3D/2D regularized CNN feature hierarchy for Hyperspectral image classification

Muhammad Ahmad, Manuel Mazzara, and Salvatore Distefano

Abstract—Convolutional Neural Networks (CNN) have been rigorously studied for Hyperspectral Image Classification (HSIC) and are known to be effective in exploiting joint spatial-spectral information with the expense of lower generalization performance and learning speed due to the hard labels and non-uniform distribution over labels. Several regularization techniques have been used to overcome the aforesaid issues. However, sometimes models learn to predict the samples extremely confidently which is not good from a generalization point of view. Therefore, this paper proposed an idea to enhance the generalization performance of a hybrid CNN for HSIC using soft labels that are a weighted average of the hard labels and uniform distribution over ground labels. The proposed method helps to prevent CNN from becoming over-confident. We empirically show that in improving generalization performance, label smoothing also improves model calibration which significantly improves beam-search. Several publicly available Hyperspectral datasets are used to validate the experimental evaluation which reveals improved generalization performance, statistical significance, and computational complexity as compared to the state-of-the-art models. The source code will be made available at https://github.com/mahmad00.

Index Terms—Beam-search; Label Smoothing; Hybrid Convolutional Neural Network (CNN); Hyperspectral Images Classification (HSIC).

I. INTRODUCTION

HYPERSPECTRAL Imaging (HSI) has been extensively utilized for many real-world applications [1]; for instance, Crop monitoring, vegetation coverage, precision agriculture, land resources, oil spills, water quality, meat adulteration [2], [3], adulteration in house hold products such as color adulteration detection in red chilli [4], microbial spoilage and shelf-life of bakery products [5].

Thus, HSI Classification (HSIC) has received remarkable attention and intensive research results have been reported in the past few decades [6]. According to the literature, HSIC can be categorized into spatial, spectral, and spatial-spectral feature methods [7]. The spectral feature can be labeled as a primitive characteristic of HSI also known as spectral curve or vector whereas the spatial feature contains the relationship between the central pixel and its context which significantly uplifts the performance [8].

In the last few years, deep learning especially Convolutional Neural Network (CNN) has received widespread attention due to its ability to automatically learn non-linear features for classification, i.e., overcome the challenges of hand-crafted features for classification. Moreover, it can jointly investigate the spatial-spectral information and such models can be categorized into two groups, i.e., single and two-stream, more information regarding single or two-stream methods can be found in [9]. This work explicitly investigate a single-stream method but in a combination of 3-D and 2-D layers.

Irrespective of the single or two-stream method, all deep learning frameworks are sensitive to the loss which needs to be minimized [10], [11]. Several classical works showed that the gradient descent to minimize cross-entropy performs better in terms of classification and has fast convergence, however, to some extent, leads to the overfitting. Several regularization techniques such as dropout, L1, L2, etc., have been used to overcome the overfitting issues together with several other exotic objectives performed exceptionally well than the standard cross-entropy [12]. Recently, a work [13] proposed a Label Smoothing (LS) technique that improves the accuracy significantly by computing cross-entropy with a weighted mixture of targets with uniform distribution instead of hard-coded targets.

Since then, LS has been known to improve the classification performance of deep models [14]. However, the original idea was used to improve the classification performance of only the inception model on ImageNet data [13]. However, since then, various image classification models (training models) have used LS [15], [16]. Though the LS technique is widely used trick to improve the classification performance and to speed up the convergence process, however, it has not been much explored for HSIC, and above all, it has not been much explored regarding when and why LS should work. This is due to, in HSIC, sometimes models learn to predict the training samples extremely confidently which is not good from a generalization point of view.

Therefore, this paper proposed an idea to enhance the generalization performance of a hybrid CNN for HSIC using soft labels that are a weighted average of the hard labels and uniform distribution over target labels. The proposed method helps to prevent Hybrid CNN from becoming over-confident. We empirically show that in improving generalization performance, LS also improves model calibration which significantly improves beam-search. Several publicly available Hyperspectral datasets are used to validate the experimental evaluation which reveals improved generalization performance, statistical significance, and computational complexity as compared to the state-of-the-art 2-D/3-D CNN models.

