Multi-Scanning Based Recurrent Neural Network for Hyperspectral Image Classification

Presenter: Weilian Zhou
From Kamata Lab.
Outline

• **Introduction**
  ① Hyperspectral image
  ② Two intrinsic properties
  ③ HSI classification

• **Existing works**
  ① Overview
  ② Classification results of examples

• **Observations**

• **Motivation**
  ① Motivation-1
  ② Motivation-2

• **Proposed method**
  ① Multi-scanning strategy
  ② Spectral-spatial structure of HSI for RNN
  ③ Whole framework of the method

• **Experiments**

• **Comparison**

• **Conclusion**
Introduction (Hyperspectral Image (HSI))

- **What is HSI?:** Hyperspectral sensors portrays the surface of the earth using hundreds of spectral bands in electromagnetic spectrum. The generated image is called hyperspectral image (HSI) considered as a 3-D cube.

- **Spectral dimension (1D):** Capture the spectral signature of each material in different bands. One pixel is a $N$-dimension feature vector representing the value of each band. ($N$ denotes the number of bands)

- **Spatial dimension (2D):** Record the locations of objects.

https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html
**Introduction** (Two intrinsic properties in HSI)

① The pixels that are close in the feature space (spectral dimension) are highly possible to have the same label. (A, B and D)

② The pixels that are spatially nearby (spatial dimension) have the large possibilities to have the same label, but exception exists. (A and B, E and D)

It’s important to consider
- **Spatial** (neighboring pixels)
- **Spectral** (hyperspectral signatures)
information together to determine pixel’s label.
Introduction (Hyperspectral Image Classification)

- **Target:** To assign a class label to each pixel.
- **Applications:** Environmental detection, urban planning and resource exploring, etc.
- **Recent situation:** Deep learning with this task got outstanding performances.

Why we require these applications?
1. Visualizing the object distributions of the ground surface.
2. Exploring what we want to discover on the ground.

Fig. 4 Main procedure

The better classification results will help improve the quality of these applications.

HSI classification is important!
## Existing work (Overview)

- Recent HSI classification works with deep learning can be divided into three categories. [12]

### 1. (1D) Spectral-based  
### 2. (2D) Spatial-based  
### 3. Spectral and Spatial-based

<table>
<thead>
<tr>
<th>Categories</th>
<th>Existing Works</th>
<th>Methods</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Spectral</td>
<td>Hu et al.[1] (J. Sensors)</td>
<td>1-D CNN for analyzing spectral signature</td>
<td>Just input the raw spectral signatures into models;</td>
</tr>
<tr>
<td></td>
<td>Mou et al.[2] (IGARSS)</td>
<td>Deep RNN for finding the relationship between spectral</td>
<td>None of them consider the ‘redundancy’ between adjacent bands and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information</td>
<td>complementarity between non-adjacent bands.</td>
</tr>
<tr>
<td>2. Spatial</td>
<td>Li et al.[7] (IGARSS)</td>
<td>Fully CNN with three principle components of PCA to</td>
<td>None of them consider the complete spectral information. And neighboring</td>
</tr>
<tr>
<td></td>
<td></td>
<td>analyze</td>
<td>inferring pixels may exist.</td>
</tr>
<tr>
<td></td>
<td>Feng et al.[5] (Neurocomputing)</td>
<td>First PCA component as input for LSTM</td>
<td></td>
</tr>
<tr>
<td>3. Spectral-Spatial</td>
<td>Leng et al.[8] (ICTAI)</td>
<td>A novel Cube-CNN-SVM. Target pixel with spectral information and its neighbors are organized into 3D-CNN</td>
<td>3D-CNN is suitable for the data structure of the HSI. Though most of the local space areas are homogeneous in an HSI, casually selecting all the pixels’ spectral information in the local area (patch) incurs multiplication and addition from different classes during the training.</td>
</tr>
<tr>
<td></td>
<td>Li et al.[9] (Remote sensing)</td>
<td>3D-CNN without relying on any pre-processing or post-processing to extract the in-depth combined features.</td>
<td></td>
</tr>
</tbody>
</table>
Existing work (classification results on three categories)

- The classification results on **Indian Pines dataset**. (145*145*200)

<table>
<thead>
<tr>
<th>Spectral-based</th>
<th>Spatial-based</th>
<th>Spectral-Spatial-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hu[1]: 75.882%</td>
<td>Mou[2]: 76.175%</td>
<td>Liu[13]: 80.804%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Li[9]: 82.642%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leng[8]: 90.488%</td>
</tr>
</tbody>
</table>

