criteria are met

Results

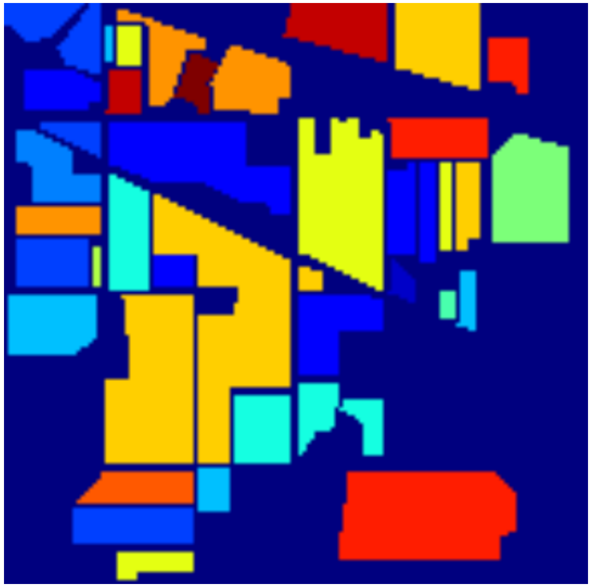
Two public available hyperspectral datasets

- Pavia University – 9 classes, 103 spectral bands
- Indian Pines – 16 classes, 224 spectral bands

Pavia Univ.