II. PROPOSED METHODOLOGY

Let us assume that the Hyperspectral data can be represented as \( R^{(M \times N) \times B^*} = [r_1, r_2, r_3, \ldots, r_S]^T \), where \( B^* \) be the total...
number of bands and \((M \times N)\) are the samples per band belonging to \(Y\) classes and \(r_i = [r_{1,i}, r_{2,i}, r_{3,i}, \ldots, r_{B,i}]^T\) is the \(i^{th}\) sample in the Hyperspectral Data. Suppose \((r_i, y_i) \in (R^{M \times N \times B}, R^Y)\), where \(y_i\) is the class label of the \(i^{th}\) sample. Later instead of using the “hot encoding” in hybrid model, we introduce a smoothing technique \(\mu(y|r_i)\). Thus the new ground truths \((r_i, y_i)\) would be:

\[
p'(y|r_i) = (1 - \varepsilon)p(y|r_i) + \varepsilon\mu(y|r_i)
\]

\[
f(x) = \begin{cases} 1 - \varepsilon + \varepsilon\mu(y|r_i) & \text{if } y = y_i \\ \varepsilon\mu(y|x_i) & \text{otherwise} \end{cases}
\]

where \(\varepsilon \in [0,1]\) is a weight factor, and note that \(\sum_{y=1}^{Y} p'(y|r_i) = 1\). These new ground truths have been used in loss function instead of hot-encoding.

\[
L' = -\sum_{i=1}^{M \times N} \sum_{y=1}^{Y} (1 - \varepsilon)p(y|r_i) + \varepsilon\mu(y|r_i)] \log q_\theta(y|r_i)
\]

\[
L' = -\sum_{i=1}^{M \times N} \sum_{y=1}^{Y} [(1 - \varepsilon)p(y|r_i) + \varepsilon\mu(y|r_i)] \log q_\theta(y|r_i)
\]

\[
L' = \sum_{i=1}^{M \times N} \left\{ (1 - \varepsilon) \left[ -\sum_{y=1}^{Y} p(y|r_i) \log q_\theta(y|r_i) \right] + \varepsilon \left[ -\sum_{y=1}^{Y} \mu(y|x_i) \log q_\theta(y|r_i) \right] \right\}
\]

\[
L' = \sum_{i=1}^{M \times N} \left[ (1 - \varepsilon)H_i(p, q_\theta) + \varepsilon H_i(u, q_\theta) \right]
\]

One can observe that each ground truth, the loss contribution is a mixture of entropy between predicted distribution \((H_i(p, q_\theta))\) and the hot-encoding, and the entropy between the predicted distribution \((H_i(\mu, q_\theta))\) and the noise distribution. While training, \(H_i(p, q_\theta) = 0\) if the model learns to predict the distribution confidently, however, \(H_i(\mu, q_\theta)\) will increase dramatically. To overcome this phenomenon, we used a regularizer \(H_i(\mu, q_\theta)\) to prevent the model from predicting too confidently. In practice, \(\mu(y|r)\) is a uniform distribution that does not dependant on Hyperspectral Data. That is to say \(\mu(y|r) = \frac{1}{Y}\). The ultimate network is trained for 50 epochs using a mini-batch size of 256 without any batch normalization and data augmentation. In a nutshell, the details of 3D/2D convolutional layers and kernels are as follows: \(3D_{\text{conv layer}}_1 = 8 \times 5 \times 5 \times 7 \times 1\) i.e. \(K_1 = 5\), \(K_2 = 5\) and \(K_3 = 7\). \(3D_{\text{conv layer}}_2 = 16 \times 5 \times 5 \times 8\) i.e. \(K_1 = 5\), \(K_2 = 5\), \(K_3 = 5\). \(3D_{\text{conv layer}}_3 = 32 \times 3 \times 3 \times 16\) i.e. \(K_1 = 3\), \(K_2 = 3\) and \(K_3 = 3\). \(3D_{\text{conv layer}}_4 = 64 \times 3 \times 3 \times 32\) i.e. \(K_1 = 3\), \(K_2 = 3\) and \(K_3 = 3\). \(2D_{\text{conv layer}}_5 = 128 \times 3 \times 3 \times 64\) i.e. \(K_1 = 3\) and \(K_2 = 3\). Three 3D convolutional layers are employed to increase the number of spectral-spatial feature maps and one 2D convolutional layer is used to discriminate the spatial features within different spectral bands while preserving the spectral information. Initially, the weights are randomized and then optimized using back-propagation with the Adam optimizer by using the loss function presented in equation 9. Further details regarding the Hybrid CNN architecture in terms of types of layers, dimensions of output feature maps and number of trainable parameters can be found in [6].