**Drawback:**
① Noise-like misclassification everywhere.
② One pixel’s neighboring pixels are mess.

**Merit:**
① Less noise-like misclassification;
② Neighboring pixels are considered.
③ More accurate and smooth in flat region

**Drawback:**
① Object-like misclassifications.
② Misclassifications mainly on boundaries.

**Drawback:**
① Object-like misclassification everywhere.
② One pixel’s neighboring pixels are mess.

**Merit:**
① Accuracy is higher.
② Misclassification regions are smaller and smoother than spatial-based method.
③ Less noise-like misclassification than spectral-based method.
④ The boundaries are cleaner.

**Drawback:**
① Computational burden with long training time.
② Still has noise-like and object-like misclassification.
Existing work (classification results on three categories)

- The classification results on **Pavia Uni. dataset** (610*340*103)

<table>
<thead>
<tr>
<th>Spectral-based</th>
<th>Spatial-based</th>
<th>Spectral-Spatial-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hu[1]: 90.741%</td>
<td>Liu[13]: 92.744%</td>
<td>Li[9]: 96.396%</td>
</tr>
<tr>
<td>Mou[2]: 91.671%</td>
<td></td>
<td>Leng[8]: 95.682%</td>
</tr>
</tbody>
</table>

It is better to combine the spatial and spectral information together.

but
**Observation** (spatial information)

- **Assumption**: in spatial domain, adjacent pixels are highly likely to have same labels.
- **Real situation**: at edge of each land-cover area, cropped HSI cubes usually contain several pixels whose labels are different from the center pixel. *(interfering pixels)*
- **Problem**: the pixels in the local patch may have different appearance variations, and this probably affects the final classification accuracy.

**Patch**: The main idea is to associate each pixel within an adaptive neighborhoods (to combine the spatial information)
Observation (good examples in a patch)

- Real situation in experiments about cropped patch. (good examples)
- Pixels in this patch will have same spectral signatures distribution.
Observation (bad examples in a patch)

- Real situation in experiments about cropped patch. (bad examples)
- Pixels in this patch will have big variations on spectral signatures. (hard to decide a class)
Motivation-1 (complex situation in a patch)

- **Problem-1**: directly use cropped patches (HSI cubes) as input into network (such as CNN), interfering pixels will give bad influence with the extraction of features.

![Diagram of CNN-based HSI classification methods](image)

**Ground truth of input patch**

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>3</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Mix up**

**Kernel**

**Convoluted feature**

**Idea**: to find a method to deal with the local pixels one by one individually instead of convolving them together.

**Recurrent neural network (RNN)** is good at dealing with sequential data with time steps.
Motivation

- Recurrent Neural Network (RNN)
- Learning the relationship between inputs step by step.

\[
\begin{align*}
    h_t & \leftarrow y_t \\
    h_{t+1} & \leftarrow U_h h_t + W_y y_{t+1} \\
    h_{t+2} & \leftarrow U_h h_{t+1} + W_y y_{t+2} \\
    h_{t+3} & \leftarrow U_h h_{t+2} + W_y y_{t+3} \\
    h_{t+4} & \leftarrow U_h h_{t+3} + W_y y_{t+4}
\end{align*}
\]

- RNN for a patch. Transfer it into a sequence.
- Each pixel or step is a feature vector.

3 × 3 patch → Flatten → 1 × 9 sequence
Motivation-2 (limited labeled pixels)

- **Problem-2**: In the HSI, unbalanced the number of pixels are severe
- In the real HSI, labeled pixels are taken in a limited set of conditions. But the target may exist in a variety of conditions, such as different location, scale, brightness etc.