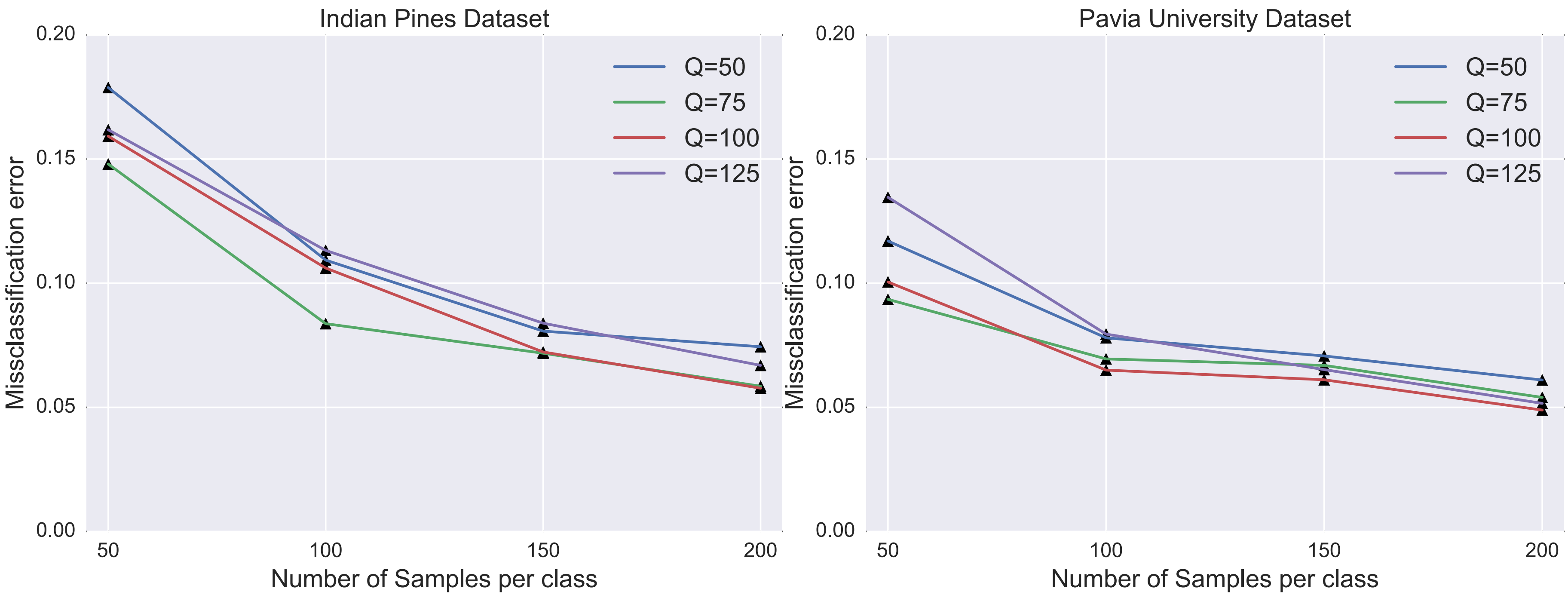


Indian Pines



Results

Miss-classification error vs complexity

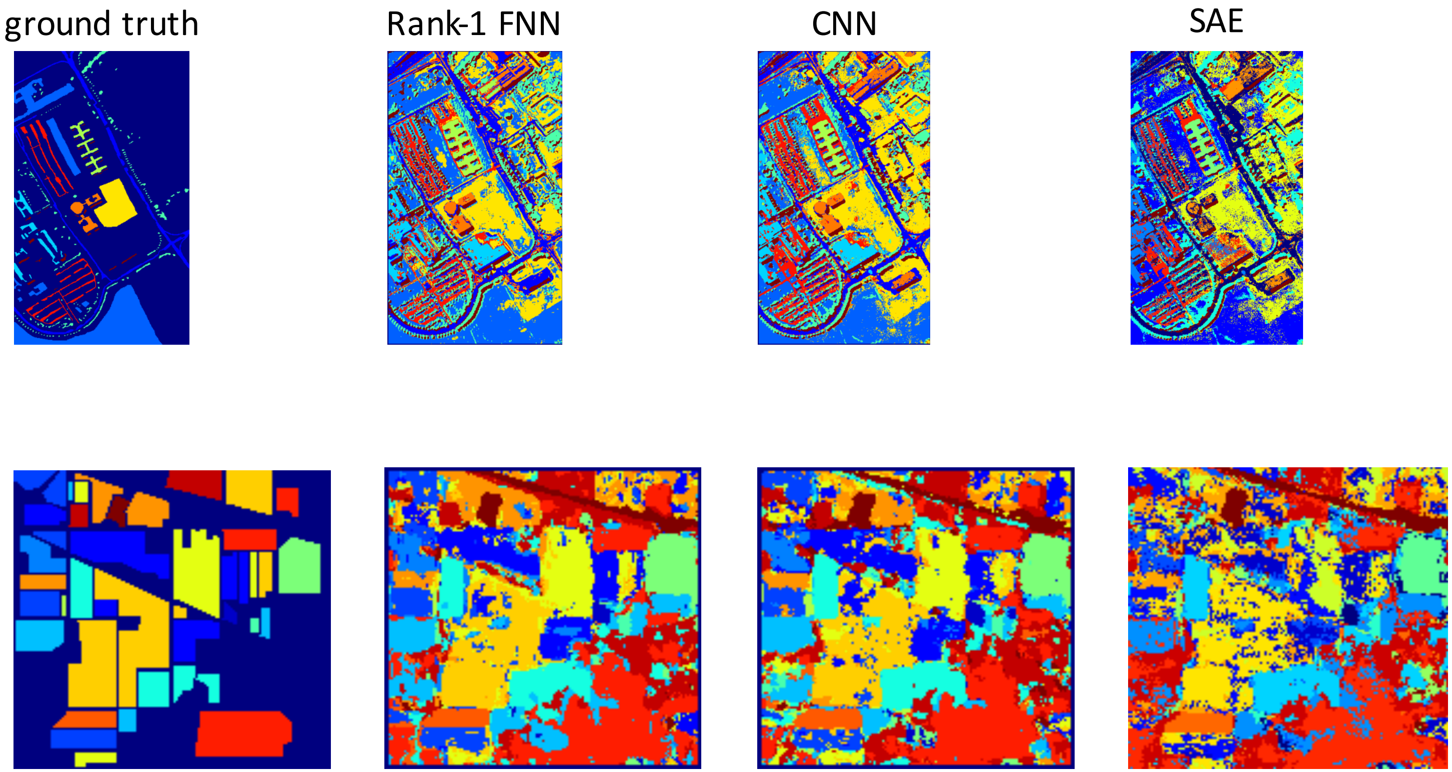


Results

Pavia University				
Samples per class	50	100	150	200
Rank-1 FNN	**89.95**	**93.50**	93.89	95.11
FNN	67.79	76.53	78.48	82.59
RBF-SVM	86.98	88.99	89.86	91.82
SAE	86.54	91.90	92.38	93.89
CNN	88.89	92.74	**94.68**	**95.89**
Indian Pines				
Samples per class	50	100	150	200
Rank-1 FNN	**85.20**	**91.63**	**92.82**	94.15
FNN	73.88	81.10	84.14	85.86
RBF-SVM	73.18	77.86	82.11	84.99
SAE	65.51	70.66	74.03	79.49
CNN	82.43	85.48	92.28	**94.81**

