Introduction

In this report, the issue "Validation accuracy is incorrect when using fully convolutional models #24" is explored by uncovering the importance of normalization methods as for the 3D-CNN (lee model) [1].

At first, the 3D-CNN 's architecture is fixed as Table 4. Note that, there are two attentive points. Firstly, the author seemed to employ 3D depthwise convolutional layers instead of common 3D convolutional layers, which can be viewed from some discussions of the original paper (**see Page 6, Paragraph 3, Line 5-7** and **Page 18, Paragraph 3, Line 1-3**). Secondly, the author seemed to use two fully connected layers: the first one is named as the **F1 layer** and the second one is named as the **Classification layer**, as the **Feature Volume** of the F1 layer is set to 144 rather than the class number of 9.

Table 4. Architecture of 3D-CNN in Pavia University scene.

Layer	Kernel Size	Kernel Number	Stride	Output Size	Feature Volumes
Input	-	-	-	$5 \times 5 \times 103$	1
C1	$3 \times 3 \times 7$	2	1	$3 \times 3 \times 97$	2
C2	$3 \times 3 \times 3$	4	1	$1 \times 1 \times 95$	8
F1	-	-	-	$1 \times 1 \times 1$	144
Classification	-	-	-	$1 \times 1 \times 1$	9

Secondly, it is probably to assume that the author resorted to a Band-wise Minmax Norm (*BMN*) before feeding HSI cubes into 3D-CNN, which consequently lead to this low validation accuracy issue when inappropriate normalization method is used. In order to testify to this assumption, I have conducted experiments on the commonly used Indian Pines data set and Pavia University data set using six normalization methods including *None*, Sample-wise L2 Norm (*SL2N*), Band-wise Minmax Norm (*BMN*), Band-wise Standard Norm (*BSN*), Image-wise Minmax Norm (*IMN*), and Image-wise Standard Norm (*ISN)*, respectively [2]. These normalizations are introduced as follows.

PS:
None: Applying none preprocessing.
SL2N: Normalizing with a unit Euclidean norm **along each sample**.
BMN: Converting the dynamics to [0, 1] **along each band**.
BSN: Normalizing first- and second-order moments **along each band**.
IMN: Converting the dynamics to [0, 1] **along the whole image**.
ISN: Normalizing first- and second-order moments **along the whole image**.

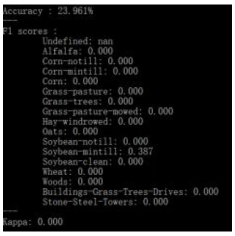
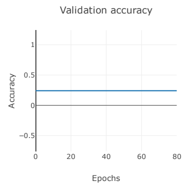
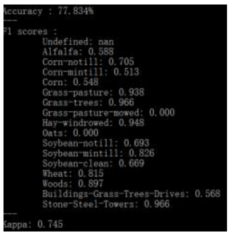
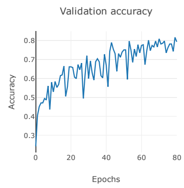
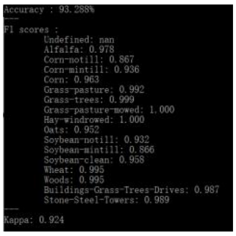
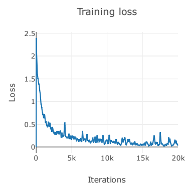
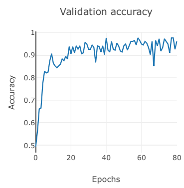
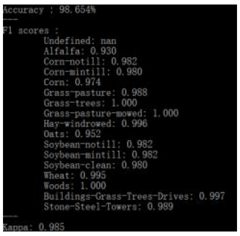
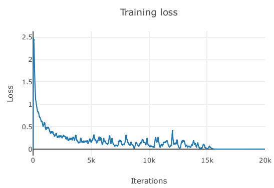
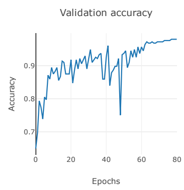
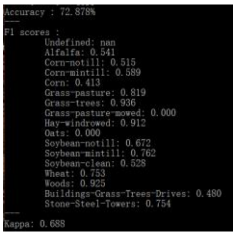
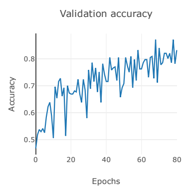
In this report, the following experiment are conducted using **DeepHyperX** with GeForce GTX 1070 (8GB), Intel Xeon E3 CPU, and 16 GB RAM.