![Land-cover classes and the number of pixels on the Indian Pines dataset](image-url)

<table>
<thead>
<tr>
<th>classes</th>
<th>Test(90%)</th>
<th>Train(10%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfalfa</td>
<td>41</td>
<td>5</td>
</tr>
<tr>
<td>Corn-notill</td>
<td>1285</td>
<td>143</td>
</tr>
<tr>
<td>Corn-mintill</td>
<td>747</td>
<td>83</td>
</tr>
<tr>
<td>Corn</td>
<td>213</td>
<td>24</td>
</tr>
<tr>
<td>Grass-pasture</td>
<td>435</td>
<td>48</td>
</tr>
<tr>
<td>Grass-tree</td>
<td>657</td>
<td>73</td>
</tr>
<tr>
<td>Grass-pasture-mowed</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>Hay-windrowed</td>
<td>430</td>
<td>48</td>
</tr>
<tr>
<td>Oats</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>Soybean-notill</td>
<td>875</td>
<td>97</td>
</tr>
<tr>
<td>Soybean-mintill</td>
<td>2209</td>
<td>246</td>
</tr>
<tr>
<td>Soybean-clean</td>
<td>534</td>
<td>59</td>
</tr>
<tr>
<td>Wheat</td>
<td>185</td>
<td>20</td>
</tr>
<tr>
<td>Wood</td>
<td>1139</td>
<td>126</td>
</tr>
<tr>
<td>Building-Grass-Tree</td>
<td>347</td>
<td>39</td>
</tr>
<tr>
<td>Stone-Stele-Tower</td>
<td>84</td>
<td>9</td>
</tr>
</tbody>
</table>
### Motivation-2

- Classification results on existing works with each class. (Indian Pines dataset)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfalfa</td>
<td>57.61%</td>
<td>78.31%</td>
<td>42.33%</td>
<td>57.12%</td>
<td>85.22%</td>
</tr>
<tr>
<td>Corn-notill</td>
<td>72.23%</td>
<td>74.90%</td>
<td>78.41%</td>
<td>77.98%</td>
<td>88.51%</td>
</tr>
<tr>
<td>Corn-mintill</td>
<td>54.23%</td>
<td>61.02%</td>
<td>65.63%</td>
<td>69.45%</td>
<td>81.71%</td>
</tr>
<tr>
<td>Corn</td>
<td>57.92%</td>
<td>46.92%</td>
<td>63.66%</td>
<td>74.11%</td>
<td>84.70%</td>
</tr>
<tr>
<td>Grass-pasture</td>
<td>89.51%</td>
<td>88.66%</td>
<td>85.70%</td>
<td>89.46%</td>
<td>96.00%</td>
</tr>
<tr>
<td>Grass-trees</td>
<td>93.77%</td>
<td>93.41%</td>
<td>96.11%</td>
<td>97.48%</td>
<td>98.23%</td>
</tr>
<tr>
<td>Grass-pasture-mowed</td>
<td>61.54%</td>
<td>71.88%</td>
<td>57.12%</td>
<td>86.41%</td>
<td>93.77%</td>
</tr>
<tr>
<td>Hay-windrowed</td>
<td>95.49%</td>
<td>96.31%</td>
<td>96.33%</td>
<td>96.33%</td>
<td>98.61%</td>
</tr>
<tr>
<td>Oats</td>
<td>42.91%</td>
<td>45.22%</td>
<td>10.52%</td>
<td>43.59%</td>
<td>75.72%</td>
</tr>
<tr>
<td>Soybean-notill</td>
<td>61.96%</td>
<td>67.81%</td>
<td>72.23%</td>
<td>76.88%</td>
<td>85.00%</td>
</tr>
<tr>
<td>Soybean-mintill</td>
<td>73.00%</td>
<td>73.15%</td>
<td>82.69%</td>
<td>81.70%</td>
<td>90.01%</td>
</tr>
<tr>
<td>Soybean-clean</td>
<td>72.38%</td>
<td>76.99%</td>
<td>71.23%</td>
<td>64.65%</td>
<td>81.76%</td>
</tr>
<tr>
<td>Wheat</td>
<td>98.11%</td>
<td>96.54%</td>
<td>99.26%</td>
<td>99.58%</td>
<td>97.91%</td>
</tr>
<tr>
<td>Woods</td>
<td>93.45%</td>
<td>92.51%</td>
<td>96.87%</td>
<td>95.26%</td>
<td>95.80%</td>
</tr>
<tr>
<td>Building-Grass-Tree</td>
<td>68.66%</td>
<td>66.53%</td>
<td>66.11%</td>
<td>75.74%</td>
<td>81.99%</td>
</tr>
<tr>
<td>Stone-Steel-Tower</td>
<td>87.32%</td>
<td>85.97%</td>
<td>91.66%</td>
<td>100%</td>
<td>96.96%</td>
</tr>
<tr>
<td>Overall accuracy</td>
<td>76.92%</td>
<td>78.77%</td>
<td>81.99%</td>
<td>83.51%</td>
<td>90.32%</td>
</tr>
</tbody>
</table>

- red parts: accuracy lower than 80%
- green parts: overall accuracy
Motivation

- **Idea**: using data augmentation methods for these limited labeled pixels.

