ANALYTICAL STUDY ON CNN BASED REMOTE SENSING DATA CLASSIFICATION

SAMRITI HEER

Research Scholar, Dept. of Computer Science,

Sri Satya Sai University of Technology & Medical Sciences,

Sehore, Bhopal-Indore Road, MadhyaPradesh, India

Dr. Jitendra Sheetlani

Research Guide, Dept. of Computer Science,

Sri Satya Sai University of Technology & Medical Sciences,

Sehore, Bhopal Indore Road,Madhya Pradesh, India

ABSTRACT

Remote-sensing image scene classification can provide significant value, ranging from forest fire monitoring to land-use and land-cover classification. Beginning with the first aerial photographs of the early 20th century to the satellite imagery of today, the amount of remote-sensing data has increased geometrically with a higher resolution. The need to analyze these modern digital data motivated research to accelerate remote-sensing image classification. Fortunately, great advances have been made by the computer vision community to classify natural images or photographs taken with an ordinary camera. Curiously, we find transfer learning from models trained on larger, more generic natural images datasets outperformed transfer learning from models trained directly on smaller remotely sensed datasets. Nonetheless, results show that transfer learning provides a powerful tool for remote-sensing scene classification.

KEYWORDS:CNN, Learning, Remote,Sensing,Image, Classification, learning, classification

INTRODUCTION

Conventional scene classification techniques rely on low-level visual features to represent the images of interest. Such low-level features can be global or local. Global features are extracted from the entire remote-sensing image, such as color (spectral) features, texture features, and shape features. Local features, like scale invariant feature transform (SIFT) are extracted from image patches that are centered about a point of interest. The main reason remote-sensing scene classification only marginally improved is due to the fact that the approaches relying on low-level features are incapable of generating sufficiently powerful feature representations for remote-sensing scenes. Despite CNNs' powerful feature extraction capabilities, Hu and others found that in practice it is difficult to train CNNs with small datasets. These general-specific CNN layer feature transitions lead to the development of transfer learning. In transfer learning, the filters learned by a CNN model on a primary task are applied to an unrelated secondary task. The primary CNN model can be used at as feature extractor, or as a starting point for a secondary CNN model.

DEEP LEARNING IN REMOTE SENSING

Remote sensing of images has been successfully applied in many fields, such as classification and change detection. However, remote-sensing image processing involves a few preprocessing procedures in addition to classification and change detection; furthermore, it is highly dependent on the method that is applied. Hence, the remote-sensing community is always committed to developing remote-sensing methods for improving the performance of aspects, such as preprocessing, segmentation, and classification. Neural networks, the basis of deep learning (DL) algorithms, have been used in the remote sensing community for many years.

LITERATURE REVIEW

Kurt Marfurt (2020) Natural image datasets can range up to millions of samples and are, therefore, amenable to deep-learning techniques. Many fields of science, remote sensing included, were able to exploit the success of natural image classification by convolutional neural network models using a technique commonly called transfer learning. We provide a systematic review of transfer learning application for scene classification using different datasets and different deep-learning models. We evaluate how the specialization of convolutional neural network models affects the transfer learning process by splitting original models in different points.

Daldegan, G.A.; Roberts, D.A.; Ribeiro, F.D (2019) Fire is used worldwide to clear natural vegetation areas for economic activities and to manage the regeneration of already opened sites. In Brazil, fire has been traditionally used to convert natural vegetation areas to agricultural lands (slash and burn) and to manage pastures for livestock. We developed the Burned Area Spectral Mixture Analysis (BASMA) algorithm in Google Earth Engine, which is designed to process Landsat data to produce a multi-temporal fire scar database representing annual burned area for an extent of 362,000 km2 in the transition zone between the Amazon forest and the Cerrado biome.

Liu, Y., Zhong, Y., Fei, F., Zhu, Q., & Qin, Q. (2018) We present an analysis of three possible strategies for exploiting the power of existing convolutional neural networks (ConvNets or CNNs) in different scenarios from the ones they were trained: full training, fine tuning, and using ConvNets as feature extractors. In many applications, especially including remote sensing, it is not feasible to fully design and train a new ConvNet, as this usually requires a considerable amount of labeled data and demands high computational costs

Zeng, D., Chen, S., Chen, B., & Li, S. (2018)recently, many researchers have been dedicated to using convolutional neural networks (CNNs) to extract global-context features (GCFs) for remote-sensing scene classification. Commonly, accurate classification of scenes requires knowledge about both the global context and local objects. However, unlike the natural images in which the objects cover most of the image, objects in remote-sensing images are generally small and decentralized. Thus, it is hard for vanilla CNNs to focus on both global context and small local objects

Zhu, X. X., Tuia, D., Mou, L. C., et al. (2017) Central to the looming paradigm shift toward data-intensive science, machine-learning techniques is becoming increasingly important. In particular, deep learning has proven tobe both a major breakthrough and an extremely powerful tool in many fields. Shall weembrace deep learning as the key to everything? Or should we resist a black-box solution?These are controversial issues within the remote-sensing community.

METHODS

This section provides some details about the datasets we use in our experiments as well as the number of samples for each one of the datasets. We use a 70%–10%–20% split between training, validation, and test sets.

To better understand the effects of different approaches and techniques used for transfer learning with remote-sensing datasets, we perform two major experiments using the models presented.

- **Model Split**

To evaluate the transfer learning process from natural images to remote sensing datasets, we use VGG19 and Inception V3 models and train a small classification network on top of such models. We refer to the original CNN model structure, part of VGG19 or part of Inception V3, as the “base model”, and the small classification network as the “top model

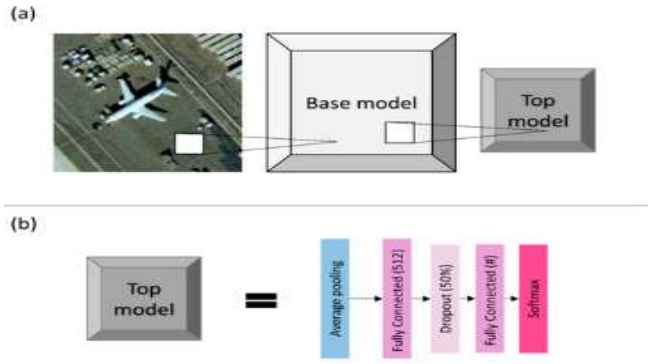

Figure 1.1: Visualization of the models used.

Figure 1.1 (**a**) Shows a sample image from UCMerced, the base model, and the top model.

(**b**) Provides more details for the top model.

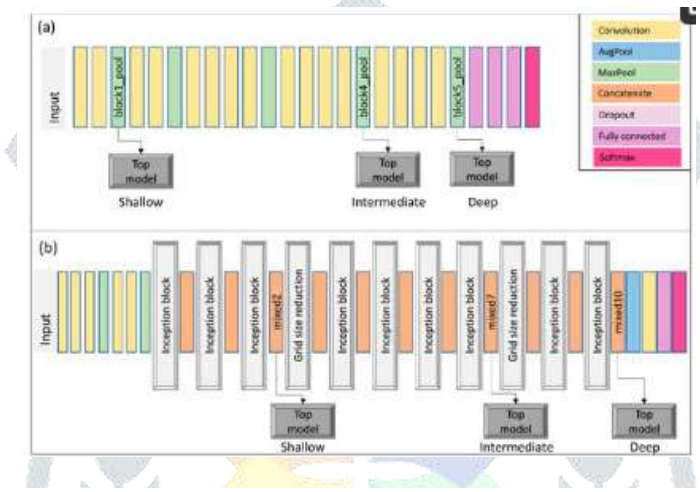

Figure 1.2: Visual representation of the models used.

In both panels, data flows from left to right. Both panes use the same color code for layer representation.

Figure 1.2 (**a**) Shows the VGG19 shallow, intermediate, and deep models – based on the naming convention we are using. (**b**) Shows the Inception V3 shallow, intermediate, and deep models. For easier reference, we wrote the layer names (as implemented in Keras) for each one of the layers we used to split the original CNN models. Note for each one of the depth levels (shallow, intermediate, deep), we simple use the model up to the detour and connect it with our top model (e.g., when training VGG19 shallow, the data goes through two convolutional layers, one max pooling layers, and exits into our top model). Please refer to Simonyan and Zisserman and Szegedy for details on VGG19 and Inception V3, respectively.

- **Stochastic Gradient Descent versus Adaptive Optimization Methods**

In the search for the global minima, optimization algorithms frequently use the gradient descent strategy. To compute the gradient of the loss function, we sum the error of each sample. Using our PatternNet data split as example, we first loop through all training set containing 21,280 samples before updating the gradient. Therefore, to move a single step towards the minima, we compute the error 21,280 times. A common approach to avoid computing the error for all training samples before moving a step is to use stochastic gradient descent (SGD).

- **Stochastic Gradient Descent versus Adaptive Optimization Methods**

We train the shallow, intermediate, and deep VGG19 and Inception V3 models using the UCMerced dataset with different optimizers

Figure 1.3 shows the accuracy on the test set obtained by each one of the models, with each one of the optimizers

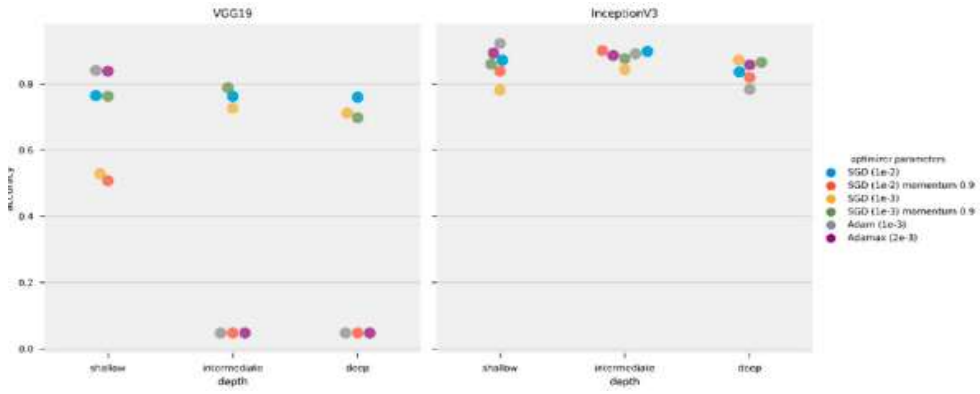

Figure 1.3: Test set accuracy obtained by the models using different optimizers training on the same UCMerced dataset.

The left panel shows VGG 19 results, right panel shows Inception V3 results.

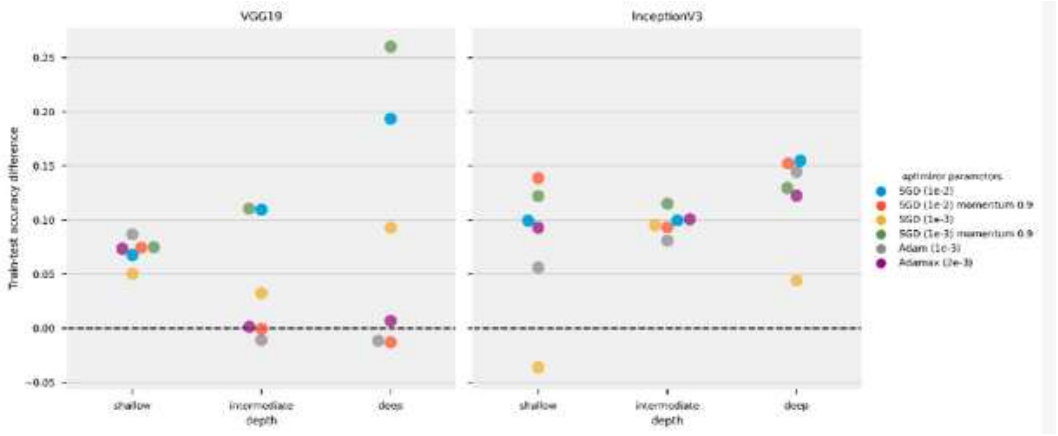

Figure 1.4: deep models remained stuck on local minima.

That SGD (1e-2), Adam (1e-2), and Adamax (2e-3)results of theVGG19 intermediate and deep models remained stuck on local minima.Figure 1.4 shows the difference in accuracy between the training set and the test set.

Table 1.1: Naming convention and optimizer details

Name	Optimizer Details
Stochastic gradient descent, SGD (1e-2)	SGD optimizer with learning rate of 0.01
SGD (1e-2) momentum 0.9	SGD optimizer with learning rate of 0.01 and momentum 0.9
SGD (1e-3)	SGD optimizer with learning rate of 0.001
SGD (1e-3) momentum 0.9	SGD optimizer with learning rate of 0.001 and momentum 0.9
Adam (1e-2)	Adam optimizer with learning rate of 0.01 and default parameters as described in [51]
Adamax (2e-3)	Adamax optimizer with learning rate of 0.02 and default parameters as described in [51]

- **General to Specific Layer Transition of CNN Models**

This section shows the results of transfer learning; both feature extraction and fine-tuning modes, as well as training the models with randomly initialized weights.

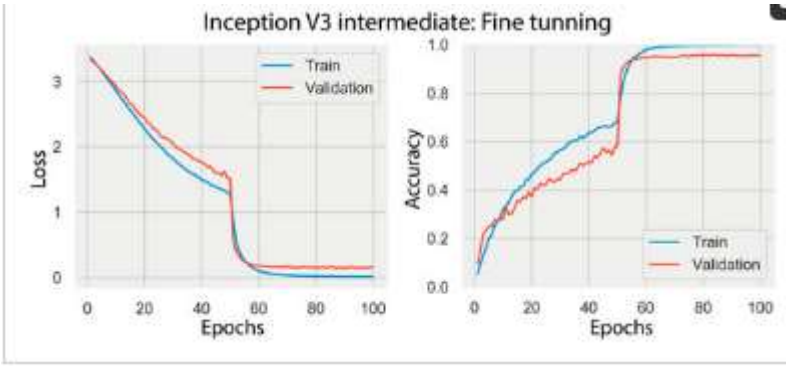

Figure 1.5: Train and validation loss and accuracy for the Inception V3 intermediate in the fine tuning mode trained on the AID dataset using SGD (1e-3) momentum 0.9.

Figure 1.6 shows the training and validation loss and accuracy through the training epoch's shows the correspondent confusion matrix computed on the AID test set.

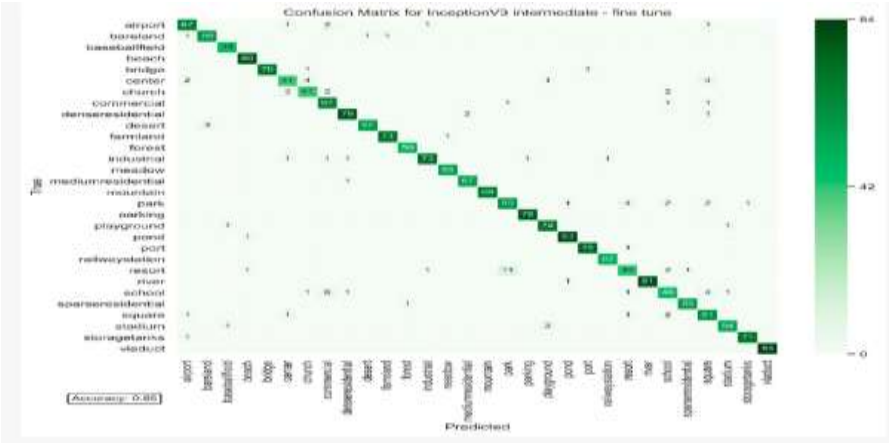

Figure 1.6: Confusion matrix for the test set of AID dataset for the Inception V3 intermediate in the fine-tune mode using SGD (1e-3) momentum 0.9

Table 1.2: Best test set accuracy for Inception V3 and VGG19 version for each dataset using SGD (1e-3) momentum 0.9optimizer

Table 7. Best test set accuracy for Inception V3 and VGG19 version for each dataset using SGD (1e-3) momentum 0.9 optimizer

