Land classification in satellite images by injecting traditional features to CNN models

Mehmet Çağrı Aksoy\textsuperscript{a}, Beril Sirmacek\textsuperscript{b} and Cem Ünsalan\textsuperscript{a}

\textsuperscript{a}Faculty of Engineering, Department of Electrical and Electronics Eng, Marmara University, Istanbul, Turkey; \textsuperscript{b}Independent scientist, Overijssel, The Netherlands

ABSTRACT

Deep learning methods have been successfully applied to remote-sensing problems for several years. Among these methods, CNN-based models have high accuracy in solving the land classification problem using satellite or aerial images. Although these models have high accuracy, this generally comes with large memory size requirements. However, it is desirable to have small-sized models for applications, such as the ones implemented on unmanned aerial vehicles, with low memory space. Unfortunately, small-sized CNN models do not provide as high accuracy as with their large-sized versions. In this study, we propose a novel method to improve the accuracy of CNN models, especially the ones with small size, by injecting traditional features into them. To test the effectiveness of the proposed method, we applied it to the CNN models SqueezeNet, MobileNetV2, ShuffleNetV2, VGG16 and ResNet50V2 having size 0.5 MB to 528 MB. We used the sample mean, grey-level co-occurrence matrix features, Hu moments, local binary patterns, histogram of oriented gradients and colour invariants as traditional features for injection. We tested the proposed method on the EuroSAT dataset to perform land classification. Our experimental results show that the proposed method significantly improves the land classification accuracy especially when applied to small-sized CNN models.

1. Introduction

Deep learning methods have proven themselves in recent years with their success in remote-sensing applications (Zhu et al. 2017; Ma et al. 2019; Zhai et al. 2015; Zhi et al. 2019). Among these, CNN-based models have high accuracy in solving the land classification problem using satellite or aerial images. On the other hand, traditional feature extraction methods were the only option before the deep learning era to solve such problems. Hence, they became mature in time. These two methods, deep learning–based and traditional, have been used separately most of the times up to now. There are also methods benefiting from both approaches in the literature.

When we consider successful deep learning–based CNN models, they are usually large due to their large number of parameters. Therefore, they require significant processing
power and memory space. Although this may not cause a problem in server-based or cloud applications, it limits the usage of such large-sized models on low-power embedded platforms such as the ones on UAV Osco et al. (2021). Several studies emphasize this limitation on object detection and land classification via UAV platforms (Ocer et al. 2020; Wang et al. 2018; Qiao et al. 2020). The main bottleneck for these systems is the computing power limits. Hence, the model size, complexity and data processing capacity of the hardware are of great importance, especially in real-time image processing tasks. Therefore, quantization, pruning, filter compression and matrix factorization are usually applied to the CNN model at hand Goel et al. (2020). The aim here is to reduce the complexity of the model such that it becomes suitable to be deployed on embedded low-power devices. However, the model performance and accuracy of the model generally decreases as a result of these operations.

In this study, we propose a novel method to increase the performance of CNN models, especially the ones with small size. For this purpose, our method injects traditional features into CNN models. The method is developed especially for low-complexity models deployed on embedded systems such as the ones on UAVs. Detailed literature review reveals that such a method has not yet been used in remote sensing as we proposed. The closest study was developed by Jbene, El Mali and El Hassouni (2019) in which statistical features are used to increase the performance of the CNN model. The authors did not report a significant performance increase with their method.

Next, we introduce the candidate CNN models used in this study. Then, we summarize the traditional feature extraction methods that we use in the proposed method. Afterwards, we explain the feature injection method we propose in detail. We next provide the performance of the proposed method on the EuroSAT dataset for land classification. Finally, we provide comments and ways of extending the proposed method.

2. Candidate CN models for land classification

We selected five candidate CNN models to be used in the proposed method: SqueezeNet, MobileNetV2, ShuffleNetV2, VGG16 and ResNet50V2. These five CNN models are the most frequently used ones in the literature. Besides, they proved themselves to be reliable in various applications. The chosen CNN models have different sizes. Hence, we can observe the effect of the proposed method on a diverse set of CNN models.

In this section, we briefly summarize the candidate CNN models. For more information on them, please check the provided references. While using the candidate CNN models, we apply transfer learning by freezing the top layer of each model with pre-trained ‘ImageNet’ weights. ImageNet is a dataset used for classification tasks (Deng et al. 2009). We provide the number of parameters and model size for these models at the end of the section.

2.1. SqueezeNet

The first CNN model we selected in this study is SqueezeNet proposed by Iandola et al. (2016). This model has a small size specifically developed to be used in embedded systems. Therefore, it represents the low end of the models available in the literature.
After freezing the top layer, we consider 512 SqueezeNet features (tensors) to be used in Section 4.

2.2. MobileNetV2

MobileNetV2 is the next CNN model considered in this study. It has been proposed by Sandler et al. (2018). As in SqueezeNet, it has a small model size such that it can be safely used on mobile and edge devices. After freezing the top layer, we consider 1280 MobileNetV2 features (tensors) to be used in Section 4.

2.3. ShuffleNetV2

ShuffleNetV2 is the third CNN model used in this study. It has been proposed by Zhang et al. (2017). ShuffleNetV2 has a similar number of parameters and model size as in MobileNetV2. Hence, it also represents a low model size option. After freezing the top layer, we consider 1024 ShuffleNetV2 features (tensors) to be used in Section 4.

2.4. VGG16

In addition to the small-sized models considered in previous sections, we also selected large models in this study. The aim of this selection is to observe the effect of the proposed feature injection method on them. Therefore, we selected the VGG16 model proposed by Simonyan and Zisserman (2014). After freezing the top layer, we consider 512 VGG16 features (tensors) to be used in Section 4.

2.5. ResNet50V2

ResNet50V2 is the fifth and final CNN model considered in this study. It has been proposed by He et al. (2016). The reason for choosing this model was to see how the largest model that we have selected would behave in the proposed method. After freezing the top layer, we consider 2048 ResNet50V2 features (tensors) to be used in Section 4.

2.6. Summary of the candidate CNN models

We summarize the CNN models considered in the previous section next. The aim of this paper is to compare these models in terms of their number of parameters and model size. Therefore, we tabulate these values in Table 1. As can be seen in this table, the selected CNN models have size between 0.5 MB and 528 MB. The number of parameters in these models is within the range of 729 K–23.5 M. We will use these models while injecting traditional features into them.

3. Traditional features for land classification

Our aim in this study is to improve the performance of candidate CNN models by injecting traditional features into them. Therefore, we summarize the traditional feature extraction
methods in this section. Therefore, here we selected the most well-known and commonly used feature extraction methods. Hence, the reader can find sample codes for them easily.

### 3.1. Sample mean

The first traditional feature extraction method used in this study is the sample mean. The mean value can be considered as a first-order statistical feature extraction method that can be used to summarize information in colour bands. As a result, we obtain the average (sample mean) normalized value for each band in the colour image at hand. Normalization refers to re-scaling real-value numbers into a 0–1 range. As a result, we extract three features based on the sample mean.

### 3.2. Grey level co-occurrence matrix features

Another traditional feature extraction method considered in this study is based on the grey-level co-occurrence matrix (GLCM) proposed by Haralick, Shanmugam and Hak Dinstein (1973) for texture analysis. This method has been generalized to other problems, including land use classification in satellite images. Haralick et al. introduced several second-order statistical features based on GLCM such as correlation, contrast, homogeneity, energy and ASM matrix features calculated on the greyscale image at hand. In this study, we use them as the second set of traditional features.

### 3.3. Hu moments

The third traditional feature extraction method used in this study is moments proposed by Hu (1962). In this study, we calculate seven such features obtained from the greyscale image at hand.

### 3.4. Local binary patterns

The fourth traditional feature extraction method used in this study is the local binary patterns proposed by Ojala, Pietikainen and Harwood (1996). This method was initially developed for texture analysis. As in other methods, it has been applied to land use classification studies. We extracted LBP 64 features from the greyscale image at hand.
3.5. **Histogram of oriented gradients**

The fifth traditional feature extraction method used in this study is the histogram of oriented gradients (HOG) proposed by Dalal and Triggs (2005). It has been widely used in computer vision to solve object detection problems. HOG counts the occurrence of gradient orientations of a portion of the image. After using the greyscale data in the HOG method, we collected HOG values created for a single colour band in a one-dimensional array, as we did in LBP. As a result, we obtain 64 features for the image at hand.

3.6. **Colour invariants**

The sixth and final traditional feature extraction method used in this study is based on colour information. We extract it via colour invariants introduced by Geusebroek et al. (2001). As a result, we obtain 64 features for the image at hand.

4. **Proposed feature injection method for land classification**

We can divide a CNN model into two parts: feature extraction and classification. The feature extraction part extracts features from a given source via successive filtering and nonlinear operations. Here, filter coefficients are learned via training. Hence, an adaptive structure can be formed for the image set at hand. Thanks to the transfer learning method that we are using, we do not learn filter coefficients every time. Instead, we learn them for the ImageNet dataset once and keep these coefficients as fixed. Mostly, the classification part is composed of a fully connected neural network (FCNN).

We provide a visual representation of the proposed feature injection method in Figure 1. As can be seen in this figure, the CNN feature extraction part provides its own features. Traditional feature extraction methods extract their own features. Then, these two feature sets are merged with the concatenation operation. Hence, the concatenate layer takes both inputs and returns a single vector as merging of all inputs just before the classification step of the CNN model. In this way, we inject the traditional features into the CNN model.

The concatenated traditional and CNN features are fed to the FCNN for the land use classification operation. We used max pooling and batch normalization operations just before concatenating the traditional features. We have 10 classes in our problem. Hence, the FCNN has 10 outputs. We train the parameters of the classification part (FCNN) of each CNN model for the problem at hand. While doing so, we also adjust the weights of the

![Figure 1](image-url)

*Figure 1.* Visual representation of the proposed feature injection method.
injected traditional features fed to the FCNN. Hence, the overall model is reconfigured to solve the land classification problem at hand.

5. Experiments

In this section, we conducted experiments to measure the performance of the proposed method. The body of the methods are written in Python language to run on a Jupyter notebook. The hardware power comes from the Google Colaboratory environment. We used Keras with the TensorFlow framework in implementation. During training, the epoch number is set to 16 with batch size 64. In the optimizer side, Adam is used in training. Each test has been repeated five times, and average values are listed in the following sections. The test code block can be accessed from GitHub by Aksoy (2022).

5.1. Dataset used in experiments

We tested the proposed method on the EuroSAT dataset introduced by Helber et al. (2019) as a benchmark for land cover classification tasks. This dataset consists of 27,000 images acquired by the Sentinel-2 satellite. The Sentinel-2 satellite imagery is open and freely available on the Earth observation program Copernicus. Spatial resolution of these images is 10 metres such that detailed object shape information, such as sharp corners and straight edges, is not available in the images. Therefore, CNN models developed for separate object detection may not be suitable for this dataset. We overcome this short-coming by applying traditional feature extraction methods developed for summarizing general information in a given window.

The EuroSAT dataset holds labels with information of 10 different classes: annual crop, forest, herbaceous vegetation, highway, industrial, pasture, permanent crop, residential, river and sea lake. The original dataset has 13 different spectral band images. In this study, we only used the RGB image bands. We took 80% of the dataset for training and the rest is used for testing. Hence, we have 21,600 and 5400 images for training and testing, respectively.

5.2. The effect of injecting traditional features into the SqueezeNet model

In order to measure the performance of our proposed method, we first used the SqueezeNet model as a base. We injected traditional feature sets into the model and tabulated the obtained results in Table 2. As can be seen in this table, the improvement in accuracy reaches 8.4% when all traditional features are injected into the SqueezeNet model.

| Test Scenario                        | Maximum Accuracy |
|--------------------------------------|------------------|
| All traditional features             | 0.5934           |
| SqueezeNet                           | 0.6778           |
| SqueezeNet + all traditional features| **0.7618**       |

Bold values indicate the best performance in these tables.
5.3. The effect of injecting traditional features into the MobileNetV2 model

We next used the MobileNetV2 model as the base to measure the performance of the proposed method. As in the previous section, we injected traditional feature sets into the model and tabulated the obtained results in Table 3. As can be seen in this table, the improvement in accuracy reaches 2.27% when all the traditional features are injected into the MobileNetV2 model.

| Test Scenario                               | Maximum Accuracy |
|---------------------------------------------|------------------|
| All traditional features                    | 0.5934           |
| MobileNetV2                                 | 0.6062           |
| MobileNetV2 + all traditional features      | 0.6289           |

5.4. The effect of injecting traditional features into the ShuffleNetV2 model

Third, we used the ShuffleNetV2 model as the base to measure the performance of the proposed method. As in the previous section, we injected traditional feature sets into the model and tabulated the obtained results in Table 4. As can be seen in this table, the improvement in accuracy reaches 4.96% when all traditional features are injected into the ShuffleNetV2 model.

| Test Scenario                               | Maximum Accuracy |
|---------------------------------------------|------------------|
| All traditional features                    | 0.5934           |
| ShuffleNetV2                                | 0.8502           |
| ShuffleNetV2 + all traditional features     | 0.8998           |

5.5. The effect of injecting traditional features into the VGG16 model

Next, we used the VGG16 model as a base in order to measure the performance of our proposed method. As in the previous section, we injected traditional feature sets into the model and tabulated the obtained results in Table 5. As can be seen in this table, the improvement in accuracy is only 0.0014% when all traditional features are injected to the VGG16 model.

| Test Scenario                               | Maximum Accuracy |
|---------------------------------------------|------------------|
| All traditional features                    | 0.5934           |
| VGG16                                       | 0.9091           |
| VGG16 + all traditional features            | 0.9105           |
Table 6. The effect of injecting traditional features to the ResNet50V2 model.

| Test Scenario                      | Maximum Accuracy |
|------------------------------------|------------------|
| All traditional features           | 0.5934           |
| ResNet50V2                         | 0.6720           |
| ResNet50V2 + All traditional Features | **0.6878**       |

5.6. The effect of injecting traditional features into the ResNet50V2 model

Finally, we used the ResNet50V2 model as a base in order to measure the performance of our proposed method. As in the previous section, we injected traditional feature sets into the model and tabulated the obtained results in Table 6. As can be seen in this table, the improvement in accuracy is only 0.0158% when all traditional features are injected to the ResNet50V2 model.

5.7. Overview of the injection performance

We can summarize all the experiments performed in the previous section as follows. When the CNN model size is small, injecting traditional features may increase the accuracy significantly as in the SqueezeNet, MobileNetV2 and ShuffleNetV2 models. Among these, improvements in the SqueezeNet model reaches up to 8.4%, which is a significant value. Therefore, SqueezeNet is a good candidate for the proposed method. The main reason for this is that, SqueezeNet cannot grasp the characteristics of the EuroSAT images due to its low parameter size. Injecting traditional features to it boosts the performance. On the other hand, injected traditional features should grasp the information within the image, which is not acquired by the CNN model. Therefore, the most suitable traditional features to be used in the proposed method should be selected by keeping this condition in mind.

Note here that injecting traditional features into the CNN model leads to a size increase of 66 KB. This addition can be accepted when the corresponding improvement in accuracy is considered. On the other hand, the improvement in accuracy for large-sized models such as VGG16 and ResNet50V2 is negligible. One reason for this result is that, these large models extract almost all information from the image due to their excessive parameters. Hence, injecting traditional features into such models increases accuracy marginally. We summarize the accuracy improvements for the selected models in Figure 2. This figure further justifies our observations that the accuracy of relatively simple CNN models can be improved significantly by injecting traditional features into them.

6. Final comments

In this study, we propose a novel method to improve the accuracy of CNN models, especially the ones with small size, by injecting traditional features into them. To test the effectiveness of the proposed method, we applied it to the CNN models SqueezeNet, MobileNetV2, ShuffleNetV2, VGG16 and ResNet50V2 having size 0.5 MB to 528 MB. We used the sample mean, grey-level co-occurrence matrix features, Hu moments, local binary patterns, histogram of oriented gradients and colour invariants as traditional features for injection. We tested the proposed method on the EuroSAT dataset, consisting of 10 land use classes, to...
perform land classification. We observed that, the improvement in land classification accuracy reaches up to 8.4% and 4.96% when all traditional features are injected into the SqueezeNet and ShuffleNetV2 models, respectively. Hence, we can deduce that the proposed method significantly improves the land classification accuracy when applied to the small-sized CNN models. This is the strength of the proposed method. On the other hand, such improvements in accuracy cannot be reached when the proposed method is applied to large-sized models. This is the shortcoming of the proposed method. One reason for this result is that, these large models extract almost all information from the image due to their excessive parameters. Hence, injecting traditional features into such models increases accuracy only marginally. An increase in accuracy by the proposed method is achieved by adding only 350 extra traditional features. Hence, we can claim that the proposed method can be helpful in solving the land classification problem on edge (low power and low memory) devices. In a future study, the reader can check the effect of feature selection step on the proposed method. Another future study may be on analysing the effect of the proposed method on decreasing the size of the training data set in CNN models.

**Disclosure statement**

No potential conflict of interest was reported by the authors.

**Funding**

This work is supported by TUBITAK under project no 221N393 and under the umbrella of ERA-NET Cofund ForestValue project ForestMap.
References

Aksoy, M. C. 2022. “Land Classification in Satellite Images by Injecting Traditional Features to CNN Models.” https://github.com/mcagriaksoy/Land-Classification-in-Satellite-Images-by-Injecting-Traditional-Features-to-CNN-Models.

Dalal, N., and B. Triggs. 2005. “Histograms of Oriented Gradients for Human Detection.” In IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, Vol. 1, 886–893. doi:10.1109/CVPR.2005.177.

Deng, J., W. Dong, R. Socher, L. Li-Jia, L. Kai, and L. Fei-Fei. 2009. “ImageNet: A Large-Scale Hierarchical Image Database.” In 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 248–255. doi:10.1109/CVPR.2009.5206848.

Geusebroek, J. M., R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. 2001. “Color Invariance.” IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (12): 1338–1350. doi:10.1109/34.977559.

Goel, A., C. Tung, Y. H. Lu, and G. Thiruvathukal. 2020. “A Survey of Methods for Low-Power Deep Learning and Computer Vision.” 2020 IEEE 6th World Forum on Internet of Things(WF-IoT), New Orleans, LA, USA, 2020, pp. 1–6. doi:10.1109/WF-IoT48130.2020.9221198.

Haralick, R. M., K. Shanmugam, and I. Hak Dinstein. 1973. “Textural Features for Image Classification.” IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (6): 610–621. doi:10.1109/TSMC.1973.4309314.

Helber, P., B. Bischke, A. Dengel, and D. Borth. 2019. “EuroSat: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification.” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12 (7): 2217–2226. doi:10.1109/JSTARS.2019.2918242.

He, K., Zhang, X., Ren, S., Sun, J. (2016). Identity Mappings in Deep Residual Networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. Lecture Notes in Computer Science, vol 9908. Springer, Cham. https://doi.org/10.1007/978-3-319-46493-0_38

Hu, M. K. 1962. “Visual Pattern Recognition by Moment Invariants.” IEEE Transactions on Information Theory 8 (2): 179–187. doi:10.1109/TIT.1962.1057692.

Iandola, F., S. Han, M. Moskewicz, K. Ashraf, W. Dally, and K. Keutzer. 2016. “SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5mb Model Size”. doi:10.48550/arxiv.1602.07360.

M. Jbene, A. D. el Mali and M. El Hassouni, “Fusion of Convolutional Neural Network and Statistical Features for Texture classification,” 2019. International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, 2019, pp. 1–4, doi:10.1109/WINCOM47513.2019.8942469.

Ma, L., Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson. 2019. “Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review.” Isprs Journal of Photogrammetry and Remote Sensing 152: 166–177. doi:10.1016/j.isprsjprs.2019.04.015.

Ocer, N. E., G. Kaplan, F. Erdem, D. Kucuk Matci, and U. Avdan. 2020. “Tree Extraction from Multi-Scale UAV Images Using Mask R-CNN with FPN.” Remote Sensing Letters 11 (9): 847–856. doi:10.1080/2150704X.2020.1784491.

Ojala, T., M. Pietikainen, and D. Harwood. 1996. “A Comparative Study of Texture Measures with Classification Based on Featured Distributions.” Pattern recognition 29 (1): 51–59. doi:10.1016/0031-3203(95)00067-4.

Osco, L. P., J. M. Junior, A. P. M. Ramos, L. A. C. Jorge, S. N. Fatholahi, J. A. Silva, E. T. Matsubara, H. Pistori, W. N. Goncalves, and J. Li. 2021. “A Review on Deep Learning in UAV Remote Sensing.” International Journal of Applied Earth Observation and Geoinformation 102: 102456. doi:10.1016/j.jag.2021.102456.

Qiao, R., A. Ghodsi, W. Honggan, Y. Chang, and C. Wang. 2020. “Simple Weakly Supervised Deep Learning Pipeline for Detecting Individual Red-Attacked Trees in VHR Remote Sensing Images.” Remote Sensing Letters 11 (7): 650–658. doi:10.1080/2150704X.2020.1752410.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. 2018. “MobileNetv2: Inverted Residuals and Linear Bottlenecks.”.
Simonyan, K., and A. Zisserman. 2014. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. doi:10.48550/ARXIV.1409.1556.

Wang, Q., X. Zhang, G. Chen, F. Dai, Y. Gong, and K. Zhu. 2018. “Change Detection Based on Faster R-CNN for High-Resolution Remote Sensing Images.” Remote Sensing Letters 9 (10): 923–932. doi:10.1080/2150704X.2018.1492172.

X. Zhang, X. Zhou, M. Lin and J. Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 6848-6856, doi:10.1109/CVPR.2018.00716.

Zhai, L., H. Sang, Y. Gao, A. Fang, Y. Zhang, and S. Wei. 2015. “A New Approach for Mapping Regional Land Cover and the Application of This Approach in Australia.” Remote Sensing Letters 6 (4): 267–275. doi:10.1080/2150704X.2015.1029089.

Zhi, L., Y. Xuchu, B. Liu, and X. Wei. 2019. “A Dense Convolutional Neural Network for Hyperspectral Image Classification.” Remote Sensing Letters 10 (1): 59–66. doi:10.1080/2150704X.2018.1526424.

Zhu, X. X., D. Tuia, L. Mou, G. S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. 2017. “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources.” IEEE Geoscience and Remote Sensing Magazine 5 (4): 8–36. doi:10.1109/MGRS.2017.2762307.