

Segmentation of Hyperspectral Images

João Resende¹, José Eduardo Silva²
FEUP - Faculdade de Engenharia da Universidade do Porto
Porto, Portugal
¹up202003528@fe.up.pt, ²up202003505@fe.up.pt

Abstract

This paper presents the work developed for the “Computer Vision” course unit of the Doctoral Program in Electrical and Computer Engineering (PDEEC), from the Faculty of Engineering of University of Porto (FEUP).

In this work, we used part of the Hyko dataset, which is a collection of labeled hyperspectral images taken from an area scan camera mounted on a moving car, to train classical (kNN) and neural-network based machine learning models.

Due to the limitations of the available training data, the results were subpar. For each technique, the limitations and possible improvements are discussed.

1. Introduction

Hyperspectral imaging is a growing topic in remote sensing in recent years. Hyperspectral imaging systems are typically used in satellites and airborne systems in order to acquire information of the Earth’s surface. This area has been firmly established for many years in several domains, such as Earth observation, agriculture, industry, inspection and surveillance [1].

A hyperspectral imaging system is able to capture “light intensity” information for multiple spectral bands, which helps in determining which elements make up the scene. In other words, each pixel in a hyperspectral image acts as an array which contains intensity values for different wavelengths [2], as shown in Figure 1.

In this work, we address the problem of hyperspectral image segmentation using the kNN algorithm and two different approaches based on neural networks. The dataset used is the HyKO dataset [3], which contains urban traffic and rural road scenes recorded using hyperspectral cameras mounted on a moving car, with per-pixel labels.

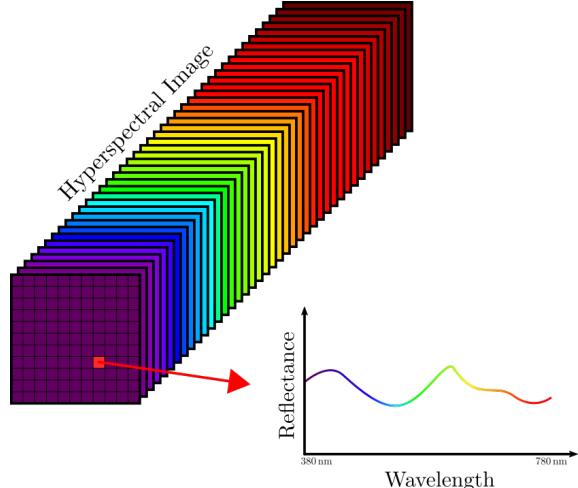


Figure 1. Spectral signatures of a pixel vector and its corresponding spectral signature.

2. Related Work

Classification of hyperspectral images is a challenging task due to relatively low spatial resolution, limited number of available training samples and the high dimensionality of the data.

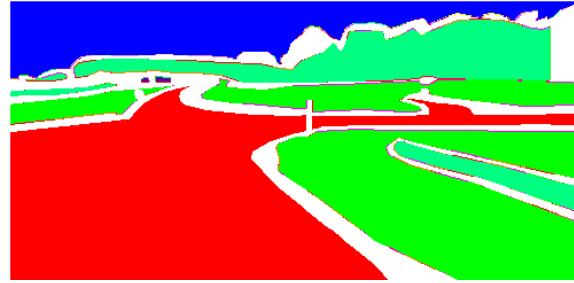
A problem arising from low spatial resolution is the mixture of several materials covered by a pixel. In order to separate these materials, a technique known as linear unmixing is used. This process consists in decomposing a spectrum into a collection of spectra and their corresponding fractions [4].

Due to the complexity of hyperspectral data, traditional techniques are used, with the most common classifiers being K-nearest neighbors ($kNNs$) and Support vector machines (SVMs).

Hyperspectral image processing has seen multiple research endeavors [1, 5, 6, 7, 8], with some studies attempting techniques based on neural networks [9, 10, 11, 12, 13, 14, 15].



(a) Sliced image (Wavelength 578.54 nm).



(b) Corresponding labeled mask.