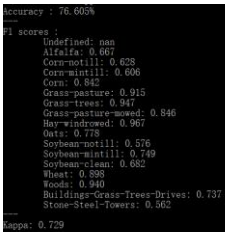
Experiment 1

In the first experiment, we attempted to uncover the significant difference between different normalization methods for 3D-CNN. For the hyperparameter setting, the iteration is set to

20000 for saving running time, thus the training epoch is the quotient of two numbers: the first one is the product of the iteration and the batch size, and the second one is the number of training samples, and others are set referred to the original paper. Then we compared above six normalization methods for 3D-CNN on the two data sets. The classification accuracies on the Indian Pines data set are reported on Table 1, and that on the Pavia University data set are reported on **Table 2**. As we can see, **the *BMN* and the *BSN* probably lead to a better accuracy while the *None* may result in the divergence of training process of 3D-CNN using the aforementioned experimental setting.**

Table 1: Classification accuracies using different normalizations for the Indian Pines data set (50% labeled sample per class used for training)

	Classification result	Training loss	Validation accuracy
None + 3D-CNN	 Accuracy : 23.961% --- F1 scores : Undefined: nan Alfalfa: 0.000 Corn-notill: 0.000 Corn-mintill: 0.000 Corn: 0.000 Grass-pasture: 0.000 Grass-trees: 0.000 Grass-pasture-mowed: 0.000 Hay-windrowed: 0.000 Oats: 0.000 Soybean-notill: 0.000 Soybean-mintill: 0.387 Soybean-clean: 0.000 Wheat: 0.000 Woods: 0.000 Buildings-Grass-Trees-Drives: 0.000 Stone-Steel-Towers: 0.000 --- Kappa: 0.000		
SL2N + 3D-CNN	 Accuracy : 77.834% --- F1 scores : Undefined: nan Alfalfa: 0.588 Corn-notill: 0.705 Corn-mintill: 0.513 Corn: 0.548 Grass-pasture: 0.938 Grass-trees: 0.966 Grass-pasture-mowed: 0.000 Hay-windrowed: 0.948 Oats: 0.000 Soybean-notill: 0.693 Soybean-mintill: 0.826 Soybean-clean: 0.669 Wheat: 0.815 Woods: 0.897 Buildings-Grass-Trees-Drives: 0.568 Stone-Steel-Towers: 0.966 --- Kappa: 0.745		
BMN + 3D-CNN	 Accuracy : 93.288% --- F1 scores : Undefined: nan Alfalfa: 0.978 Corn-notill: 0.867 Corn-mintill: 0.936 Corn: 0.963 Grass-pasture: 0.992 Grass-trees: 0.999 Grass-pasture-mowed: 1.000 Hay-windrowed: 1.000 Oats: 0.952 Soybean-notill: 0.932 Soybean-mintill: 0.866 Soybean-clean: 0.958 Wheat: 0.995 Woods: 0.995 Buildings-Grass-Trees-Drives: 0.987 Stone-Steel-Towers: 0.989 --- Kappa: 0.924		
BSN + 3D-CNN	 Accuracy : 98.654% --- F1 scores : Undefined: nan Alfalfa: 0.930 Corn-notill: 0.982 Corn-mintill: 0.980 Corn: 0.974 Grass-pasture: 0.988 Grass-trees: 1.000 Grass-pasture-mowed: 1.000 Hay-windrowed: 0.996 Oats: 0.952 Soybean-notill: 0.982 Soybean-mintill: 0.982 Soybean-clean: 0.980 Wheat: 0.995 Woods: 1.000 Buildings-Grass-Trees-Drives: 0.997 Stone-Steel-Towers: 0.989 --- Kappa: 0.985		
IMN + 3D-CNN	 Accuracy : 72.878% --- F1 scores : Undefined: nan Alfalfa: 0.541 Corn-notill: 0.515 Corn-mintill: 0.589 Corn: 0.413 Grass-pasture: 0.819 Grass-trees: 0.936 Grass-pasture-mowed: 0.000 Hay-windrowed: 0.912 Oats: 0.000 Soybean-notill: 0.672 Soybean-mintill: 0.762 Soybean-clean: 0.528 Wheat: 0.753 Woods: 0.925 Buildings-Grass-Trees-Drives: 0.480 Stone-Steel-Towers: 0.754 --- Kappa: 0.688		