- **Data augmentation**: To get more data. Popular data augmentation methods: flip, rotation, scale, crop, translation, reflection, mirror, noise, radiation, GAN.

![Motivation Diagram]

Original data

Augmented data

Network

Class: dog

Augmented data

Inspired

Original patch (5*5)
label: Corn-notill

ground truth

Augmented data

original

mirroring

corner

rotation

flipping

number means class.
Proposed method (Multi-scanning strategy)

- In order to feed a local patch into RNN, we must transfer the patch into the sequence. But there's no fixed scanning direction exists.
- Take 3*3 patch (9 pixels) as an example. Start from first pixel.
Proposed method (Multi-scanning strategy)

- After augmenting (mirroring) the patch.

<table>
<thead>
<tr>
<th></th>
<th>9</th>
<th>6</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Mirrored patch from original one

\[ D_1^{\text{mirror}} \]

\[ D_2^{\text{mirror}} \]

\[ D_3^{\text{mirror}} \]

\[ D_4^{\text{mirror}} \]
Proposed method (Multi-scanning strategy)

- After augmenting (rotation) the patch.
Proposed method (Multi-scanning strategy)

- After augmenting (flipping) the patch.
Proposed method (Multi-scanning strategy)

- In 16 directions, each of them is paired with another one.
- Forward and back-forward direction sequences are better to construct a bidirectional-RNN (Bi-RNN).
- Bidirectional-RNN has wider reception steps and consider the whole sequence.
Proposed method (spectral-spatial structure)

- Spectral-spatial structure for HSI with RNN.

By the multi-scanning strategy, we will get 16 sequences.

P: patch size
Proposed method (whole framework)

- 16 scanning directions are complementary for one patch.
- I learned their complementary features by designing an Outer-RNN.

**Inner-RNN**: Bidirectional RNN, has 25 or 49 steps (patch * patch).

**Outer-RNN**: has 8 steps (from \( P^1 \) to \( P^8 \)).

\( p^i \): a pair of sequences

\( y^i \): output feature from Bi-RNN.
Experiments with a single direction

$D_1^{\text{original}}$ 91.503%

$D_2^{\text{mirror}}$ 90.789%

$D_2^{\text{original}}$ 90.484%

$D_1^{\text{mirror}}$ 91.839%

$D_1^{\text{flip}}$ 89.844%

$D_1^{\text{rotate}}$ 90.072%

$D_1^{\text{flip}}$ 89.877%

$D_2^{\text{flip}}$ 89.844%

$D_2^{\text{rotate}}$ 90.939%

$D_2^{\text{flip}}$ 89.877%

$D_3^{\text{original}}$ 90.484%

$D_3^{\text{mirror}}$ 90.744%

$D_3^{\text{rotate}}$ 90.939%

$D_3^{\text{flip}}$ 90.570%

$D_4^{\text{original}}$ 90.527%

$D_4^{\text{mirror}}$ 90.744%

$D_4^{\text{rotate}}$ 91.087%

$D_4^{\text{flip}}$ 89.548%
Experiments with several directions

- Classification results with 1, 4, 8 and 16 directions on Indian Pines dataset.

1-direction: 91.839%  
4-directions: 92.531%  
8-directions: 93.106%  
16-directions: 93.989%

Multi-scanning:
1. Increasing the number of directions, the accuracy also increases.
2. It is more accurate and smoother in the flat area.
3. The boundary or outlines of regions are clearer.
Experiments with several directions

- Classification results on each class. (Indian Pines)

