Issues in Information Systems
Volume 25, Issue 2, pp. 267-276, 2024

DOI: https://doi.org/10.48009/2_iis_2024_122

Land use classification of satellite images with convolutional neural networks (CNNs)

Ice Asortse, *Robert Morris University, inast139@mail.rmu.edu*
John C. Stewart, *Robert Morris University, stewartj@rmu.edu*
G. Alan Davis, *Robert Morris University, davisl@rmu.edu*

Abstract

Deep-learning neural networks continue to improve in effectiveness and capability. Specifically, Convolutional Neural Networks (CNNs) have proven to offer highly accurate results in differentiating graphical images. Using satellite imagery, CNNs have been highly successful in the classification of land use. This successful classification, however, depends on the model structure, data components, and classification methods utilized. While alternative approaches exist, the complexity and sophistication of such classification methods impose increased computational costs. Effective manipulation of CNN components is shown to offer computational savings and accurate classification of land use in satellite images.

Keywords: artificial intelligence, machine learning, deep learning, convolutional neural networks, satellite imaging, land use classification

Introduction

The combination of high-quality image technology and improvements in deep learning algorithms has led to the potential for more reliable graphic image classification. In order to accurately classify a specific image, computer vision relies on analysis at the pixel-level (Long, et al., 2015). This classification capability has wide applications within remote sensing, including applications in environmental and natural resource management (Boon, et al., 2016; Laliberte, et al., 2007). The capabilities of Deep Learning Networks have advanced tremendously in recent years, creating an entirely new discipline of Artificial Intelligence (AI) study and research. Specifically, Deep Convolutional Neural Networks (CNNs) have had particular success within image identification applications (Zeng, 2018).

Remote sensors, drones (i.e., unmanned aircraft), and satellite-generated images can be used by researchers and land managers to accurate classify and monitor changes in land cover types. These capabilities allow for high resolution images that can be fed into a CNN model for land use classification. For example, the determination of wetlands, or wetland delineation, may be accomplished via aerial image classification methodologies (Rezaee, et al., 2018). In recent decades, such vital wetland ecosystems have been degraded and lost at an alarming rate. The Fish and Wildlife Service, National Oceanic and the Atmospheric Administration’s (NOAA) National Marine Fisheries Service (NMFS) reported a net loss of roughly 361,000 acres of coastal wetlands in the eastern United States in the years 1998 through 2004. This significant loss of wetlands averages 59,000 acres per year (Stedman & Dahl, 2008). The dynamic nature of coastal wetland creates numerous environmental benefits. These benefits include, support of diverse wildlife and plant habitats, the minimization of erosion effects from storms, and the natural filtering of pollutants and excessive nutrients.

Loss of wetlands is not the only environmental problem that can be evaluated via image analysis. Deforestation is also a universal and critical problem for our planet. As with wetland analysis, image classification can be used to determine levels of forest degradation on a global scale.

Managing or mitigating habitat loss requires a comprehensive knowledge of different types of land cover. This knowledge, in combination with computer vision using CNNs, can allow high resolution images to be properly and accurately analyzed. The analysis and comparison of image changes over time can be useful for documentation, and also as the catalyst for policy changes (Hsieh, et al., 2001)

Several studies have evaluated the use of deep learning neural networks (DLNN) for land cover mapping and identification of specific land uses and features. These studies have looked at the classification of images, such as forests, lakes, grasslands, etc. (Storie & Henry, 2018; Mahdianpari, et al., 2018; Yoshida & Omatu, 1994).

Research Question

The objective of the current research study was to develop a deep learning Convolutional Neural Network model to determine the accuracy of predicting land use classifications of satellite images using pixel-based evaluation. The research question is: "Can a simplified CNN be created to accurately differentiate various terrestrial and water images using readily available TensorFlow components and the computer power associated with a laptop computer?" To evaluate deep CNN models and to address this question, this study used common performance metrics. In addition, predicted classifications for the analyzed images were compared to the actual classifications to determine classification accuracy.

Literature Review

Convolutional Neural Networks (CNNs) were first proposed decades ago. However, advances in recent years have increased their capability and their potential for use in a wide variety of applications (Lecun, et al., 1998). These potential uses include the classification of images by land use, in order to determine changes over time, or to determine the extent and degree of habitat loss, etc. (Martins, 2020). CNNs have a number of advantages over traditional, shallower neural network models. CNNs use convolutional operations on the image pixels in order to extract features from the images. Unlike many other traditional neural network models, CNNs are specifically modeled to identify shapes that are two-dimensional.