Figure 2. Example of the per-pixel labeled data.

3. Hyko Dataset

3.1. Overview

The HyKo dataset¹ is a hyperspectral image dataset captured with low-cost, area scan cameras which are capable of capturing an entire spectral cube in one shot.

The hyperspectral cameras used for the creation of the dataset were the MQ022HG-IM-SM4X4-VIS (VIS) camera, with a visual spectrum ranging from 470 nm to 630 nm, and the MQ022HG-IM-SM5X5-NIR (NIR) camera, with a near infrared spectrum ranging from 600 nm to 975 nm [3].

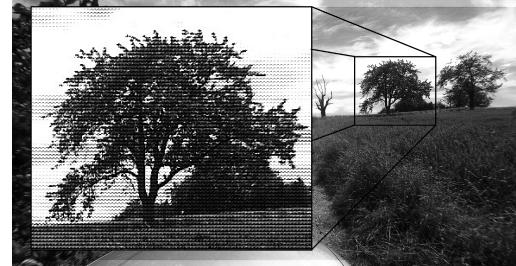
Both cameras use one spectral filter per each pixel, arranged in mosaics over the entire sensor. For the VIS camera, each filter has a size of 4×4 (resulting in 16 spectral bands). For the NIR camera, each filter has a size of 5×5 (resulting in 25 spectral bands). In Figure 3, the raw images taken by the VIS and NIR cameras are shown.

By taking a raw image and arranging each filter mosaic into a super-pixel (i.e. a pixel containing), an hyperspectral image is obtained. After processing, each hyperspectral image has a spatial resolution of 254×510 and 214×407 for the VIS and NIR cameras, respectively.

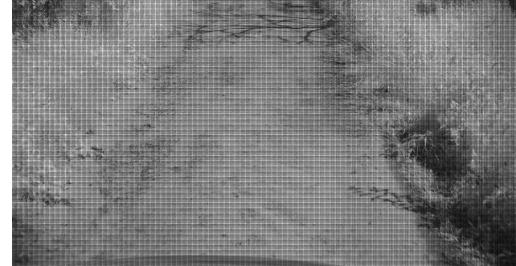
The HyKo dataset is actually 2 different datasets. The Hyko 1 dataset contains more images but only 4 labels per pixel (excluding unlabeled data), while the Hyko 2 contains less images but has 10 labels. For this work, we choose only the images from the Hyko 2 dataset, example illustrated in Figure 2.

3.2. File Format and Labels

The entirety of the Hyko 2 dataset hyperspectral data is saved as MATLAB level 5 mat-files, where every mat-file holds exactly one hyperspectral image and the per-pixel manually-assigned labels. The labels are divided into three categories: semantic, material and



(a) Example raw image taken by the VIS camera.



(b) Example raw image taken by the NIR camera.

Figure 3. Raw images of NIR and VIS camera with visible mosaic pattern [3].

driveability. The semantic category was selected for this work. It should be noted, however, that some pixels were not labeled as they belong to the borders between clearly defined regions.

The semantic labels are as follows:

- ID 0:** Undefined/Unlabeled pixel;
- **ID 1:** Road;
- **ID 2:** Sidewalk;
- **ID 3:** Lane Markers;
- **ID 4:** Grass;
- **ID 5:** Vegetation (not grass - not drivable);
- **ID 6:** Panels/Signs/Traffic Lights;
- **ID 7:** Building/Wall/Others;
- **ID 8:** Vehicle (Car/Truck/Train/Bus etc...);
- **ID 9:** Pedestrian/Cyclist/Motocyclist/Animal;
- **ID 10:** Sky;

¹Download: <https://wp.uni-koblenz.de/hyko/>

3.3. Dataset Errors

Interestingly, the mat-files of the VIS images from the HyKo 2 dataset contain only 15 spectral bands, as opposed to the expected 16 spectral bands. We do not know the reason for the exclusion of one of the bands, as the authors themselves indicate that the VIS images in the dataset contain 16 of them [3].