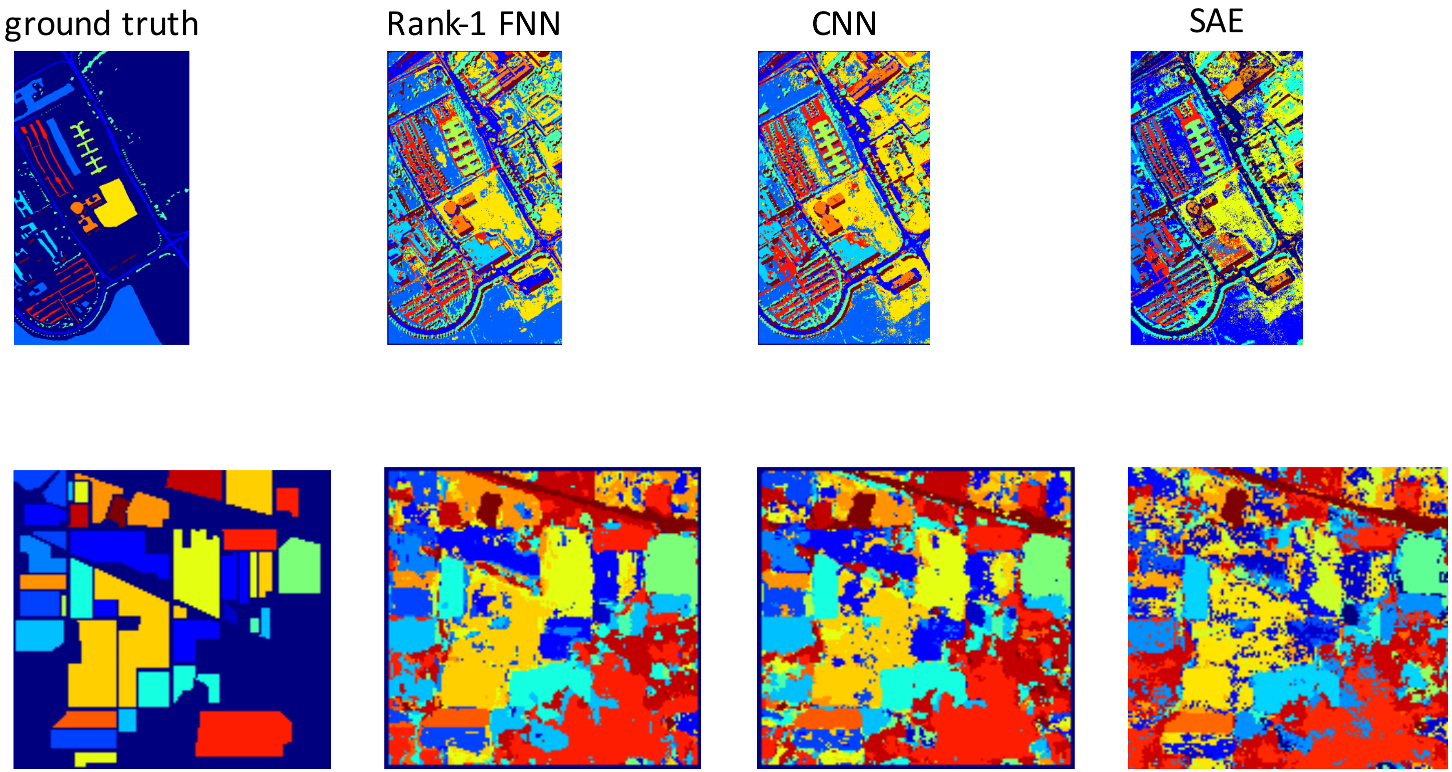
Results

Visualization of the classification – 50 samples per class



Results

Visualization of the classification – 50 samples per class



Thanks

for

your attention