Generally, a CNN is comprised of convolutional and pooling layers that alternate in the initial layers of the model. These layers are followed by the fully-connected layer. An output layer produces the classification. The convolutional layer is the basic layer in a CNN model and works systematically on a small area at a time with a filter. Finally, an activation function takes the output of the model's convolutional layer and generates a feature map. The feature map can then become the input for subsequent layers of the model.

A pooling layer follows the convolutional layer. One of the pooling layer's functions is dimension reduction. The pooling layer decreases the spatial dimensionality of the feature map, which decreases the computation requirements of the model. The feature maps are then flattened into one-dimension before they become input to the fully-connected layer. The fully-connected layer then maps the features into a linearly-distinguishable space, which is synchronized with the output layer. As with other neural network models, the output layer performs the classification using a function. The Softmax function (Liu, et al., 2016) and Support Vector Machines (SVM) are the most often used classification functions in CNNs.

Activation functions are added after any convolutional or fully-connected layer. Activation functions allow for the introduction of non-linearity, subsequent to the linear operations in the convolutional layer. An activation function acts as a circuit switch, determining if a neuron should fire or not. To make this determination, the activation function uses the weighted sum of the input to the neuron. The most common activation functions are Sigmoid, Rectified Linear Unit (ReLU) (Hara, Saito, & Shouno, 2015), Leaky ReLU, and Maxout functions (Goodfellow, Warde-Farley, Mirza, Courville, & Bengio, 2013). Each of the aforementioned activation functions has its advantages and disadvantages, which are beyond the scope of this paper.

A loss function (also called a cost or objective function) determines the difference between the model's predicted values and the actual values. Backpropagation (Hecht-Nielsen, 1989) is used to train the network. Initial errors between predicted and actual values are backpropagated through the network and weights are adjusted accordingly. As with other neural networks, large differences result in large adjustments to the weights, and small differences result in small adjustments to the weights. This process continues iteratively until the loss is minimized and there is a convergence.

In recent years, CNNs have proven very successful when applied to Remote Sensing (RS) classification of images. RS images inherently contain more abundant spectrum information and spatial information than traditional images in the red, green, and blue (RGB) bands. This increase of information in RS images results in a more robust characterization of image structure, shape, and consistency.

Several studies have focused on the use of different types of images, as well as a wider use of spectrum information and texture data to achieve higher classification accuracy (Xu, et al., 2016; Xia, et al.,2017; Xu, et al., 2018; Zhang, et al., 2018). For example, Li, et al. used *dropout* layers to reduce the parameters in the fully-connected layer. This technique allows the training data to be reduced, and results in a 95% classification accuracy (Li, et al., 2017). Zhong, et al., initiated the use of a global average pooling layer. This technique allows the average of the activations to be calculated in each feature map, thereby reducing the dimensionality of the spatial. Using this approach, the authors reported accuracies between 90% and 90.5% on three different datasets (Zhong et al., 2016).

While Softmax and SVM classifiers are the most common classifiers used in CNNs, the Softmax classifier has proven to have a limitation when applied to remote sensing data. Remote sensing data have comparatively extensive feature differences and elevated noise levels. A few past studies achieved high accuracies with variations on the SVM classifier (Bai, et al., 2018; Xu, et al., 2016). An ongoing issue with loss functions is their difficulty in the differentiation of feature components in the same class. Several attempts to rectify this issue have been explored with regularization (Chang, et al., 2016; Cheng, et al., 2016). This issue has also been addressed by using a function to minimize within-class distances and to concurrently maximize between-class separability (Li, et al., 2018).

Methodology

The objective of this current study is to build a convolutional neural network (CNN) model that can accurately classify land images. To train and test the model, the EuroSAT dataset was used. The Eurosat dataset is based on Sentinel-2 satellite images. These images cover 13 spectral bands and consist of 10 classes with 27,000 labels and geo-referenced samples (Eurosat, 2018, 2019). To derive these data, remote sensors onboard satellites detect solar radiation reflected from the Earth's surface. The amount of radiation reflected is determined by the radiation wavelength and the specific properties of the target's surface. Remote sensing methods use, as a standard, wavelengths in the 0.5 – 3.0 µm range.

The images in the dataset consisted of 10 different classifications of land cover types. The 10 classes of outcomes in the target variable were based on land use specifications with the following designations: Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River, and Sea Lake. The target variable then had 10 possible outcomes for the model to distinguish for each image (see Appendix 1).

The data were split with 80% of the images in the dataset used for training, and the remaining 20% used for testing. Therefore, 21,600 images were used for training and 5,400 were used for testing. A CNN model was developed using the TensorFlow Python Library. This neural network model consisted of four convolutional layers and five max pooling layers. Dropout layers were used to minimize overfitting of the data. The activation function was 'relu' and the optimizer was 'adam' (see Appendix 2). The model was fit over 10 epochs, with a batch size of 16.