<table>
<tr><td>ISN + 3D-CNN</td><td>
Accuracy : 76.605%

F1 scores :
Undefined: nan
Alfalfa: 0.667
Corn-notill: 0.628
Corn-mintill: 0.606
Corn: 0.842
Grass-pasture: 0.915
Grass-trees: 0.947
Grass-pasture-mowed: 0.846
Hay-windrowed: 0.967
Oats: 0.778
Soybean-notill: 0.576
Soybean-mintill: 0.749
Soybean-clean: 0.682
Wheat: 0.898
Woods: 0.940
Buildings-Grass-Trees-Drives: 0.737
Stone-Steel-Towers: 0.562

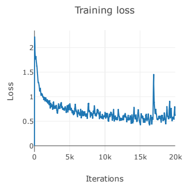
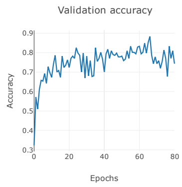
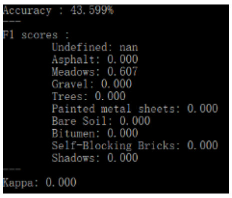
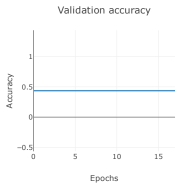
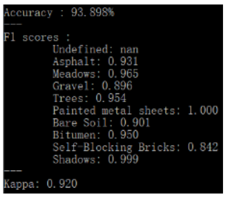
Kappa: 0.729</td><td></td><td></td></tr>
</table>

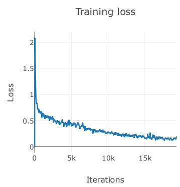
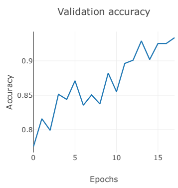
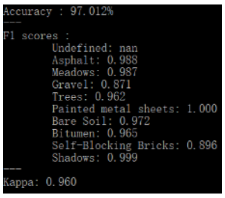
Table 2: Classification accuracies using different normalizations for the Pavia University data set (50% labeled sample per class used for training)

<table>
<tr><th></th><th>Classification result</th><th>Training loss</th><th>Validation accuracy</th></tr>
<tr><td>None + 3D-CNN</td><td>
Accuracy : 43.599%

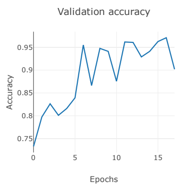
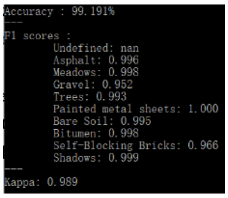
F1 scores :
Undefined: nan
Asphalt: 0.000
Meadows: 0.607
Gravel: 0.000
Trees: 0.000
Painted metal sheets: 0.000
Bare Soil: 0.000
Bitumen: 0.000
Self-Blocking Bricks: 0.000
Shadows: 0.000

Kappa: 0.000</td><td></td><td></td></tr>
<tr><td>SL2N + 3D-CNN</td><td>
Accuracy : 93.898%

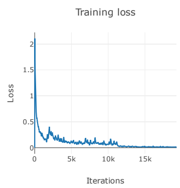
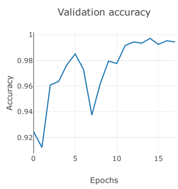
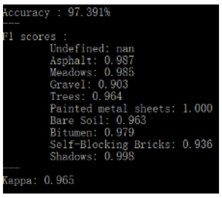
F1 scores :
Undefined: nan
Asphalt: 0.931
Meadows: 0.965
Gravel: 0.896
Trees: 0.954
Painted metal sheets: 1.000
Bare Soil: 0.901
Bitumen: 0.950
Self-Blocking Bricks: 0.842
Shadows: 0.999