Dataset	Model	Depth	Mode	Accuracy
PatternNet	Inception V3	intermediate	fine tune	0.997
	VGG19	deep	fine tune	0.995
AID	Inception V3	intermediate	fine tune	0.950
	VGG19	deep	fine tune	0.936
UCMerced	Inception V3	intermediate	fine tune	0.983
	VGG19	deep	fine tune	0.981

RESULT AND DISCUSSION

Using ImageNet data, Yosinski found that transfer learning, even when applied to a secondary task not similar to the primary task; perform better than training CNN models with randomly initialized weights. Using medical image data, Tajbakhsh found that fine-tuning achieved results comparable to or better than results from training a CNN model with randomly initialized weights. Our results align with their findings. Both Figure 8 and Table 7 show the fine-tuning mode of training outperforming randomly initialized weights when using SGD (1e-3) momentum 0.9. Show that transfer learning performs best with the Adamax (2e-3) optimizer. However, it seems that the step size (2e-3) is too large for fine tuning in the VGG19 model, such that the VGG19 intermediate and deep models trained on fine tune and randomly initialized weights modes fall in local minima. The primary task (ImageNet composed of natural images) is not very similar to the secondary task (remote-sensing scene classification). While there is a similarity in primary and secondary tasks datasets, such as the number of channels (red-green-blue components), images

are from the visible spectra, and some objects might be present in both tasks (e.g., airplanes), the tasks are fundamentally different.

CONCLUSION

Our objective with this paper was to investigate the use of transfer learning in the analysis of remote-sensing data, as well as how the CNN performance depends on the depth of the network and on the amount of training data available. Our experiments, based on three distinct remote-sensing datasets and two popular CNN models, show that transfer learning, specifically fine tuning CNNs is a powerful tool for remote-sensing scene classification. Much like the findings in other experiments, the results we found show that transfer from natural images (ImageNet) to remote-sensing imagery is possible. Despite the relatively large difference between primary and secondary tasks, transfer learning training mode generally outperformed training a CNN with randomly initialized weights and achieved the best results overall. Curiously, fine-tuning models primarily trained on the generic ImageNet dataset overperformed fine-tuning models primarily trained on PatternNet dataset. As expected, our results also indicate that for a particular application, the amount of training data available plays a significant role on the performance of the CNN models.

REFERENCES

1. Zhu, Q., Zhong, Y., Liu, Y., Zhang, L., & Li, D. (2018). A deep-local-global feature fusion framework for high spatial resolution imagery scene classification. Remote Sensing, 10, 568.

2. Zhu, X. X., Tuia, D., Mou, L. C., et al. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine, 5, 8–36.

3. Zhong, Y., Fei, F., & Zhang, L. (2016). Large patch convolutional neural networks for the scene classification of high spatial resolution imagery. Journal of Applied Remote Sensing, 10, 025006.

4. Zou, Q., Ni, L., Zhang, T., & Wang, Q. (2015). Deep learning based feature selection for remote sensing scene classification. IEEE Geoscience & Remote Sensing Letters, 12, 2321–2325.

5. Chen, Y., Fan, R. S., Wang, J. X., Wu, Z. L., & Sun, R. X. (2017). High resolution Image Classification method combining with minimum noise fraction rotation and convolution neural network. Laser & Optoelectronics Progress, 54, 434–439.

6. Cheng, G., & Han, J. (2016). A survey on object detection in optical remote sensing images. ISPRS Journal of Photogrammetric and Remote Sensing, 117, 11–28.

7. He, H. Q., Du, J., Chen, T., & Chen, X. Y. (2017). Remote sensing image water body extraction combing NDWI with convolutional neural network. Remote Sensing Information, 32, 82–86.

8. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceeding IEEE Conference Computer Vision and Pattern Recognition, Las Vegas, NV

9. Li, Y., Zhang, H., Xue, X., Jiang, Y., & Shen, Q. (2018). Deep learning for remote sensing image classification: A survey. WIREs Data Mining and Knowledge Discovery, 8(6), [e1264]. https://doi.org/10.1002/widm.1264

10. Hu, F., Xia, G. S., Wang, Z., Huang, X., Zhang, L., & Sun, H. (2017). Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(5), 2015–2030