Results

The deep learning CNN model that was developed resulted in an accuracy of nearly 88% at predicting the land cover outcomes on the test data. The loss and accuracy results for the training and test data can be seen in Table 1.

Table 1: Loss Accuracy for Training and Test Data

	Training Data	Testing Data
Accuracy	0.8552	0.8783
Loss	0.4238	0.3422

The plots of model accuracy and loss for the training and testing data are shown in Figures 1 and 2, respectively. These two figures show correlated and comparable outputs for both training and testing loss, as well as training and testing classification accuracy. These results indicate that no overfitting of the training data has occurred.

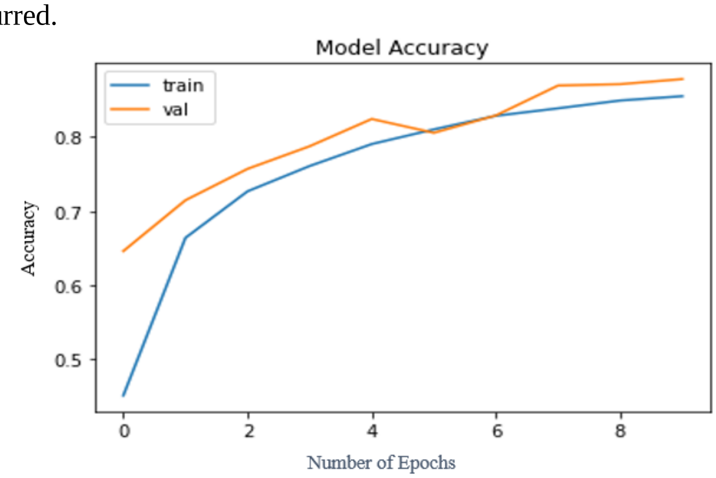

Figure 1: Model Accuracy for Training and Testing Data

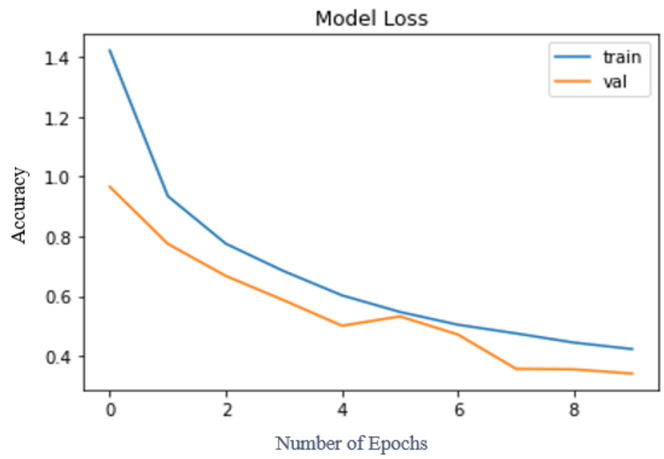

Figure 2: Model Loss for Training and Testing Data

As seen in these results, a CNN model can be developed using the basic building blocks inherent in the Tensorflow open-source machine learning library. Further, the resulting CNN model can provide significant accuracy in determining the land use type on a dataset of satellite images. As discussed previously, more sophisticated techniques exist for spectral analysis and image classification. However, the current study shows that when employing a less complex and minimally computational technique (i.e., using dropouts) a CNN model can produce highly accurate results.

Discussion

A number of factors affecting model results are related to the CNN structure, including the classification process, the spectrum explored, and the richness of the image data. However, the model used in the current study was able to show that high accuracy is possible with a CNN model that uses a traditional activation function and classifier. Land cover categories are often not easily delineated with clearly defined edges. Obviously, these uncertainties at the pixel level can greatly affect a model's classification accuracy. Also, the issue of spatial autocorrelation affecting classification accuracy can result from differences in adjacent pixels. This difference is not considered in classical analysis, but can be overcome with additional computation (Blaschke, 2010).

A classification problem arises with a pixel-based analysis of natural areas. A typical approach in CNN is to utilize only spectral information, as opposed to a spatial approach. In a spatial approach, a group or cluster of pixels is evaluated together and identified as an object. At least one past study found a significantly increased accuracy from an object-based versus a pixel-based model. This difference in accuracy is likely due to a greater within-class similarity versus the high between-class variability inherent in natural terrestrial images (Myint, et al., 2011; Blaschke, 2010; Dronova, et al., 2011).