Kappa: 0.920</td><td></td><td></td></tr>
<tr><td>BMN + 3D-CNN</td><td>
Accuracy : 97.012%

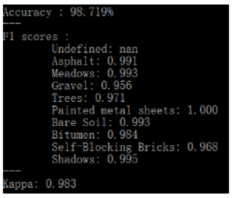
F1 scores :
Undefined: nan
Asphalt: 0.988
Meadows: 0.987
Gravel: 0.871
Trees: 0.962
Painted metal sheets: 1.000
Bare Soil: 0.972
Bitumen: 0.965
Self-Blocking Bricks: 0.896
Shadows: 0.999

Kappa: 0.960</td><td></td><td></td></tr>
<tr><td>BSN + 3D-CNN</td><td>
Accuracy : 99.191%

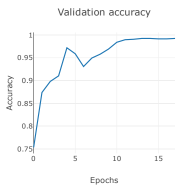
F1 scores :
Undefined: nan
Asphalt: 0.996
Meadows: 0.998
Gravel: 0.952
Trees: 0.993
Painted metal sheets: 1.000
Bare Soil: 0.995
Bitumen: 0.998
Self-Blocking Bricks: 0.966
Shadows: 0.999

Kappa: 0.989</td><td></td><td></td></tr>
<tr><td>IMN + 3D-CNN</td><td>
Accuracy : 97.391%

F1 scores :
Undefined: nan
Asphalt: 0.987
Meadows: 0.985
Gravel: 0.903
Trees: 0.964
Painted metal sheets: 1.000
Bare Soil: 0.963
Bitumen: 0.979
Self-Blocking Bricks: 0.936
Shadows: 0.998

Kappa: 0.965</td><td></td><td></td></tr>
<tr><td>ISN + 3D-CNN</td><td>
Accuracy : 98.719%

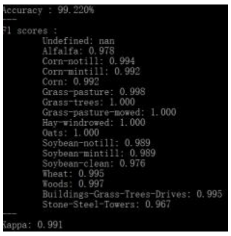
F1 scores :
Undefined: nan
Asphalt: 0.991
Meadows: 0.993
Gravel: 0.956
Trees: 0.971
Painted metal sheets: 1.000
Bare Soil: 0.993
Bitumen: 0.984
Self-Blocking Bricks: 0.968
Shadows: 0.995

Kappa: 0.983</td><td></td><td></td></tr>
</table>

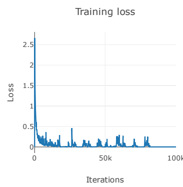
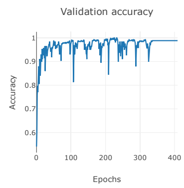
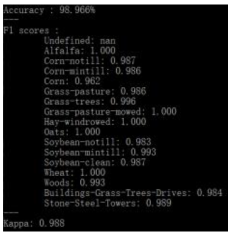
Experiment 2

In the second experiment, our goal is to replicate the results in paper [1] on the two data sets. For the hyperparameter setting, the iteration is set to 100000 referred to the original paper. We employ the *BMN* and the *BSN* respectively as they perform better in *Experiment 1*. The ultimate results are listed in **Table 3**. Note that our results are slightly higher than those of the original paper. For example, in our experiments, 3D-CNN with *BMN* obtained an OA of 99.20% which is 0.13% higher than the OA of 99.07% [1] on the Indian Pines data set, and 3D-CNN with *BMN* obtained an OA of an OA of 99.61% which is 0.22% higher than the OA of 99.39% [1] on the Pavia University data set. This phenomenon may be explained by several reasons: firstly, the randomness existed in the training process and sample selection; secondly, the different initialization of parameters; thirdly, the different numbers of training samples per class. Regardless of this exceptional improvement, this experiment has successfully replicated the results of 3D-CNN on the two data sets, and demonstrated the necessity of choosing an appropriate normalization for 3D-CNN.

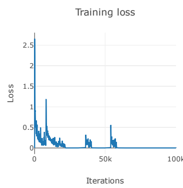
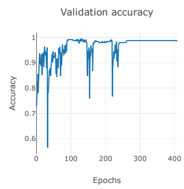
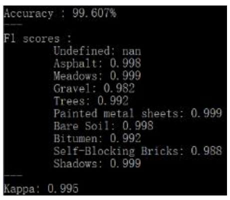
Table 3: Classification accuracies for the two data set (50% labeled sample per class used for training)

<table>
<tr><th>Indian Pines</th><th>Classification result</th><th>Training loss</th><th>Validation accuracy</th></tr>
<tr><td>BMN + 3D-CNN</td><td>
Accuracy : 99.220%

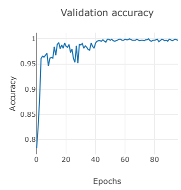
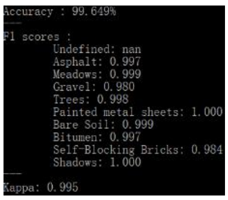
F1 scores :
Undefined: nan
Alfalfa: 0.978
Corn-notill: 0.994
Corn-mintill: 0.992
Corn: 0.992
Grass-pasture: 0.998
Grass-trees: 1.000
Grass-pasture-mowed: 1.000
Hay-windrowed: 1.000
Oats: 1.000
Soybean-notill: 0.989
Soybean-mintill: 0.989
Soybean-clean: 0.976
Wheat: 0.995
Woods: 0.997
Buildings-Grass-Trees-Drives: 0.995
Stone-Steel-Towers: 0.967

Kappa: 0.991</td><td>
</td><td>
</td></tr>
<tr><td>BSN + 3D-CNN</td><td>
Accuracy : 98.966%

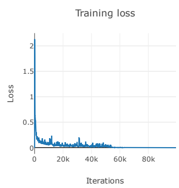
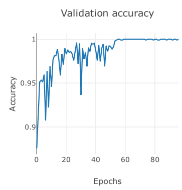
F1 scores :
Undefined: nan
Alfalfa: 1.000
Corn-notill: 0.987
Corn-mintill: 0.986
Corn: 0.962
Grass-pasture: 0.986
Grass-trees: 0.996
Grass-pasture-mowed: 1.000
Hay-windrowed: 1.000
Oats: 1.000
Soybean-notill: 0.983
Soybean-mintill: 0.993
Soybean-clean: 0.987
Wheat: 1.000
Woods: 0.993
Buildings-Grass-Trees-Drives: 0.984
Stone-Steel-Towers: 0.989

Kappa: 0.988</td><td>
</td><td>
</td></tr>
<tr><th>Pavia University</th><th>Classification result</th><th>Training loss</th><th>Validation accuracy</th></tr>
<tr><td>BMN + 3D-CNN</td><td>
Accuracy : 99.607%

F1 scores :
Undefined: nan
Asphalt: 0.998
Meadows: 0.999
Gravel: 0.982
Trees: 0.992
Painted metal sheets: 0.999
Bare Soil: 0.998
Bitumen: 0.992
Self-Blocking Bricks: 0.988
Shadows: 0.999

Kappa: 0.995</td><td>
</td><td>
</td></tr>
<tr><td>BSN + 3D-CNN</td><td>
Accuracy : 99.649%

F1 scores :
Undefined: nan
Asphalt: 0.997
Meadows: 0.999
Gravel: 0.980
Trees: 0.998
Painted metal sheets: 1.000
Bare Soil: 0.999
Bitumen: 0.997
Self-Blocking Bricks: 0.984
Shadows: 1.000

Kappa: 0.995</td><td>
</td><td>
</td></tr>
</table>

Conclusion

In summary, even though the original author clearly delineated that 3D-CNN does not rely on any preprocessing, choosing a suitable normalization method is vital for its training process. In this circumstance, **DeepHyperX** tools can realize similar or even higher accuracy of 3D-CNN if an appropriate normalization is used. **In a word, the discussed issue may be the consequence of an unsuitable normalization method**.

Reference

[1] Li Y, Zhang H, Shen Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network[J]. Remote Sensing, 2017, 9(1): 67.

[2] N. Audebert, B. Le Saux and S. Lefevre, "Deep Learning for Classification of Hyperspectral Data: A Comparative Review," in *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 159-173, June 2019, doi: 10.1109/MGRS.2019.2912563.