Additionally, we also found a VIS mat-file - named "set_11564_spec.annotated.mat" - that did not contain any labels whatsoever. We can only assume this was an error on the dataset authors' behalf, and as such we have opted to exclude this image.

4. Proposed Solutions

As previously mentioned, a kNN algorithm and two different architectures of neural networks were used for segmentation. Whenever possible, class ID 0 (Undefined) was removed from the dataset as to not influence the results.

4.1. kNN Algorithm

The kNN algorithm takes a set of classified training pixels and an unclassified training pixel, classifying it using the k nearest neighbors by majority rule.

For each type of image (VIS and NIR), 2 models were used - one with $k = 3$ and another with $k = 5$ - resulting in a total of 4 different models.

For both VIS and NIR models, the pixels contained in 80% of the available images were used as training samples, and 20% were used as test samples. For the VIS images, this corresponds to 130 images (16 840 200 pixels) for training and 32 images (4 145 280 pixels) for testing. For the NIR images, this corresponds to 62 images (5 400 076 pixels) for training and 16 images (1 393 568 pixels) for testing.

4.2. Neural Network - Full Image

The first neural network architecture used takes the full hyperspectral image as input and attempts to classify all pixels in the image at once. The network is fully convolutional, comprising 3D convolutional layers followed by 3D max pooling layers on the down-sampling path, and transpose 3D convolutional layers followed by 3D upsampling layers on the upsampling path. Both downsampling and upsampling paths have a depth D equal to 3. The network's final layer is a single-filter 3D convolutional layer. The architecture is shown in Figure 4.

Since the network relies solely on 3D layers, the output is also 3D, with the dimensions of the input hyperspectral image. To train the network, each 2D labeled

image was repeated L times to get a 3 dimensional output. When testing the network, its output is averaged along the extra dimension in order to collapse it, and each pixel label is rounded to the nearest integer, thus obtaining the desired 2D output. With this technique, the network can be treated as a regression model for training and validation, but the test results can be analyzed on a per-pixel basis, as if the model was a typical classification model.

For both VIS and NIR models, 70% of the available images were used for training, 10% for validation, and 20% for testing. For the VIS images, this corresponds to 114 images for training, 16 images for validation, and 32 images for testing. For the NIR images, this corresponds to 55 images for training, 7 images for validation, and 16 images for testing.

4.3. Neural Network - Patches

The second neural network architecture used is based on current state-of-the-art architectures[16, 17, 13], taking patches of $N \times N \times L$, where N denotes the spatial resolution of the patch and L is the number of spectral bands, and attempting to classify the center pixel. The architecture is shown in Figure 5.

For both VIS and NIR models, the pixels contained in 70% of the available images were used as training samples, 10% for validation, and 20% for testing. For the VIS images, this corresponds to 114 images (14 767 560 pixels) for training, 16 images for validation (2 072 640 pixels), and 32 images (4 145 280 pixels) for testing. For the NIR images, this corresponds to 55 images (4 790 390 pixels) for training, 7 images for validation (609 686 pixels), and 16 images (1 393 568 pixels) for testing.

5. Results and Discussion

5.1. kNN Algorithm

Tables 1 through 4 show the results for the kNN Algorithm. Since the spectra are used as the feature vectors, the classes are not well defined since objects in the same class might not necessarily share the same spectral signature. This is particularly obvious, for instance, in Tables 1 and 2, where class ID 8 (Vehicles) has zero F1-Score, since their spectral signature changes with chassis composition and paint color.

Increasing k appears to have no effect on the overall quality of the results. One possible way to improve the results would be to use a weighted kNN algorithm, but further analysis of the available data would be necessary to assess the validity of this claim. Another possible approach would be to use feature extraction techniques, such as Principal component analysis (PCA),