Conclusion

An assumption in classic remote sensing classification methods is that each pixel is independent, and proximal pixels are not considered in the analysis. Using object-based classification, geographic objects become the focus of image analysis. As shown in past studies, consideration of both spectral and object image features has proven superior to solely pixel-based analysis (Myint, et al., 2011).

Again, these more sophisticated and computationally expensive analyses can be undertaken. However, it can be concluded from the model and results of the current study that relatively high classification accuracy can be achieved with CNN models that rely on pixel-based analysis. The current study also shows that careful model development and use of dropouts, can reduce performance issues and increase average results (Piotrowski, et al., 2020; Avudaiammal, et al., 2023; Peng, et al., 2024).

The application of CNNs to new and diverse use cases will undoubtedly continue. In terms of land use classification using CNNs, countless opportunities exist for future research. As demonstrated in the current research study, future studies involving CNNs could explore model structures, choices in data components, and/or classification methods. An awareness of computationally-costly CNN structures should be a consideration in the development of CNN models to address land use classification

References

Avudaiammal, R., Rajangam, V., Swarnalatha, A., Nancy, P.S., & Pavithra, S. (2023). Investigation on Performance of CNN Architectures for Land Use Classification. In: Subhashini, N., Ezra, M.A.G., Liaw, SK. (eds) Futuristic Communication and Network Technologies. Lecture Notes in Electrical Engineering, vol 966. Springer, Singapore. https://doi.org/10.1007/978-981-19-8338-2_21.

Bai, Y., Jiang, D. M., Pei, J. J., Zhang, N., & Bai, Y. (2018). Application of an improved ELUconvolution neural network in the SAR image ship detection. Bulletin of Surveying and Mapping, (1), 125–128.

Blaschke, T. (2010). Object based image analysis for remote sensing. ISPRS J. Photogramm. Remote Sens. (65), 2–16. [CrossRef]

Boon, M., Greenfield, R., & Testamichael, S. (2016). Wetland assessment using unmanned aerial vehicle (UAV) photogrammetry. In Proceedings of the XXIII ISPRS Congress, Prague, Czech Republic, 12–19 July 2016.

Chang, L., Deng, X. M., Zhou, M. Q., Wu, Z. K., Yuan, Y., Yang, S., & Wang, H. A. (2016). Convolutional neural networks in image understanding. Acta Automatica Sinica,42, 1300–1312.

Cheng, G. Zhou, P. & Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. IEEE Transactions on Geoscience and Remote Sensing. 54(12), 7405-7415. doi: 10.1109/TGRS.2016.2601622.

Dronova, I., Gong, P., & Wang, L. (2011). Object-based analysis and change detection of major wetland cover types and their classification uncertainty during the low water period at Poyang Lake, China. Remote Sens. Environ. (115), 3220–3236. [CrossRef]

Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. Patrick Helber, Benjamin Bischke, Andreas Dengel, Damian Borth. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks.arXiv preprint arXiv.

Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A. & Bengio, Y. (2013). Maxout Networks. Proceedings of the 30th International Conference on Machine Learning, in Proceedings of Machine Learning Research. 28(3), 1319-1327 Available from https://proceedings.mlr.press/v28/goodfellow13.html

Hara, K., Saito, D., & Shouno, H. (2015). Analysis of function of rectified linear unit used in deep learning. International Joint Conference on Neural Networks, Killarney, Ireland.

Hecht-Nielsen, R. (1989). Theory of the backpropagation neural network. Neural Networks. IJCNN. International Joint Conference, Washington, DC.

Hsieh, P.F., Lee, L.C., & Chen, N.Y. (2001). Effect of spatial resolution on classification errors of pure and mixed pixels in remote sensing. IEEE Trans. Geosci. Remote Sens. (39), 2657–2663. [CrossRef]

Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. Patrick Helber, Benjamin Bischke, Andreas Dengel. 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018.

Laliberte, A., Rango, A., & Herrick, Jeffrey. (2007). Unmanned aerial vehicles for rangeland mapping and monitoring: A comparison of two systems. Conference: Proceedings of the ASPRS Annual Conference, Tampa, FL, USA.

Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient based learning applied to document recognition, Proceedings of the IEEE, 86 (11), 2278–2324.

Li, H., Fu, K., Xu, G., Zheng, X., Ren, W., & Sun, X. (2017). Scene classification in remote sensing images using a two-stage neural network ensemble model. Remote Sensing Letters. (8), 557–566.

Li, J. W., Qu, C. W., & Peng, S. J. (2018). Ship classification for unbalanced SAR dataset based on convolutional neural network. Journal of Applied Remote Sensing. 12(3), 035010.

Liu, W., Wen, Y., Yu, Z., & Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. ICML, 2(3), 7.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; 3431–3440.

Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. Remote Sens. (10), 1119. https://doi.org/10.3390/rs10071119

Martins, V. (2020). Deep Learning for Land Cover Classification and Environmental Analysis Using High-Resolution Remote Sensing Data, 1–149.

Myint, S.W., Gober, P., Brazel, A., Grossman-Clarke, S., & Weng, Q. (2011). Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. Remote Sens. Environ. (115), 1145–1161. [CrossRef]

Peng, M., Liu, Y., Khan, A., Ahmed, B., Sarker, S., Yasin, Y., Uzair Aslam Bhatti, G., Al-Razgan, M., & Ali,Y. (2024). Crop monitoring using remote sensing land use and land change data: Comparative analysis of deep learning methods using pre-trained CNN models, Big Data Research, (36), 100448 ISSN 2214-5796, https://doi.org/10.1016/j.bdr.2024.100448 .

Piotrowski, A., Napiorkowski, J., & Piotrowska, A. (2020). Impact of deep learning-based dropout on shallow neural networks applied to stream temperature modeling. Earth-Science Reviews, (201), 103076, ISSN 0012-8252, https://doi.org/10.1016/j.earscirev.2019.103076.

Rezaee, M., Mahdianpari, M., Zhang, Y., & Salehi, B. (2018). Deep convolutional neural network for complex wetland classification using optical remote sensing imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(9), 3030-3039.

Stedman, S.M., Dahl, & T.E. Status, (20024). Trends of wetlands in the coastal watersheds of the eastern United States, 1998 to 2004. 2008. Available online: https://www.fws.gov/wetlands/Documents/StatusaTrends-of-Wetlands-in-the-Coastal-Watersheds-of-the-Eastern-United-States-1998-to-2004.pdf

Storie, C. & Henry, C. (2018). Deep learning neural networks for land use land cover mapping, IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 2018, 3445-3448, doi: 10.1109/IGARSS.2018.8518619.

Xia, M., Cao, G., Wang, G. Y., & Shang, Y. F. (2017). Remote sensing image classification based on deep learning and conditional random fields. Journal of Image and Graphics. (22), 1289–1301.

Xu, S. H., Mu, X. D., Zhao, P., & Ma, J. (2016). Scene classification of remote sensing image based on multi-scale feature and deep neural network. Acta Geodaetica Et Cartographica Sinica. (45), 834–840.

Xu, Y., Wu, L., Xie, Z., & Chen, Z. (2018). Building extraction in very high-resolution remote sensing imagery using deep learning and guided filters. Remote Sensing. (10), 144.

Yi, Zheng. (2018). Evaluation and implementation of convolutional neural networks in image recognition. Journal of Physics: Conference Series. (1087), 062018. 10.1088/1742-6596/1087/6/062018.

Yoshida, T., & Omatu, S. (1994). Neural network approach to land cover mapping. IEEE Transactions on Geoscience and Remote Sensing, 32(5), 1103-1109.

Zhang, Y. H., Xia, G. H., Han, X., He, J., Ge, T. T., & Wang, J. G. (2018). Road extraction from multi-source high resolution remote sensing image based on fully convolutional network. 2018International Conference on Audio, Language and Image Processing (ICALIP). 201–204,IEEE, Shanghai, China.

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.

Appendix 1: Land Use Images

Appendix 2: Structure of the CNN Model

```
# Build the CNN Model

tf.random.set_seed(2)
cnn = Sequential()
cnn.add(Conv2D(32,
kernel_size = (3, 3),

input_shape = (64, 64, 3),

padding = 'same',

activation = 'relu'))
# Add a layer
cnn.add(MaxPooling2D(2,2))
cnn.add(Conv2D(32,

kernel_size = (3,3),

padding = 'same',

activation = 'relu'))
# Add a layer
cnn.add(MaxPooling2D(2,2))
cnn.add(Conv2D(32,

kernel_size = (3,3),

padding = 'same',

activation = 'relu'))
# Add a layer
cnn.add(MaxPooling2D(2,2))
cnn.add(Conv2D(64,

kernel_size = (3,3),

padding = 'same',

activation = 'relu'))
# Add a layer with dropout to prevent overfitting
cnn.add(MaxPooling2D(2,2))
cnn.add(Dropout(0.5))
cnn.add(Flatten())
cnn.add((Dense(128, activation = 'relu')))
cnn.add(Dropout(0.5))
cnn.add(Dense(13, activation = 'softmax'))

# Compile the model..
cnn.compile(optimizer = 'adam',
```