Ensemble Encoder–Decoder Models for Predicting Land Transformation

Pariya Pourmohammadi, *Member, IEEE*, Michael P. Strager, and Donald A. Adjeroh, *Member, IEEE*

***Abstract*—Land development is a dynamic and complex process influenced by a system of interconnected driving variables. Predicting such a process is important in mitigating severe climate situations and improving the resiliency of communities. Current predictive models in land transformation have not paid a serious attention to capturing and exploiting the interchannel relationships. Moreover, these models often have problems with generalization, which results in poor performance during testing. In this study, we use a novel multichannel data cube, constructed from socioeconomic attributes, terrain characteristics, and landscape traits, to predict land transformation in a watershed in the US. In particular, we introduce methods for projecting impervious land transformations using these data cubes, using 2-D and 3-D convolutional neural networks (CNNs) and their ensembles. We apply fusion at decision, score, and feature levels to improve the generalization ability and robustness of the proposed predictive models. Performance is assessed using the Dice coefficient, receiver operating characteristic curves, data visualization, and running time. Our study shows that the use of 2-D and 3-D CNN ensembles improved the performance of the models in terms of model stability, precision and recall, and Dice coefficient.**

***Index Terms*—Convolutional neural networks (CNNs), developed land expansion, evidence fusion, land transformation prediction.**

I. Introduction

LAND development has substantial impacts on different aspects of both the local and global environmental conditions. It can place a significant burden on the resources of a given region [1]. The impact of developed and urbanized lands on the global ecosystem and climate change has been widely discussed [2]. In the past few years, the influence of urban footprint in the creation of urban heat island and the global temperature has been a ubiquitous discussion among environmental scientists. Furthermore, because of their population density, developed regions are vulnerable in the event of natural catastrophes, and this is a matter of concern for risk management modelers [3]. Accordingly, accurate models for projecting the complex process of land development [4]–[6] are utilized in natural catastrophe risk modeling, and in management and climate change modeling [3], [7], [8].

Improving prediction outcomes for such complex and dynamic processes requires models to learn the complexities of land transformation and land development processes. Recent advancements in machine learning and deep learning have led to improvements in predictive land transformation models (LTMs) [6], [9]–[12]. Neural networks, as universal function approximators, have been applied in land change detection (CD) and prediction, land cover classification, and object detection [8], [13], [14]. In this work, our focus is on land change prediction. We use the terms "land transformation" and "land change" interchangeably to refer to land cover change. These terms have been used previously in the literature [8] and for similar problems. The proposed model uses the historic land cover data to project these changes. We are interested in predicting impervious land, which, according to the National Land Cover Dataset (NLCD) of the US, represents urban impervious land cover, and the regions with human made structures [15]. The process of impervious land transformation is a complex and dynamic procedure influenced by numerous underlying mechanisms [16], [17]. We use an ensemble approach to integrate features and/or results from multiple models that are developed in the prediction of the land transformation process.

Multispectral data cubes are used to model land transformation using both 2-D and 3-D convolutional neural networks (CNNs); then, we incorporate ensemble classification methods to establish stable and generalized models. Our results show how these techniques enhance the model performance in predicting the land cover transformation. The results of previously proposed models of *deepLandU, deepLandS*, and *U-Net* [11], [18] are used as the baseline models. A major shortcoming of the baseline models, as suggested in [18], is that these models do not incorporate interchannel relationships. In addition, there is need for these models to exhibit better generalization performance [19]. Our suggested framework, based on ensemble models, considers this drawback to improve model performance. A key contribution of this work is the use of a novel 3-D data cube to represent various types of data related to land transformation.

Manuscript received May 31, 2021; revised September 20, 2021; accepted October 6, 2021. Date of publication October 19, 2021; date of current version November 19, 2021. This work was supported in part by the National Science Foundation under Cooperative Agreement number OIA-1458952, in part by the USDA National Institute of Food and Agriculture through Hatch Project 1015648, in part by the West Virginia Agricultural and Forestry Experiment Station Scientific Article number 3415, and in part by the National Science Foundation under Grant ACI-1548562. *(Corresponding author: Pariya Pourmohammadi.)*

Pariya Pourmohammadi is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA, and also with the Sierra Nevada Research Institute, University of California, Merced, CA 95343 USA (e-mail: papourmohammadi@mix.wvu.edu).

Michael P. Strager is with the School of Natural Resources, Davis College of Agriculture, Natural Resources and Design, West Virginia University, Morgantown, WV 26506 USA (e-mail: mstrager@wvu.edu).

Donald A. Adjeroh is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA (e-mail: don@csee.wvu.edu).

Digital Object Identifier 10.1109/JSTARS.2021.3120659

Another is the use of 2-D and 3-D CNN models to exploit potential interchannel relationships in this data representation. A third contribution is the ensemble model built on these 2-D and 3-D CNN models for improved generalization capability. Our results show the significant improvement in model performance in terms of model robustness and accuracy using the proposed 3-D convolutions, ensemble learning, and multilevel evidence fusion.

More specifically, we incorporate a third dimension in the CNN models for analyzing land transformations. The formulated models do not necessarily result in higher model performance; however, they provide a compelling ground for incorporating the intervariable relationships in the models. One more implication of this contribution is that these models create a schema for the datasets with higher temporal dimensions, which opens up new perspectives for further improvements in predictive models for land cover. Furthermore, we show how structuring ensemble models based on evidence fusion at both the score and label decision levels improve the overall model performance. This serves to balance the tradeoff between model bias and variance. Thus, another contribution of this work is the introduction of an ensemble method for improved robustness in land change prediction.

The rest of this article is organized as follows. In Section II, we describe related efforts on machine learning models and the application of CNNs in predicting land cover transformation. Section III presents the data and the baseline models, 3-D convolutional networks, evidence fusion and ensemble methods, and the applied metrics for performance measurement. Section IV presents the results. Section V concludes this article.

II. Related Work

CNNs are a group of deep models, in which an order of increasingly complex features is generated. These groups of features are basically the outcome of sequences of trainable filters and poolings in a defined window. In CNNs, predefined kernels are used for convolving the features; predefined windows are also used for poolings [20].

The past few years have seen rapid developments in deep CNNs, with various applications in different disciplines [20], [21]. These methods have become very popular in image analysis and classification. One of the major advances in this area was a large deep CNN (AlexNet) proposed by Krizhevsky *et al.* [22], which was able to classify samples from a large dataset of high-resolution images. The dominance of accuracy of these models in classification problems has led to the growing application of CNNs.

Deep CNNs have also been widely applied in many areas other than computer vision, for instance, in genomics sequence studies, healthcare, speech recognition, geoscience, and earth science [18], [23]–[26]. In remote sensing, various architectures of neural networks models have been deployed to perform object detection, CD [27], land cover classification, subpixel mapping for land use applications [19], [28], and semantic segmentation of the land [29], [30]. Shi *et al.* [27] used convolutional block attention modules and applied a deeply supervised attention metric-based network from CD on the CD dataset of Sun Yat-Sen University. The CNN has also been used in developing frameworks for subpixel mapping to extract information using super-resolution networks [19], [28]. U-Net, ResNet, and AlexNet are some of the popular CNN models that have been used for land classification [31], [32]. Hu *et al.* [13] and Castelluccio *et al.* [14] used AlexNet, VGG (from Visual Geometry Group), Caffe, and PlaceNet to label patches of land images. Kampffmeyer *et al.* [29] and Sherrah [30] applied the CNN and a fully connected network to perform patch-based and pixelwise segmentation. Kampffmeyer *et al.* [29] used these methods to perform segmentation of true orthophoto images into six classes of impervious surfaces, building, low vegetation, tree, car, and clutter/background.

Although the CNN has been widely applied for different tasks in land data analysis, there are limited applications of deep learning in land change prediction. The LTM is one of the well-known models that were developed to project land change prediction; this model was based on a multilayer perceptron network [8], [10]. The accuracy of this model depends on the input of the number of transformed cells (PIDs), which is a factor of population. Pourmohammadi *et al.* [18] introduced deepLandU and deepLandS models and showed that these models outperform the LTM. deepLandU and deepLandS are shallow 2-D networks that are used for predicting the land transformation. The number of filters in the convolutional layers is different in these models. deepLandU has less trainable filters and less poolings. The deepLandS model has a larger number of filters, one more maxPooling and upsampling layer compared to deepLandU, and also encompasses a dense layer. Results in [18] demonstrated that although deepLandU is a shallower network, it outperforms deepLandS. In both deepLand models, the first two convolutional layers include 64 filters followed by average pooling, and convolutional layers in the second and third parts include 128 and 256 filters, respectively. The convolving filters that were used in the first two convolutional layers were 5×5 kernels. In the first pooling layer, these networks applied average pooling, while the remaining pooling layers were based on max pooling.

The 2-D CNN is the prevalent model in most image-based applications of the CNN. Yu *et al.* [26] used 2-D convolutional layers to extract features in the spatial domain and constructed a nonlinear architecture, where interspectral relationships are fed into the model. They used their proposed model to classify hyperspectral satellite image datasets. They further broadened the scope of their work by introducing a 3-D neural network, where the third dimension was deployed to ingrate interchannel relationships [26]. Aside from the data manipulation, for some applications, 3-D CNNs have shown improvements in the model output. For instance, Ji *et al.* [20] proposed an application of 3-D CNNs to extract spatiotemporal features from multiple frames. They expanded the structure of their proposed network by exploring the model performance on variations of 3-D networks. Tran *et al.* [33] trained their so-called C3D CNNs for the purpose of action recognition. They studied variations of convolutional kernel cubes and showed that a CNN of $3 \times 3 \times 3$ kernels outperformed the other models in their experiments.

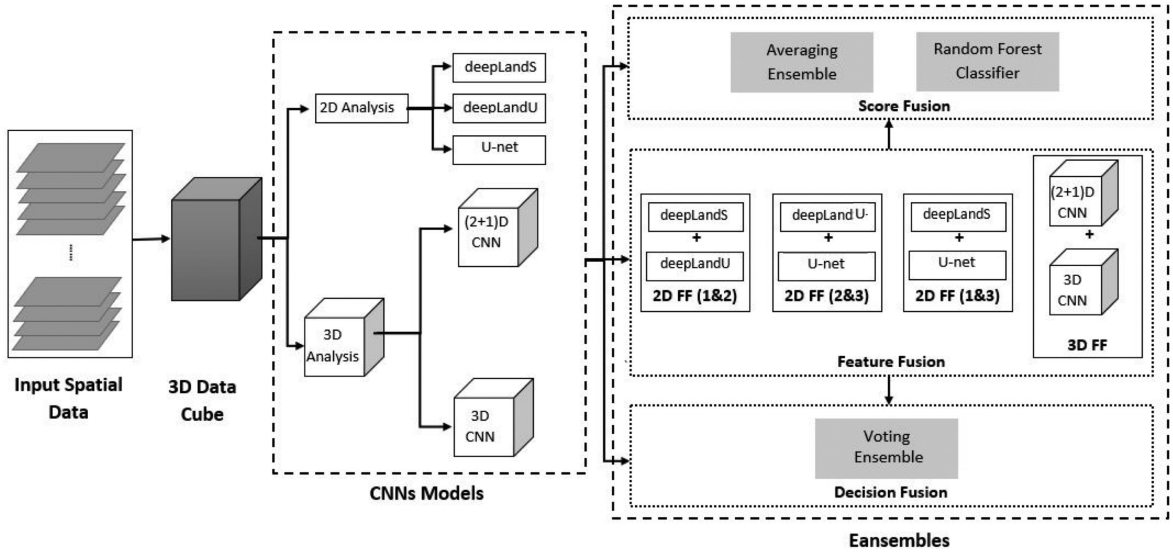


Fig. 1. Proposed framework for data representation, spatiospectral data analysis, and evidence fusion for prediction of impervious/developed land expansion.

In this work, we apply variation of 3-D CNN models on the spatiospectral data cubes to predict land development expansion. We fuse different model structures to develop a predictive deep learning model for land cover changes. Then, we integrate different methods to improve the model generalization. These methods include stacking the network weights in the training process, creating linear and nonlinear combinations of model scores, and deploying a voting ensemble on the model output. We use deepLandU, deepLandS, and U-net as the baseline models in this study.

III. Methods

The basic approach used in this work is that the proposed models will take data from a historical state of land in small patches, represented as a multichannel data cube, as their input. The output of each model is then a binary classification of the land cover in each patch. For each patch, our classification of interest is whether the patch belongs to developed land or to undeveloped land. Fig. 1 presents a general schematic diagram for the research methods. This framework first captures how various types of information about the land cover are represented in a multichannel spatiotemporal data cube. Then, the data cube is passed to a deep learning framework for classification at the pixel level, where the classification result implies whether each cell will go through land development. The results are further combined using various fusion techniques. In this section, we described our data representation in Section III-A. The model structures used in this study are explained in Section III-B. We present our approach to evidence fusion and performance measurement metrics in Sections III-C and III-D, respectively.

A. Database and Data Representation

The study area for this research is the Monongahela watershed (HUC6-050200) that is located along the ridge of the Appalachian Mountains in the U.S. (see Fig. 2). The data in this research are structured in such a way that each land characteristic is considered as a different image channel. We used multiple representations of landscape characteristics, geopolitical boundaries, and demographic traits in the feature class. To make a proper representation of these data attributes, the raster maps are formatted, georeferenced, and then coregistered as a multispectral data cube. A cross section of data cubes (which we call patches) with dimensions of $32 \times 32 \times 71$ was generated from the study area.[1] Any of these patches is then considered as one data sample.

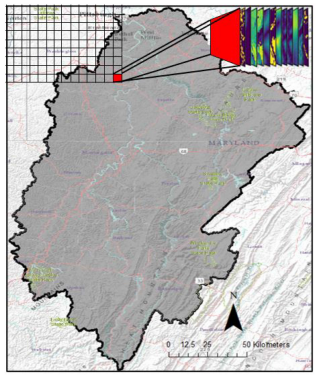


Fig. 2. Study area (showing the region in the Appalachian Mountain range in the U.S.) and data representation using the spatiotemporal data cube (shown in the cut-out).

We used distance to water, road, recreational sites, crop land, oil and gas wells, mines, development, financial regions along with, multiple density variables, geopolitics information, terrain-related variables, and socioeconomic variables in this study. Major demographic, socioeconomic, and land-cover-related features were encompassed in the raster data layers;

[1]After changing the categorical variables to binary dummy variables, we obtain a 71-D feature space.

moreover, we used a digital elevation model as a graphical representation of terrain. We used the data of 2001 and 2011 to be able to align the temporal data with other census-based socioeconomic datasets in the study. We did not use any spectral feature in this study, because many of the applied feature classes were initially extracted from spectral images [15] and the applied data represented enough input for this model [18].

In total, there were 21 785 patches located within the study area. We partitioned them as 66.6% for training and the rest as test and validation data. The raster and rasterized datasets that are used in this work are obtained from public national data repositories, including the United States Geological Survey [15] and TIGER shapefiles for U.S. Census [34]. The model was trained on binary data of developed versus nondeveloped regions using the NLCD of 2001 and on a set of variables; the test data are the NLCD of 2011. The class of developed land in this study is defined based on low-, medium-, and high-intensity classes of Anderson classes [35], which are used in classifications of the NLCD. These classes of land have more than 20% of imperviousness.

The Monongahela watershed is located in the north central part of West Virginia (WV) and has experienced the most population growth from 2010 to 2020 based on the latest census of the counties in which the watershed intersects [34]. This region of WV characterizes the Appalachian region's composition of many small rural towns (70 towns with less than 3000 people) with a clustered corridor of metropolitan areas (three cities with a population of 15 000 or more). In terms of land cover and use, it contains a distributed mix of built-up developed areas, residential areas, and large expanses of unaltered natural composition of forests, fields, and agriculture [15]. The area is also mostly private land with few restrictions and lacks zoning except in the downtown areas of the three largest cities. No one industry or land cover or use dominates the area and, therefore, provides the opportunity to study and model the potential change as a good precursor for the larger Appalachian region. Moreover, the variety of land development types in this region (from very rural to urbanized developments) increases the complexity of the land change prediction process. In such situations, the major driving factors of change act locally. Thus, the models that consider conditions in neighboring cells' would have a better capability in dealing with the varying local conditions. Besides the urban–rural interactions and configurations in the study area, dominance of nondeveloped regions (95.9% of region comprises nondeveloped lands in 2001, the base year of the study) creates highly imbalanced data. In assessment of the model performance on class imbalanced data, it is crucial to use metrics that can factor in such data imbalance into the evaluation. Considering this, we used different metrics to measure the performance, so the measures are less sensitive to the skewed and imbalanced sample classes.

B. Model Structure

In this work, we use the spatiospectral data cubes and also investigate the significance of 3-D convolutional networks in capturing interchannel interactions and in evidence fusion using features and outputs from different models. Trainable image filters, local neighborhood operations using convolutional filters, and subsampling operations are applied in an alternating manner on the original raw input data, resulting in an increasingly complex hierarchy of feature maps. We use patches of size 32×32 because the model best responds to this patch size. The values at a cell (pixel location) in the output image denote the probability value for the developed land at the given cell. The probability of 0.50 or higher represents a class value of 1 (developed land), and lower values denote a class value of 0 (nondeveloped land).

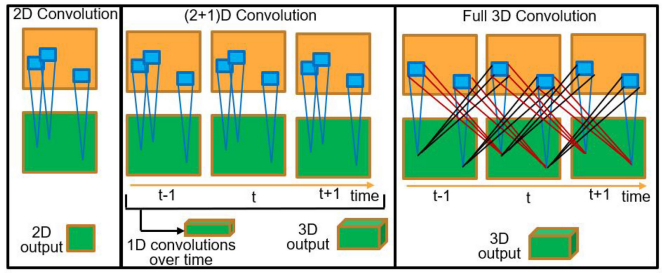


Fig. 3. 2-D versus 3-D convolutions. Here, 3-D convolution assumes a spectral dimension of 3 for the kernel. The method of (2+1)D first applies 2-D convolutions on each band; then, 1-D convolution is applied separately, along the spectral dimension in the spatiospectral data cube. Lines with different colors show the contribution of different spatiospectral bands in the 3-D convolution. Our figure is motivated by the approach used in [20].

1) Convolutional Neural Network: In 2-D convolutions, a small rectangular or square window with weights is moved around the image. The window (also called convolution kernel) is centered around each point in the image, and the weights are applied to the corresponding values in the original image. The weighted values from each position within the window are now added to obtain the final value for the center position. Thus, 2-D convolutions only account for spatial information in the image. For 3-D convolutions, the window is now extended to three dimensions (typically, using time for the third dimension). This small 3-D window is now moved around positions in the images, and the weights are applied as before, with final results as the sum of the weighted values within the 3-D window. The 3-D convolution captures the temporal dynamics. Fig. 3 shows the significant differences between 2-D and 3-D convolutions. In either case, different results can be obtained by simply changing the weights, window size, window shape, or how much overlap is allowed when the window is moved.

Tran *et al.* [33] used an intermediate approach between 2-D and full 3-D convolutions. They first performed 2-D convolutions on each frame. Then, for each position in the resulting 2-D image, they formed a temporal sequence and then applied 1-D convolution on this temporal sequence. They showed that this approach (called (2+1)-D) resulted in a better performance than using full 3-D convolutions. In this work, we will replace the temporal dimension used in video analysis with the spectral dimension in our data cube. Thus, the single "spectral" bands in our data cube will be analogous to individual frames in a video sequence. The 3-D convolutions will, thus, capture the spectral changes between bands in our spatiospectral data cube.

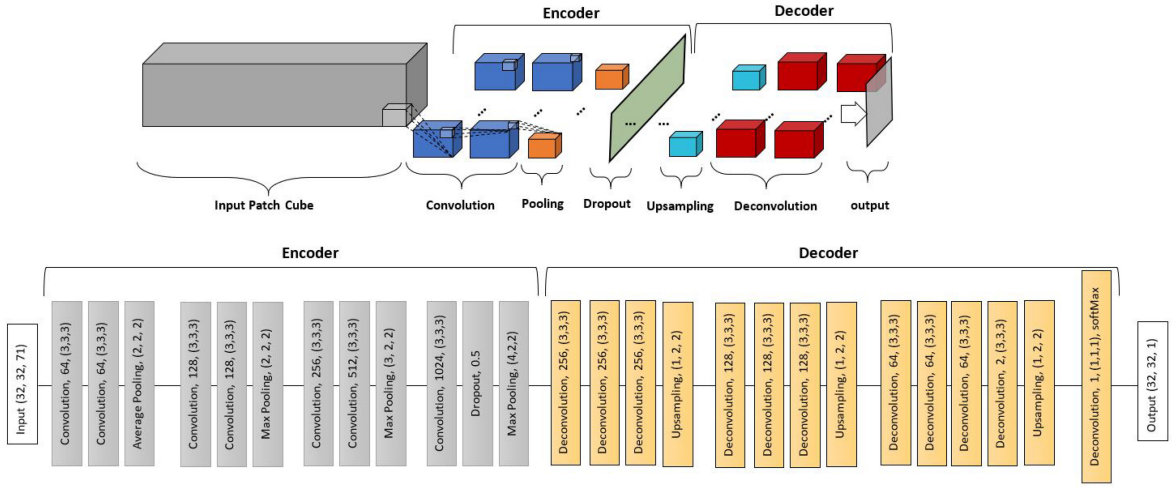


Fig. 4. Proposed model structure and deep learning architecture for predicting impervious land expansion. Top illustration is the 3-D model. Illustration at the bottom is the 2-D model with its layers as introduced in [11]. The same layers are used in the 3-D model.

We expand the dimension of the convolutional layers and poolings along the third dimension. By applying cubic convolutional filters, we aim to capture cross-variable relationships. Fig. 4 shows the network architecture for our proposed deep learning models using 3-D convolutions. The order of the layers for the 3-D CNN was similar to that of deepLandU; however, instead of 2-D convolutions and poolings, we used 3-D ones. The kernel size in the full 3-D model was $3 \times 3 \times 3$ with stride of one. Except for the first convolutional layer that had $ReLU$ activation, in other convolutional layers, we used $LeakyReLU$ activation, and the activation function of last layer was $softmax$. After encoding the input cubes, we transposed and upsampled the feature map back to 32×32. The kernel size of the convolutional layer in this model convolves across three of the channels. A variation of 3-D CNN kernels was also applied, where we broke the 3-D kernels into a (2+1)-D kernel. In the (2+1)-D CNN instead of applying full 3-D kernels with weights, we used a combination of 1-D and 2-D kernels. This breakdown considerably reduces the number of weights that need to be fine-tuned and, hence, the computations required.

Model Parameters: In the proposed 3-D models, just like 2-D models of deepLandU and deepLandS (see [11]), we use dropout at the final convolutional layer of the encoder, which accelerates the training and prevents overfitting. The decoder has more convolutional layers with more upsamplings. In all the layers except the first two layers, $LeakyReLU$, which is a version of rectified linear unit (ReLU) function, is used. The first two layers use ReLU as the activation function.

a) Activation functions: In the applied models, we use two activation functions of $LeakyReLU$, which is applied after each convolutional layer, and $softmax$ that is used at the last layer of each model for computing the class probability. $LeakyReLU$ minimizes the effect of the class imbalance data (see the following equation)

$$f(x) = \begin{cases} \text{x}, & \text{if } x \geq 0 \\ \alpha\text{x}, & \text{if } x < 0 \end{cases} \tag{1}$$

where x is the value of the input for the function from the convolutional layer and α is a coefficient for negative values of x (we use a value of 0.05 for α). $f(x)$ is the output of the activation function which is passed to the next layer. At the last layer, $softmax$ function [see (2)] is used as the activation function. Since our output in binary $softmax$ returns a probability value in the range of (0,1) per cell

$$\sigma(\vec{z_i}) = \frac{e^{z_i}}{\Sigma_{j=1}^{K} e^{z_j}} \tag{2}$$

where $\sigma(\vec{z_i})$ is the probability of the ith cell to be a hit, and cells with values higher than 0.5 are labeled as developed lands, $\vec{z_i}$ is the vector of model output from the last convolutional layer, e^{z_i} is the exponential function of this vector, and e^{z_j} is the exponential function of the output vector for K classes, where in our study we have only two classes, namely, developed land and nondeveloped land.

b) Optimizer: We use the adaptive moment estimation optimizer, which calculates the adaptive learning rates for each parameter [36].

c) Loss function: The loss function for this model was binary cross entropy, which computes the loss value as follows:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{3}$$

where $\hat{y}_i$ is the estimated value per example, $\hat{y}$ is the true value, and N is the number of samples.

C. Evidence Fusion

Training deep networks requires investment of time and resources, yet there is no guarantee that the generalized error is low for the test samples [37]. Ensemble models use the outcomes coming from different networks and select the best. Training multiple networks carries higher computational and time expenses; nevertheless, the combination of results adds to the bias, which balances the variance of only one trained

network [37]. Through combining the predictions from multiple models, ensembles reduce the variance and generalization error of predictions. In this study, we used evidence fusion methods at decision, feature, and score levels to avoid overfitting drawbacks of applying single networks and improve the model performance.

1) Decision-Level Fusion (DF): We utilized DF using majority vote. This method is applied to identify the label of cell c at patch p. The label of c_{ij} in row i and column j is the most frequent label from $l = \{l_{ij}^1, l_{ij}^2, l_{ij}^3, \ldots, l_{ij}^m\}$, where l_{ij}^t is the label from the tth model in row i and column j, and $l_{ij}^t \in \{0, 1\}$.

2) Score-Level Fusion (SF): We used two methods of averaging and classification for conducting SF.

a) Averaging the scores: For cell x at patch k, we implemented the following equation to compute the score of x:

$$s_{ij} = \frac{\sum_{t=1}^{m} s_{ij}^t}{m} \tag{4}$$

where s_{ij} is the score of cell c at row i and column j of patch k, s_{ij}^t is the output score of the tth model at row i and column j, and m is the total number of models, whose results are considered in this method.

b) Learning-based SF: We used a random forest classifier to classify the output scores of m models. Random forest is capable of dealing with class imbalanced data, which is the case for our dataset. Random forest deploys a random subset of features and looks for the most discriminative threshold, where these thresholds are randomly drawn for each feature and the splitting rule sets the best of these randomly generated thresholds. This causes a variance reduction for the model at the expense of higher bias. We used extremely randomized trees, which goes one step further than the regular random forest method and allows a high level of randomness in splitting the trees. This method requires adjustments in two parameters of *the number of trees (estimators)* in the forest and *maximum depth of the trees*. Large number of trees will be computationally expensive and at some threshold changes in the number of trees does not improve the results significantly. The number of trees represents the size of the random subsets for the features that are considered when splitting a node. The smaller this value, the greater the variance reduction and bias increase. To find the best values for the number of estimators and trees, we cross validated the results. Such k-fold cross- validation is performed on sets of $s = \{s^1, s^2, s^3, \ldots, s^m\}$, where s^i is a set the score results acquired from the ith model. And for set s in $\{s^1, s^2, s^3, \ldots, s^m\}$, the value of $|s|$ is equal.

3) Feature-Level Fusion (FF): At the feature-level data fusion, a pairwise merge of feature maps is conducted before applying the $softmax$ function at each epoch. A linear combination of the features from the last convolution layer is passed to the activation function at the last layer. We used a sum operation of weights, such that for models $m_1, m_2, m_3, \ldots, m_n$, the stacked weight is computed as

$$W = \sum_{i=1}^{n} w_i \tag{5}$$

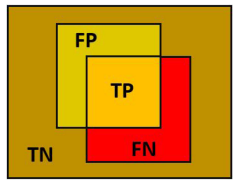


Fig. 5. Sensitivity and specificity diagram.

where w_i is the output weights of model i, n is the number of featurewise fused models, and W is the total weight feature that is passed to the $softmax$ function. In this analysis, the value of n is set to two. This procedure helps the models to essentially get the weights fine-tuned over a linear combination of weights.

D. Performance Measurement

To evaluate the model performance, we used the methods that avoid the problems of class imbalanced data. These methods moderate or remove the impact of true negative (TN in Fig. 5) in the assessment. Moreover, the time complexity of CNN model run encouraged us to use time efficiency as a performance measurement metric.

1) Area Under Receiver Operating Curve (AUROC): The AUROC shows the tradeoff between true positive rate (TPR or recall) and false positive rate (FPR) across different decision thresholds (see the following equations):

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{7}$$

AUROC is the probability that a classifier will rank a random positive instance higher than a negative one [38]. This metric is used for assessing the performance of binary classification models. We used AUROC in [11] to measure the model performance; however, we noticed that the AUROC value can be too optimistic about the performance of our models. The reason is the highly imbalanced datasets used in this study, in which an excessive improvement in the number of false positives changes the FPR negligibly. This happens because small TN value will produce very high FPR, which will reduce the impact of FP on the FPR [38]. Still we are interested in considering AUROC to compare models; we also used precision and recall (TPR) metrics in model evaluation [see (6) and (8)]

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{8}$$

2) Dice Coefficient: In semantic segmentation, the Dice similarity coefficient approximates the similarity of positive labeled cells to the positive ground truth [see (9)]. This similarity metric has been widely used in evaluation of image segmentation results. Since TN regions do not impact the magnitude of the Dice coefficient, it is robust to class imbalance (see Fig. 5)

$$\text{Dice} = \frac{2 \times \text{TP}}{(\text{TP} + \text{FN}) + (\text{TP} + \text{FP})}. \tag{9}$$

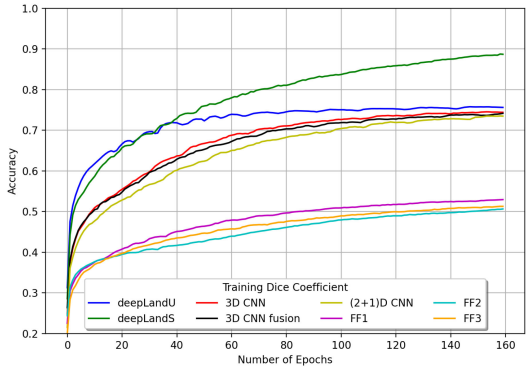


Fig. 6. Training Dice coefficient.

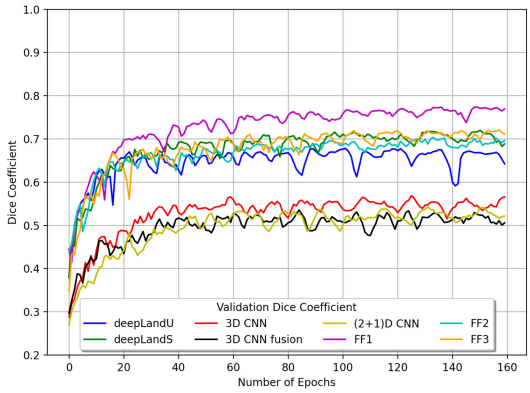


Fig. 7. Validation Dice coefficient.

3) Time Complexity: A major drawback in application of deep learning is the time complexity for training the features. Runtime of these models depends on the number of trainable parameters. In this research, we investigated the running time of the models as a performance measurement. To verify that the reported times are standardized, we used NVIDIA Tesla K80 GPU nodes on Pittsburgh Supercomputing Center Bridges for all the reported times. Access to the GPUs was provided through the Extreme Science and Engineering Discovery Environment virtual system [39]. However, due to limitations in the access time to Bridges GPUs, we used EC2 g3.16xlarge instances of Amazon Web Services to run the models with high total time.

IV. Results

The results of this study are based on 160 epochs per model. We used the AUROC and the Dice coefficient in performance measurement of training. Figs. 6 and 7 illustrate the Dice coefficient value of training and validation. Fig. 6 shows that the training process of 2-D fusion models have similar learning trend, and the training Dice coefficient value of these models is lower than that of other models. Figs. 6 and 7 also show similarity in the Dice values in 3-D models. We tested the results of this similarity in learning and fine-tuning trends by creating a Dice similarity confusion matrix of the model outputs.

The pairwise Dice coefficient of model outputs (excluding the score and feature fusion results) (see Fig. 8) shows that

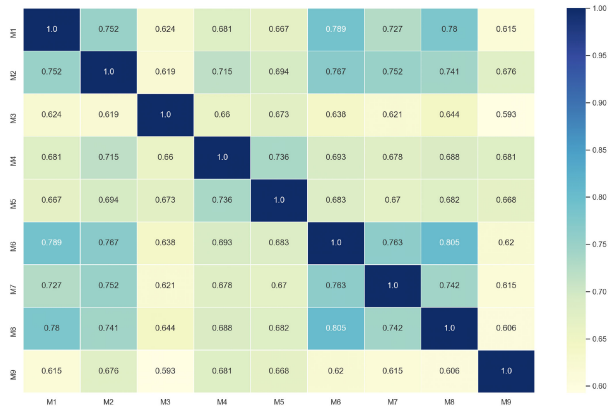


Fig. 8. Pairwise Dice coefficient of model predictions (M1): deepLandS, (M2): deepLandU, (M3): U-Net, (M4): 3-D deepLand, (M5): (2+1)D deepLand, (M6): 2-D FF(1&2), (M7): 2-D FF(2&3), (M8): 2-D FF(1&3), and (M9): 3-D Fusion.

the results of deepLand-U, deepLand-S, and 2-D FF models confirm each other. On the other hand, the results of 3-D models, U-net, and 3-D FF models have a low level of agreement with the other models. This confusion matrix brings a legitimate description on the bias–variance balance of the fusion models. It shows that since 2-D models of deepLand-U and deepLand-S generate similar results, fusing these models at the feature level would still lead to similar results. On the other hand, U-Net and deepLand-U have lower similarity, and the fusion of these two models represents much higher accuracy. But, this is not the case for 3-D and (2+1)-D models, which represent 73.59% Dice similarity. Feature fusion of these two models in 160 epochs represents a lower Dice coefficient value compared to independent implementation of these models. Table I indicates that the precision value has increased in 3-D FF results, but recall has decreased, which means that the false negative labels are higher in the fusion of 3-D and (2+1)-D models.

The results of SF indicate the highest Dice similarity and precision values among all the results. After conducting fivefold cross validation for the random forest models, the RF classifier has been trained with 1000 estimators and 41 trees. This model outperformed all other models in the study. The model results of SF-RF show that this model is capable of predicting both positive and negative classes at high accuracy, and it has the lowest difference between precision and recall values. The data used in this work are class imbalanced, and as discussed in Section III-D, we can also observe that all the models exhibit higher AUROC in both training and validation compared to the Dice values (see Table I). This indicates that the nature of the data, with majority of 0 values, has a significant impact on the optimistic AUROC values, where the Dice coefficient disregards the high TN values and demonstrates lower accuracy measures.

Running time of the models shows that the time expenses correlate with the number of trainable parameters (see Table II). Also, adding a dimension to the convolutional kernel window and poolings has considerable impacts on the required time for training the models. As mentioned, variation of 3-D network requires less time for training; still this runtime is much higher than time expenses of 2-D CNN model training.

TABLE I
COMPARISON OF THE MODEL ACCURACY RESULTS

ID	Model	Training		Validation			
		AUROC	Dice	AUROC	Precision	Recall	Dice
1	deepLandS	0.9884	0.9098	0.9884	0.7785	0.7110	0.7175
2	deepLandU	0.9866	0.7565	0.9866	**0.8350**	0.6293	0.6617
3	U-Net	0.9811	**0.9264**	0.9820	0.5807	0.6321	0.5392
4	3D deepLand	0.9846	0.7528	0.9846	0.6886	0.6222	0.5508
5	(2+1)D deepLand	0.9785	0.7438	0.9785	0.6554	0.6380	0.5313
6	2D FF(1&2)	0.9877	0.5342	0.9877	0.8104	0.7970	0.7729
7	2D FF(2&3)	0.9758	0.5461	0.9758	0.7407	0.7767	0.7123
8	2D FF(1&3)	0.9741	0.5383	0.9742	0.7557	0.7013	0.6762
9	3D FF	0.9849	0.7449	0.9848	0.7506	0.4740	0.5134
10	SF-averaging	0.9979	0.8822	0.9940	0.8283	0.7293	0.7755
11	SF-RF	0.9853	0.8083	0.9927	0.8011	**0.8130**	**0.8070**
12	LF-voting	**0.9986**	0.8470	**0.9941**	0.8244	0.6963	0.7548

The highest scores in each column are highlighted.*Notations:* FF(1&2): Feature fusion of deepLandS and deepLandU; FF(2&3): Feature fusion of deepLandU and U-Net; and FF(1&3): Feature fusion of deepLandS and U-Net.

TABLE II
COMPARISON OF THE TIME REQUIRED FOR THE MODEL RUNS

Model	Time Per Step	Time Per Epoch	Number of Trainable Parameters
deepLandS	6ms	86s	17,630,913
deepLandU	4ms	64s	10,772,997
U-Net	5ms	65s	30,764,805
3D deepLand	42ms	606s	32,190,981
(2+1)D deepLand	28ms	398s	12,208,927
2D FF(1&2)	9ms	123s	28,403,910
2D FF(2&3)	11ms	164s	48,395,718
2D FF(1&3)	12ms	171s	41,537,802
3D Fusion	69ms	988s	44,399,908

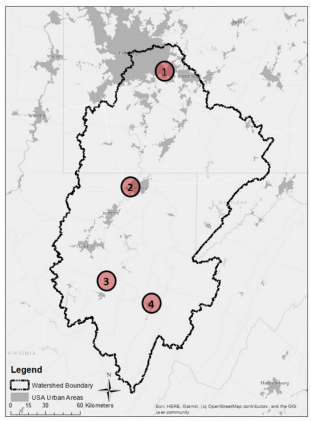


Fig. 9. Location of the patches.

An illustration of the results in four different patches (see Fig. 9) is represented in Fig. 10. These four patches show model prediction on the test dataset in four areas. We selected these patches such that they represent model prediction in geographic regions of different developed land density and urbanization level. The first patch is located in Pittsburgh, PA metropolitan urban area, which according to US census data in 2010 has a population of more than 2 300 000. Patch 2 is in Morgantown, WV metropolitan area, a college town with a population of 137 251. Two other patches are selected from rural regions.Fig. 10 shows a visualization of model performance in predicting land development in both rural and urban areas. Fig. 10 shows that the performance of models for different forms of development in rural and urban areas varies. Visualized results of first five models show that these models predict denser land development area than the ground truth, and they were not capable of teasing out the state of adjacent cells. So, their prediction, mostly in urban areas, creates aggregated new developments. Visualization of 2-D FF (1&2) shows that in patch 1 (P1)—more urbanized regions—the results are akin to the ground truth. However, in patches 2–4, the results of this model are not as promising as SF-RF model. SF-RF demonstrates promising results at different levels of development density. Visualization of SF-RF in all patches exhibits similar results to the ground truth. Since the whole study area is a combination of different levels of land development with dominance of rural areas, the generalization of the model results in different types of regions becomes a critical consideration in overall model performance.

V. CONCLUSION

The results of this study imply that evidence fusion at decision and score levels improves the model performance. With FF, the model output shows some improvement in the results after stacking the weights in some of the models. 3-D FF results did not show much of improvements in the output; this is, in fact, the result of the number of parameters that need to be fine-tuned at this level of fusion. We suggest more consideration on the learning rate parameter and the number of epochs, which could improve the output of these models. Due to time constraints for access to computational facilities, we did not explore an extensive range of parameter values. This could be done as part of future work in this area. Cross-channel relationships impact the output of the 3-D CNN models. Hence, to some

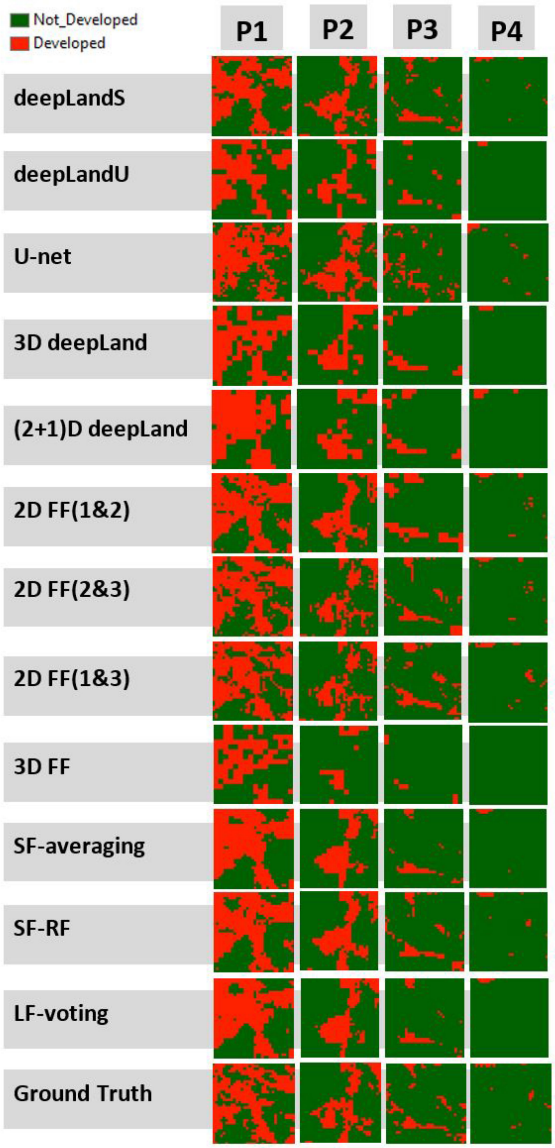


Fig. 10. Model prediction in four patches; each patch column includes the labeled results. See Fig. 9 for patch locations.

extent, the order of the channels could impact the outcome of the 3-D CNN models. This question could be further investigated, for instance, by considering the different permutations of the feature space and the accompanying data management issues this will raise. Thus, additional research is needed to better understand how the interaction between the spectral bands can affect the results. On the other hand, FF of 2-D CNNs indicates higher performance measurement on validation data compared to training. The results of this research also indicate that the SF-RF model, which is an ensemble model with ransom forest classifier applied to the score outputs, has promising performance in different geographic regions with various forms of land development. This idea could be further elaborated by applying similar methods to other geographic regions and different classes of land cover.

TABLE III
FUSION RESULTS WITH AND WITHOUT 3-D MODELS

	Model	Dice
With 3D Models	Averaging	0.7755
	SF-RF	0.8070
	Voting	0.7548
Without 3D Models	Averaging	0.7601
	SF-RF	0.8013
	Voting	0.7234

Although 3-D CNN models did not show very high accuracy on their own (see Table I), still the results of these models can be investigated in terms of their role in casting good votes and balancing the scores using decision fusion or score fusion. We used the output of 2-D models to conduct DF and SF (see Table III). The results in Table III imply that fusion with 3-D models leads to a higher accuracy when compared to fusion results without the 3-D models. This suggests that we can improve the overall results by incorporating the results from 3-D models, even when using only 3-D models independently may not always lead to the highest accuracy.

For future work, we suggest the application of spectral remote sensing images along with other land characteristics. We anticipate that including spectral images directly as part of the input data would allow the model to more effectively capture cross-spectral interactions in modeling the land transformation process.

ACKNOWLEDGMENT

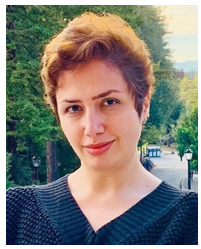
The authors would like to thank Dr. Jack Smith, a Cyberinfrastructure Coordinator of the National Science Foundation's Experimental Program to Stimulate Competitive Research in West Virginia, for providing access to computational resources at bridge nodes.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

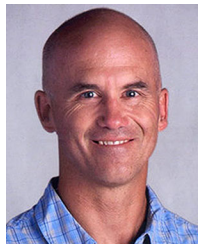
[1] G. C. Lin, "Understanding land development problems in globalizing China," *Eurasian Geogr. Econ.*, vol. 51, no. 1, pp. 80–103, 2010.
[2] B. Wilson and A. Chakraborty, "The environmental impacts of sprawl: Emergent themes from the past decade of planning research," *Sustainability*, vol. 5, no. 8, pp. 3302–3327, 2013.
[3] A. A. Ntelekos, M. Oppenheimer, J. A. Smith, and A. J. Miller, "Urbanization, climate change and flood policy in the United States," *Climatic Change*, vol. 103, no. 3-4, pp. 597–616, 2010.
[4] B. Pijanowski, A. Tayyebi, M. Delavar, and M. J. Yazdanpanah, "Urban expansion simulation using geospatial information system and artificial neural networks," *Int. J. Environ. Res.*, vol. 3, no. 4, pp. 493–502, 2009.
[5] S. Jin, L. Yang, P. Danielson, C. Homer, J. Fry, and G. Xian, "A comprehensive change detection method for updating the national land cover database to circa2011," *Remote Sens. Environ.*, vol. 132, pp. 159–175, 2013.
[6] C. Kamusoko and J. Gamba, "Simulating urban growth using a random forest-cellular automata (RF-CA) model," *ISPRS Int. J. Geo-Inf.*, vol. 4, no. 2, pp. 447–470, 2015.

[7] H. M. Rizeei, B. Pradhan, and M. A. Saharkhiz, "Surface runoff prediction regarding LULC and climate dynamics using coupled LTM, optimized ARIMA, and GIS-based SCS-CN models in tropical region," *Arabian J. Geosci.*, vol. 11, no. 3, 2018, Art. no. 53.
[8] B. C. Pijanowski, D. Hyndman, and B. A. Shellito, "The application of the land transformation, groundwater flow and solute transport models for Michigan's grand traverse bay watershed," in *Proc. Nat. Amer. Plann. Assoc. Meeting*, New Orleans, LA, USA, 2001, pp. 1–12.
[9] B. C. Pijanowski, D. G. Brown, B. A. Shellito, and G. A. Manik, "Using neural networks and GIS to forecast land use changes: A land transformation model," *Comput., Environ. Urban Syst.*, vol. 26, no. 6, pp. 553–575, 2002.
[10] B. C. Pijanowski, A. Tayyebi, J. Doucette, B. K. Pekin, D. Braun, and J. Plourde, "A big data urban growth simulation at a national scale: Configuring the GIS and neural network based land transformation model to run in a high performance computing (HPC) environment," *Environ. Model. Softw.*, vol. 51, pp. 250–268, 2014.
[11] P. Pourmohammadi, D. Adjeroh, and M. P. Strager, "Predicting impervious land expansion using deep deconvolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 9855–9858.
[12] A. E. Maxwell, M. P. Strager, T. A. Warner, C. A. Ramezan, A. N. Morgan, and C. E. Pauley, "Large-area, high spatial resolution land cover mapping using random forests, GEOBIA, and NAIP orthophotography: Findings and recommendations," *Remote Sens.*, vol. 11, no. 12, 2019, Art no. 1409.
[13] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, 2015, Art. no. 258619.
[14] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," 2015, *arXiv:1508.00092*.
[15] *United States Geological Survey (USGS)*, 2020. [Online]. Available: URL https://www.usgs.gov/core-science-systems/national-geospatial-program/national-map
[16] D. Wu, J. Liu, G. Zhang, W. Ding, W. Wang, and R. Wang, "Incorporating spatial autocorrelation into cellular automata model: An application to the dynamics of Chinese Tamarisk (Tamarix Chinensis Lour)," *Ecol. Model.*, vol. 220, no. 24, pp. 3490–3498, 2009.
[17] P. H. Verburg, T. C. de Nijs, J. R. van Eck, H. Visser, and K. de Jong, "A method to analyse neighbourhood characteristics of land use patterns," *Comput., Environ. Urban Syst.*, vol. 28, no. 6, pp. 667–690, 2004.
[18] P. Pourmohammadi, D. Adjeroh, M. Strager, and Y. Farid, "Predicting developed land expansion using deep convolutional neural networks," *Environ. Model. Softw.*, vol. 134, 2020, Art. no. 104751.
[19] D. He, Q. Shi, X. Liu, Y. Zhong, and X. Zhang, "Deep subpixel mapping based on semantic information modulated network for urban land use mapping," *IEEE Trans. Geosci. Remote Sens.*, 2021, doi: 10.1109/TGRS.2021.3050824.
[20] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.
[23] S. Ashiqur Rahman, P. Giacobbi, L. Pyles, C. Mullett, G. Doretto, and D. A. Adjeroh, "Deep learning for biological age estimation," *Brief. Bioinf.*, vol. 22, no. 2, pp. 1767–1781, 2021.
[24] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[25] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.
[26] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial-spectral fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2485–2501, 2020.
[27] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, "A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3085870.
[28] D. He, Y. Zhong, X. Wang, and L. Zhang, "Deep convolutional neural network framework for subpixel mapping," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3032475.
[29] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 1–9.
[30] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," 2016, *arXiv:1606.02585*.
[31] K. Zhou, D. Ming, X. Lv, J. Fang, and M. Wang, "CNN-based land cover classification combining stratified segmentation and fusion of point cloud and very high-spatial resolution remote sensing image data," *Remote Sens.*, vol. 11, no. 17, pp. 1–28, 2019.
[32] B. Huang, L. M. Collins, K. Bradbury, and J. M. Malof, "Deep convolutional segmentation of remote sensing imagery: A simple and efficient alternative to stitching output labels," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2018, pp. 6899–6902, doi: 10.1109/IGARSS.2018.8518701.
[33] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6450–6459.
[34] U.S. Census, 2019. [Online]. Available: URL https://www.census.gov/
[35] J. R. Anderson, "A land use and land cover classification system for use with remote sensor data," U.S. Geological Survey Professional Paper 964, United States Geological Survey, Washington, DC, USA, 1976.
[36] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[37] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[38] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[39] J. Towns *et al.*, "XSEDE: Accelerating scientific discovery," *Comput. Sci. Eng.*, vol. 16, no. 5, pp. 62–74, 2014.



Pariya Pourmohammadi (Member, IEEE) received the M.Sc. degree in computer science in the area of computational data science and the Ph.D. degree in human and community development in 2019, from the West Virginia University, Morgantown, WV, USA.

She is currently a Postdoctoral Scholar with the Sierra Nevada Research Institute, University of California, Merced, CA, USA. She has a multidisciplinary background in computer science and environmental planning. Her research interests fall into the intersection of spatially explicit models, computer science, and land change analysis.



Michael P. Strager received his Ph.D. in natural resource economics from West Virginia University in 2004, He is a Professor with the Division of Resource Economics and Management, West Virginia University, Morgantown, WV, USA. His research interests include applied spatial analysis and applications for natural resource management.

Donald A. Adjeroh (Member, IEEE) received the Ph.D. degree in computer science from the Chinese University of Hong Kong, Hong Kong, in 1997.

He is currently a Professor and Associate Department Chair with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA. His work has been supported by grants from various federal agencies in the U.S. His research interests include machine learning, data analytics, search data structures, bioinformatics, and digital health.

Dr. Adjeroh received the Department of Energy CAREER Award in 2002 and the WVU Statler Researcher of the Year Award in 2021.