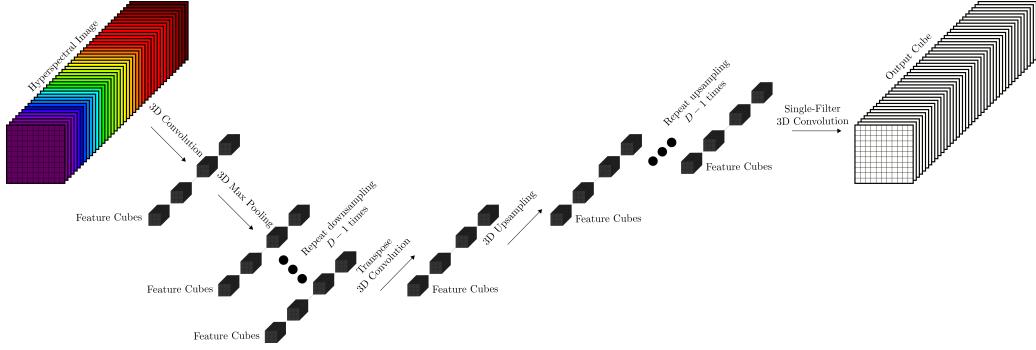


Figure 4. Full image neural network architecture.

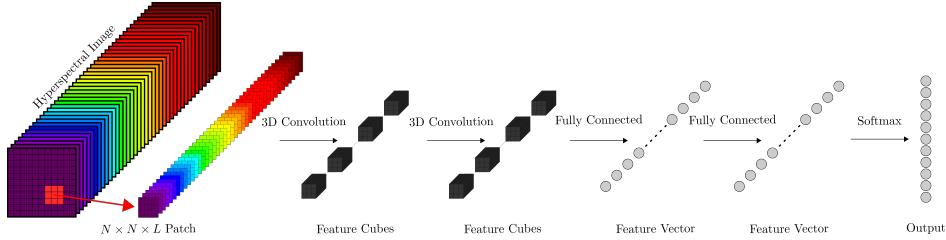


Figure 5. Patch-based neural network architecture.

to identify the most relevant wavelengths to use as feature vectors. Finally, it would also be possible to add spatial features (e.g. via kernel convolutions) to the kNN's feature vector to better define the classes.

5.2. Neural Network - Full Image

For the full image neural network, two versions were tested. The first version (8F) uses 8 filters in the first convolutional layer, while the second version (16F) uses 16. In both networks, the number of filters double per each depth level D .

As can be seen in Tables 5 to 8, the results were similar for the NIR images and significantly worse for the VIS images, when compared to the kNN algorithm. Part of the reason is due to the existence of the ID 0 (Undefined) class, which is not possible to remove from the dataset since the network takes the full image.

It is important to note that even though the available training data is the same, the quantity of data for the full network is orders of magnitude smaller than the other approaches since the network takes a full image as input, where the other methods take individual pixels or patches of pixels. This possibly results in underfitting, particularly in the NIR images where most of the training data are images of asphalt roads.

5.3. Neural Network - Patches

For patch-based neural network, two versions were tested. The first version uses a patch of $N = 3$, while

the second one uses a patch of $N = 5$. Even though the network itself is state-of-the-art, the results obtained for both versions are the worst of all approaches. Looking at the results shown in Tables 9 to 12, the reason is fairly obvious. The network seems to have learned that by always classifying pixels as belonging to class ID 1 (Road), it manages to get high accuracies in the NIR images. The same can be seen, although to a lesser extent, in the VIS images, where recall values for class ID 1 (Road) are also extremely high.

This problem could probably be solved by oversampling the dataset, thus compensating for the huge class imbalance.

Table 1. kNN NIR (k=3)

Class	Precision	Recall	F1-Score	Support
ID 1	0.87	0.75	0.81	973733
ID 2	0.12	0.02	0.03	38085
ID 3	0.08	0.85	0.14	34398
ID 4	0.76	0.61	0.68	128155
ID 5	0.00	0.00	0.00	686
ID 6	0.00	0.00	0.00	0
ID 7	0.00	0.00	0.00	591
ID 8	0.00	0.00	0.00	170582
ID 9	0.00	0.00	0.00	0
ID 10	0.00	0.00	0.00	0

Accuracy	0.63	1346230
----------	------	---------

Table 2. kNN NIR (k=5)

Class	Precision	Recall	F1-Score	Support
■ ID 1	0.87	0.75	0.81	973733
■ ID 2	0.13	0.02	0.03	38085
■ ID 3	0.08	0.85	0.14	34398
■ ID 4	0.77	0.60	0.68	128155
■ ID 5	0.00	0.00	0.00	686
■ ID 6	0.00	0.00	0.00	0
■ ID 7	0.00	0.00	0.00	591
■ ID 8	0.00	0.00	0.00	170582
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.00	0.00	0.00	0

Accuracy	0.62	1346230
----------	------	---------

Table 3. kNN VIS (k=3)

Class	Precision	Recall	F1-Score	Support
■ ID 1	0.75	0.89	0.81	1063158
■ ID 2	0.37	0.16	0.23	345318
■ ID 3	0.21	0.25	0.23	32996
■ ID 4	0.62	0.60	0.61	269587
■ ID 5	0.57	0.51	0.54	306966
■ ID 6	0.13	0.05	0.07	61665
■ ID 7	0.24	0.37	0.29	115656
■ ID 8	0.40	0.32	0.36	58135
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.97	1.00	0.98	454003

Accuracy	0.68	2707484
----------	------	---------

Table 4. kNN VIS (k=5)

Class	Precision	Recall	F1-Score	Support
■ ID 1	0.76	0.90	0.83	1063158
■ ID 2	0.42	0.15	0.22	345318
■ ID 3	0.23	0.25	0.24	32996
■ ID 4	0.62	0.63	0.63	269587
■ ID 5	0.57	0.54	0.55	306966
■ ID 6	0.13	0.05	0.07	61665
■ ID 7	0.25	0.40	0.31	115656
■ ID 8	0.42	0.33	0.37	58135
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.97	1.00	0.98	454003

Accuracy	0.69	2707484
----------	------	---------

Table 5. Full Image Network NIR (8F)

Class	Precision	Recall	F1-Score	Support
□ ID 0	0.18	0.00	0.01	47338
■ ID 1	0.89	0.72	0.80	973733
■ ID 2	0.08	0.32	0.13	38085
■ ID 3	0.08	0.45	0.14	34398
■ ID 4	0.45	0.94	0.61	128155
■ ID 5	0.08	0.01	0.02	686
■ ID 6	0.00	0.00	0.00	0
■ ID 7	0.00	0.00	0.00	591
■ ID 8	0.00	0.00	0.00	170582
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.00	0.00	0.00	0

Accuracy	0.61	1393568
----------	------	---------

Table 6. Full Image Network NIR (16F)

Class	Precision	Recall	F1-Score	Support
□ ID 0	0.10	0.03	0.04	47338
■ ID 1	0.86	0.74	0.79	973733
■ ID 2	0.09	0.31	0.14	38085
■ ID 3	0.05	0.31	0.08	34398
■ ID 4	0.61	0.92	0.73	128155
■ ID 5	0.00	0.00	0.00	686
■ ID 6	0.00	0.00	0.00	0
■ ID 7	0.00	0.00	0.00	591
■ ID 8	0.00	0.00	0.00	170582
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.00	0.00	0.00	0

Accuracy	0.62	1393568
----------	------	---------

Table 7. Full Image Network VIS (8F)

Class	Precision	Recall	F1-Score	Support
□ ID 0	0.33	0.00	0.00	1437796
■ ID 1	0.61	0.98	0.75	1063158
■ ID 2	0.16	0.13	0.14	345318
■ ID 3	0.01	0.04	0.01	32996
■ ID 4	0.27	0.69	0.39	269587
■ ID 5	0.28	0.43	0.34	306966
■ ID 6	0.08	0.21	0.11	61665
■ ID 7	0.25	0.18	0.21	115656
■ ID 8	0.03	0.02	0.02	58135
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.98	0.00	0.01	454003

Accuracy	0.35	4145280
----------	------	---------

Table 8. Full Image Network VIS (16F)

Class	Precision	Recall	F1-Score	Support
□ ID 0	0.00	0.00	0.00	1437796
■ ID 1	0.61	0.96	0.75	1063158
■ ID 2	0.18	0.13	0.15	345318
■ ID 3	0.00	0.02	0.01	32996
■ ID 4	0.37	0.64	0.47	269587
■ ID 5	0.33	0.71	0.45	306966
■ ID 6	0.06	0.23	0.09	61665
■ ID 7	0.25	0.25	0.25	115656
■ ID 8	0.05	0.04	0.05	58135
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.03	0.00	0.00	454003

Accuracy	0.36	4145280
----------	------	---------

Table 9. Patch-based Network NIR (N=3)

Class	Precision	Recall	F1-Score	Support
■ ID 1	0.72	1.00	0.84	973733
■ ID 2	0.00	0.00	0.00	38085
■ ID 3	0.00	0.00	0.00	34398
■ ID 4	0.00	0.00	0.00	128155
■ ID 5	0.00	0.00	0.00	686
■ ID 6	0.00	0.00	0.00	0
■ ID 7	0.00	0.00	0.00	591
■ ID 8	0.00	0.00	0.00	170582
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.00	0.00	0.00	0

Accuracy	0.72	1346230
----------	------	---------

Table 10. Patch-based Network NIR (N=5)

Class	Precision	Recall	F1-Score	Support
■ ID 1	0.75	0.80	0.77	973733
■ ID 2	0.00	0.00	0.00	38085
■ ID 3	0.00	0.00	0.00	34398
■ ID 4	0.21	0.50	0.29	128155
■ ID 5	0.00	0.00	0.00	686
■ ID 6	0.00	0.00	0.00	0
■ ID 7	0.00	0.00	0.00	591
■ ID 8	0.00	0.00	0.00	170582
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.00	0.00	0.00	0

Accuracy	0.63	1346230
----------	------	---------

Table 11. Patch-based Network VIS (N=3)

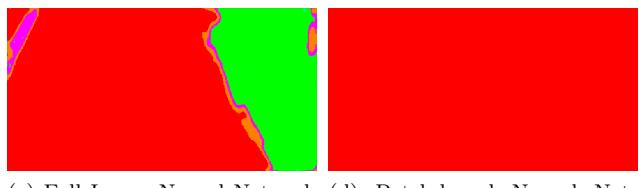
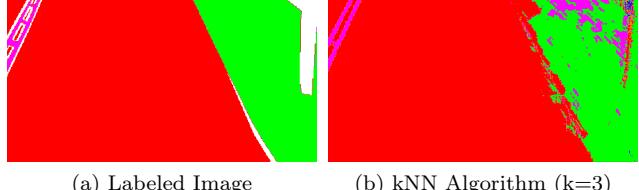
Class	Precision	Recall	F1-Score	Support
■ ID 1	0.40	0.97	0.56	1063158
■ ID 2	0.00	0.00	0.00	345318
■ ID 3	0.00	0.00	0.00	32996
■ ID 4	0.00	0.00	0.00	269587
■ ID 5	0.00	0.00	0.00	306966
■ ID 6	0.00	0.00	0.00	61665
■ ID 7	0.00	0.00	0.00	115656
■ ID 8	0.03	0.04	0.03	58135
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.00	0.00	0.00	454003

Accuracy	0.38	2707484
----------	------	---------

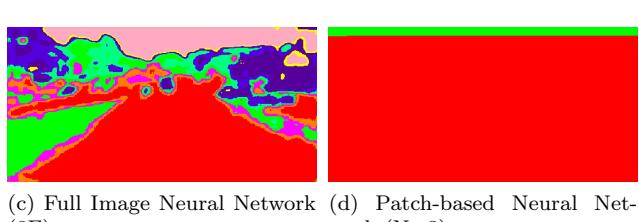
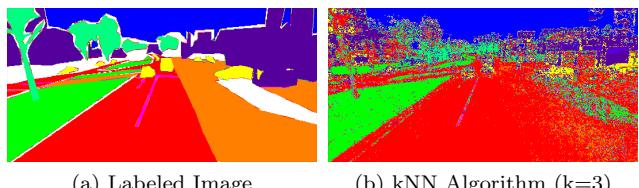
Table 12. Patch-based Network VIS (N=5)

Class	Precision	Recall	F1-Score	Support
■ ID 1	0.48	0.77	0.59	1063158
■ ID 2	0.12	0.19	0.14	345318
■ ID 3	0.00	0.01	0.00	32996
■ ID 4	0.14	0.01	0.02	269587
■ ID 5	0.00	0.00	0.00	306966
■ ID 6	0.00	0.00	0.00	61665
■ ID 7	0.07	0.02	0.03	115656
■ ID 8	0.00	0.00	0.00	58135
■ ID 9	0.00	0.00	0.00	0
■ ID 10	0.00	0.00	0.00	454003

Accuracy	0.33	2707484
----------	------	---------



(c) Full Image Neural Network (8F) (d) Patch-based Neural Network (N=3)
Figure 6. Example of the outputs of different methods (NIR)



(c) Full Image Neural Network (8F) (d) Patch-based Neural Network (N=3)
Figure 7. Example of the outputs of different methods (VIS)

References

- [1] Pedram Ghamisi, Naoto Yokoya, Jun Li, Wenzhi Liao, Sicong Liu, Javier Plaza, Behnood Rasti, and Antonio Plaza. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 2017. 1
- [2] Chein-I Chang. *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Springer US, 2003. 1
- [3] Christian Winkens, Florian Sattler, Veronika Adams, and Dietrich Paulus. Hyko: A spectral dataset for scene understanding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 1, 2, 3
- [4] Jiaojiao Wei and Xiaofei Wang. An overview on linear unmixing of hyperspectral data. *Mathematical Problems in Engineering*, 2020:1–12, 08 2020. 1
- [5] Pedram Ghamisi, Javier Plaza, Yushi Chen, Jun Li, and Antonio J Plaza. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geoscience and Remote Sensing Magazine*, 2017. 1
- [6] Yushi Chen, Shunli Ma, Xi Chen, and Pedram Ghamisi. Hyperspectral data clustering based on density analysis ensemble. *Remote Sensing Letters*, 2017. 1
- [7] Pedram Ghamisi, Abder-Rahman Ali, Micael S. Couceiro, and Jón Atli Benediktsson. A novel evolutionary swarm fuzzy clustering approach for hyperspectral imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2015. 1
- [8] Q. Jackson and D.A. Landgrebe. Adaptive bayesian contextual classification based on markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 2002. 1
- [9] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. Deep learning-based classification of hyperspectral data. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 2014. 1
- [10] Chao Tao, Hongbo Pan, Yansheng Li, and Zhengrou Zou. Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, 2015. 1
- [11] Yushi Chen, Xing Zhao, and Xiuping Jia. Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2015. 1
- [12] Chen Ding, Ying Li, Yong Xia, Wei Wei, Lei Zhang, and Yanning Zhang. Convolutional neural networks based hyperspectral image classification method with adaptive kernels. *Remote Sensing*, 2017. 1
- [13] Jiaojiao Li, Xi Zhao, Yunsong Li, Qian Du, Bobo Xi, and Jing Hu. Classification of hyperspectral imagery using a new fully convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 2018. 1, 3
- [14] Kunshan Huang, Shutao Li, Xudong Kang, and Leyuan Fang. Spectral-spatial hyperspectral image classification based on knn. *Sensing and Imaging*, 17(1):1, 2016. 1
- [15] T A Moughal. Hyperspectral image classification using support vector machine. *Journal of Physics: Conference Series*, 439:012042, jun 2013. 1
- [16] Jiayan Cao, Zhao Chen, and Bin Wang. Deep convolutional networks with superpixel segmentation for hyperspectral image classification. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3310–3313, 2016. 3
- [17] Fahim Irfan Alam, Jun Zhou, Alan Wee-Chung Liew, and Xiuping Jia. Crf learning with cnn features for hyperspectral image segmentation. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 6890–6893, 2016. 3