### III. Experimental Settings and Results

The experiments have been conducted on two real HSI datasets, namely, Indian Pines (IP), and Pavia University (PU). These datasets are acquired by different sensors i.e. Reflective Optics System Imaging Spectrometer (ROSIS) and Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [6]. The experimental results explained in this work have been obtained through Google Colab [17] which is an online platform to execute any python environment while having a good internet speed to execute the code. Google Colab provides the option to execute many versions of python, Graphical Process Unit (GPU), upto 358+ GB of cloud storage, and upto 25 GB of Random Access Memory (RAM).

In all the experiments, the initial size of the train/validation/test sets is set to 25%/25%/50% to validate the proposed model as well as several other state-of-art-deep models. The comparative methods include AlexNet [18], LeNet [19], 2D CNN [20], and 3D CNN [21] models. Moreover, the following accuracy metrics have been conducted to validate the claims made in this manuscript. The accuracy metrics include Kappa \((\kappa)\)^2 Average (AA)\(^3\) and Overall (OA)\(^4\) All these metrics are computed using the following equations.

\[
Kappa (\kappa) = \frac{P_o - P_e}{1 - P_e}
\]

\[
P_e = P^+ + P^-
\]

\[
P^+ = \frac{TP + FN}{TP + FN + FP + TN} \times \frac{TP + FN}{TP + FN + FP + TN}
\]

\[
P^- = \frac{FN + TN}{TP + FN + FP + TN} \times \frac{FP + TN}{TP + FN + FP + TN}
\]

\[
P_o = \frac{TP + TN}{TP + FN + FP + TN}
\]

\[
\text{Overall (OA)} = \frac{1}{K} \sum_{i=1}^{K} TP_i
\]

where \(TP\) and \(FP\) are true and false positive, \(TN\) and \(FN\) are true and false negative, respectively. For fair comparison purposes, the learning rate for all these models including hybrid models is set to 0.001, Relu as the activation function

\(^1\kappa\) is known as a statistical metric that considered the mutual information regarding a strong agreement among classification and ground-truth maps

\(^2\text{AA represents the average class-wise classification performance}\)

\(^3\text{OA is computed as the number of correctly classified examples out of the total test examples}\)
for all layers except the output layer on which Softmax is used, patch size is set of 15, and for all the experiments, 15 most informative bands have been selected using principal component analysis to reduce the computational load. The convergence accuracy and loss of our proposed pipeline and all other competing methods for 50 epochs is presented in Figure 1. From loss and accuracy curves, one can conclude that the smoothing has faster convergence than the model without convergence.

(A) Accuracy (b) Loss

Fig. 1: Accuracy and Loss for Training and Validation sets on Indian Pines for 50 epochs.

### A. Indian Pines

Indian pines (IP) dataset is acquired using AVIRIS sensor over the northwestern Indiana test site. IP data consist of 2/3 agriculture, 1/3 forest, and other vegetation. Some of the water absorption bands are removed and the remaining 200 bands are used for the experimental process. This data is consists of 2/3 agriculture, 1/3 forest, and other vegetation. Less than 5% of total coverage consists of crops that are in an early stage of growth. Building, low-density housing, two dual-lane highway, small roads, and a railway line are also a part of it. Further details about the experimental datasets can be found at [22]. Table I and Figure 2 presents in-depth comparative accuracy analysis on the IP dataset.

### TABLE I: Indian Pines: Performance analysis of different state-of-the-art models. The higher accuracies are emphasised.