<table>
<thead>
<tr>
<th>Class</th>
<th>1 direction</th>
<th>4 direction</th>
<th>8 direction</th>
<th>16 direction</th>
<th>Li[9]</th>
<th>Leng[8]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfalfa</td>
<td>58.15%</td>
<td>95.12%</td>
<td>95.79%</td>
<td>91.26%</td>
<td>57.12%</td>
<td>85.22%</td>
</tr>
<tr>
<td>Corn-notill</td>
<td>89.33%</td>
<td>92.23%</td>
<td>93.44%</td>
<td>96.11%</td>
<td>77.98%</td>
<td>88.51%</td>
</tr>
<tr>
<td>Corn-mintill</td>
<td>83.55%</td>
<td>86.99%</td>
<td>90.29%</td>
<td>91.33%</td>
<td>69.45%</td>
<td>81.71%</td>
</tr>
<tr>
<td>Corn</td>
<td>73.02%</td>
<td>84.42%</td>
<td>84.01%</td>
<td>90.87%</td>
<td>74.11%</td>
<td>84.70%</td>
</tr>
<tr>
<td>Grass-pasture</td>
<td>94.68%</td>
<td>94.21%</td>
<td>95.59%</td>
<td>94.18%</td>
<td>89.46%</td>
<td>96.00%</td>
</tr>
<tr>
<td>Grass-trees</td>
<td>98.66%</td>
<td>97.49%</td>
<td>98.77%</td>
<td>97.21%</td>
<td>97.48%</td>
<td>98.23%</td>
</tr>
<tr>
<td>Grass-pasture-mowed</td>
<td>91.30%</td>
<td>95.78%</td>
<td>84.21%</td>
<td>92.77%</td>
<td>86.41%</td>
<td>93.77%</td>
</tr>
<tr>
<td>Hay-windrowed</td>
<td>97.11%</td>
<td>99.81%</td>
<td>99.11%</td>
<td>99.54%</td>
<td>96.33%</td>
<td>98.61%</td>
</tr>
<tr>
<td>Oats</td>
<td>66.69%</td>
<td>97.21%</td>
<td>96.88%</td>
<td>100%</td>
<td>43.59%</td>
<td>75.72%</td>
</tr>
<tr>
<td>Soybean-notill</td>
<td>89.59%</td>
<td>87.10%</td>
<td>94.51%</td>
<td>88.71%</td>
<td>76.88%</td>
<td>85.00%</td>
</tr>
<tr>
<td>Soybean-mintill</td>
<td>90.55%</td>
<td>92.14%</td>
<td>94.12%</td>
<td>94.15%</td>
<td>81.70%</td>
<td>90.01%</td>
</tr>
<tr>
<td>Soybean-clean</td>
<td>85.29%</td>
<td>91.99%</td>
<td>91.83%</td>
<td>93.18%</td>
<td>64.65%</td>
<td>81.76%</td>
</tr>
<tr>
<td>Wheat</td>
<td>99.59%</td>
<td>98.89%</td>
<td>99.17%</td>
<td>100%</td>
<td>99.58%</td>
<td>97.91%</td>
</tr>
<tr>
<td>Woods</td>
<td>96.08%</td>
<td>96.51%</td>
<td>97.34%</td>
<td>97.85%</td>
<td>95.26%</td>
<td>95.80%</td>
</tr>
<tr>
<td>Building-Grass-Tree</td>
<td>79.35%</td>
<td>81.29%</td>
<td>80.89%</td>
<td>79.89%</td>
<td>75.74%</td>
<td>81.99%</td>
</tr>
<tr>
<td>Stone-Steel-Tower</td>
<td>98.28%</td>
<td>98.20%</td>
<td>98.71%</td>
<td>98.59%</td>
<td>100%</td>
<td>96.96%</td>
</tr>
<tr>
<td>Overall accuracy</td>
<td>91.84%</td>
<td>92.53%</td>
<td>93.11%</td>
<td>93.99%</td>
<td>83.51%</td>
<td>90.32%</td>
</tr>
</tbody>
</table>

Yellow parts: obvious improvements by my method. Green parts: overall accuracy. Red parts: lower accuracy relatively in 3D-CNN.
Experiments with several directions

- Classification results with 1, 4, 8 and 16 directions on PaviaU. dataset.

Multi-scanning:
1. Increasing the number of directions, the accuracy also increases.
2. It is more accurate and smoother in the flat area.
3. The boundary or outlines of regions are clearer.
Comparison

- Classification probability for good examples.

- Spectral values for class 2

- Probability to each class for Li[9] 3D-CNN

- Probability to each class for Leng[8] 3D-CNN

- Probability to each class for my method
  - 4 directions
  - 8 directions
  - 16 directions
Comparison

- Classification probabilities for **bad examples**.
Comparison

- Classification probabilities for **bad examples**.
Conclusion

① My method can fully explore the spatial and spectral information together to discriminate each pixel.

② The results by my method become more accurate and smoother in the flat region.

③ The boundary or outlines of regions are more clear.

④ We hope it is a viable alternative for using CNN-based method in this task.
Reference