| Class          | Train/Yeast Test | 2D | 3D | AlexNet | LeNet | WoS | WS | WS |
|----------------|------------------|----|----|---------|------|----|----|----|
| Corn-needl     | 3527/3577/714    | 98.319 | 93.574 | 97.3389 | 97.0588 | 98.4593 | 98.3795 |
| Corn-needl     | 3527/3577/714    | 99.5180 | 66.7469 | 98.3132 | 99.5180 | 99.5180 |
| Corn           | 5959/5918 | 94.0677 | 90.6795 | 93.2203 | 99.1525 | 100 |
| Grass-pasture  | 1211/1210/242 | 93.3471 | 97.1070 | 98.2809 | 94.3148 | 95.0243 | 96.2809 |
| Grass-trees    | 1821/183/365 | 99.0041 | 97.5482 | 98.3651 | 99.0041 | 99.4250 | 99.7200 |
| Grass-mowed    | 77/74 | 92.8571 | 92.8571 | 92.8571 | 92.8571 | 92.8571 |
| Hay-windowed   | 1191/2629 | 100 | 100 | 100 | 100 | 100 |
| Oats           | 9510 | 100 | 100 | 100 | 100 | 100 |
| Soybean-needl  | 24/443/1046 | 95.5906 | 82.9455 | 93.2312 | 99.3277 | 97.8423 | 97.8423 |
| Soybean-needl  | 64/444/1228 | 99.6742 | 92.4267 | 97.8013 | 99.4815 | 97.0588 | 99.7200 |
| Soybean-silica | 146/146/307 | 97.8430 | 96.6541 | 96.8096 | 95.2661 | 93.3632 | 99.0041 |
| Wheat          | 515/162 | 90.1906 | 90.0392 | 100 | 99.0196 | 99.0196 | 99.0196 |
| Woods          | 316/317/633 | 99.8420 | 99.3680 | 99.5260 | 98.8941 | 99.9628 |
| Buildings      | 99/97/193 | 99.4818 | 90.6779 | 98.8795 | 99.4818 |
| Stone-Steel    | 232/2486 | 100 | 97.8260 | 100 | 97.8260 | 100 |

### TABLE II: Pavia University: Performance analysis of different state-of-the-art models. The higher accuracies are emphasised.

| Class            | Train/Yeast Test | 2D | 3D | AlexNet | LeNet | WoS | WS | WS |
|------------------|------------------|----|----|---------|------|----|----|----|
| 2D              | 100              | 100 | 100 | 100     | 100  | 100 |
| 3D              | 100              | 100 | 100 | 100     | 100  | 100 |
| AlexNet          | 99.0196          | 99.0196 | 99.0196 | 99.0196 | 99.0196 | 99.0196 |
| LeNet            | 99.0196          | 99.0196 | 99.0196 | 99.0196 | 99.0196 | 99.0196 |
| WoS (Hybrid net without Smoothing) | 99.9115 | 99.9115 | 99.9115 | 99.9115 | 99.9115 | 99.9115 |
| WS (with Smoothing) | 99.6283 | 99.6283 | 99.6283 | 99.6283 | 99.6283 | 99.6283 |

Fig. 3: Pavia University: Classification accuracy: Fig. 3a 2D-CNN = 99.9070%, Fig. 3b 3D CNN = 99.9256%, Fig. 3c AlexNet = 99.0768%, Fig. 3d LeNet = 99.9318%, Fig. 3e WoS (Hybrid net without Smoothing) = 99.9504%, and Fig. 3f WS (with Smoothing) = 99.9628%.

### B. Pavia University

Pavia University (PU) dataset acquired using Reflective Optics System Imaging Spectrometer (ROSIS) optical sensor over Pavia in northern Italy. PU dataset is distinguished into 9 different classes. PU consists of 610 × 610 spatial samples per spectral band and 103 spectral bands with a spatial resolution of 1.3m. Further details about the experimental datasets can be found at [22]. Table II and Figure 3 presents in-depth comparative accuracy analysis on the IP dataset.
extremely confidently which is not good from a generalization point of view. Therefore, this work proposed the use of an entropy-based regularization process to improve the generalization performance using soft labels. These soft labels are the weighted average of the hard labels and uniform distribution over entire ground truths. The entropy-based regularization process prevents CNN from becoming over-confident while learning and predicting thus improves the model calibration and bean-search. Extensive experiments have confirmed that the proposed pipeline outperformed several state-of-the-art methods.

REFERENCES

[1] A. Alcolea, M. E. Paoletti, J. M. Haut, J. Resano, and A. Plaza, “Inference in supervised spectral classifiers for on-board hyperspectral imaging: An overview,” Remote Sensing, vol. 12, no. 3, p. 534, 2020.

[2] H. Ayaz, M. Ahmad, A. Sohaib, M. N. Yasir, M. A. Zaidan, M. Ali, M. H. Khan, and Z. Saleem, “Myoglobin-based classification of minced meat using hyperspectral imaging,” Applied Sciences, vol. 10, no. 19, p. 6862, 2020.

[3] H. Ayaz, M. Ahmad, M. Mazzara, and A. Sohaib, “Hyperspectral imaging for minced meat classification using nonlinear deep features,” Applied Sciences, vol. 10, no. 21, p. 6862, 2020.

[4] M. H. Khan, S. Zainab, M. Ahmad, A. Sohaib, H. Ayaz, and M. Mazzara, “Hyperspectral imaging for color adulteration detection in red chili,” Applied Sciences, vol. 10, no. 17, p. 5955, 2020.

[5] Z. Saleem, M. H. Khan, M. Ahmad, A. Sohaib, H. Ayaz, and M. Mazzara, “Prediction of microbial spoilage and shelf-life of bakery products through hyperspectral imaging,” IEEE Access, vol. 8, pp. 176986–176996, 2020.

[6] M. Ahmad, S. Shabbir, R. A. Raza, M. Mazzara, S. Distefano, and A. M. Khan, “Hyperspectral image classification: Artifacts of dimension reduction on hybrid cnn,” arXiv preprint arXiv:2101.10532, 2021.

[7] J. Wang, R. Huang, S. Guo, L. Li, M. Zhu, S. Yang, and L. Jiao, “Nas-guided lightweight multiscale attention fusion network for hyperspectral image classification,” IEEE Transactions on Geoscience and Remote Sensing, pp. 1–14, 2021.

[8] S. Shabbir and M. Ahmad, “Hyperspectral image classification—traditional to deep models: A survey for future prospects,” arXiv preprint arXiv:2101.06116, 2021.

[9] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, “A survey: Deep learning for hyperspectral image classification with few labeled samples,” Neurocomputing, vol. 448, pp. 179–204, 2021. [Online]. Available: [https://www.sciencedirect.com/science/article/pii/S0925231221004033](https://www.sciencedirect.com/science/article/pii/S0925231221004033)

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning Representations by Back-Propagating Errors. Cambridge, MA, USA: MIT Press, 1988, p. 696–699.

[11] H. Sha, M. Al Hasan, and G. Mohler, “Learning network event sequences using long short-term memory and second-order statistic loss,” Stat. Anal. Data Min., vol. 14, no. 1, p. 61–73, Jan. 2021. [Online]. Available: [https://doi.org/10.1002/sam.11489](https://doi.org/10.1002/sam.11489)

[12] H. Bi, R. Santos-Rodriguez, and P. Flach, “Polarim image classification via robust low-rank feature extraction and markov random field,” in IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 708–711.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 06 2016.

[14] C. Zhang and M. Han, “Multi-feature hyperspectral image classification with 12,1 norm constrained joint sparse representation,” International Journal of Remote Sensing, vol. 42, no. 12, pp. 4789–4808, 2021. [Online]. Available: [https://doi.org/10.1080/01431161.2021.1899854](https://doi.org/10.1080/01431161.2021.1899854)

[15] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, “Regularized evolution for image classifier architecture search,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 4780–4789, Jul. 2019. [Online]. Available: [https://ojs.aaai.org/index.php/AAAI/article/view/4405](https://ojs.aaai.org/index.php/AAAI/article/view/4405)

[16] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning recognition,” CoRR, vol. abs/1707.07012, 2017. [Online]. Available: [http://arxiv.org/abs/1707.07012](http://arxiv.org/abs/1707.07012)

[17] T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. Reboucas Filho, “Performance analysis of google colaboratory as a tool for accelerating deep learning applications,” IEEE Access, vol. 6, pp. 61677–61685, 2018.

[18] S. Mei, X. Chen, Y. Zhang, J. Li, and A. Plaza, “Accelerating convolutional neural network-based hyperspectral image classification by step activation quantization,” IEEE Transactions on Geoscience and Remote Sensing, pp. 1–12, 2021.

[19] Y. Yuan, C. Wang, and Z. Jiang, “Proxy-based deep learning framework for spectral-spatial hyperspectral image classification: Efficient and robust,” IEEE Transactions on Geoscience and Remote Sensing, pp. 1–15, 2021.

[20] H.-C. Li, S.-S. Li, W.-S. Hu, J.-H. Feng, W.-W. Sun, and Q. Du, “Recurrent feedback convolutional neural network for hyperspectral image classification,” IEEE Geoscience and Remote Sensing Letters, pp. 1–5, 2021.

[21] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, “A fast and compact 3-d cnn for hyperspectral image classification,” IEEE Geoscience and Remote Sensing Letters, pp. 1–5, 2020.

[22] Hyperspectral Datasets Description, 2021 (accessed 2020-04-01), [http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes](http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes)