Spatial Variability Aware Deep Neural Networks (SVANN): A General Approach

JAYANT GUPTA and CARL MOLNAR, University of Minnesota
YIQUN XIE, University of Maryland
JOE KNIGHT and SHASHI SHEKHAR, University of Minnesota

Spatial variability is a prominent feature of various geographic phenomena such as climatic zones, USDA plant hardiness zones, and terrestrial habitat types (e.g., forest, grasslands, wetlands, and deserts). However, current deep learning methods follow a spatial-one-size-fits-all (OSFA) approach to train single deep neural network models that do not account for spatial variability. Quantification of spatial variability can be challenging due to the influence of many geophysical factors. In preliminary work, we proposed a spatial variability aware neural network (SVANN-I, formerly called *SVANN*) approach where weights are a function of location but the neural network architecture is location independent. In this work, we explore a more flexible SVANN-E approach where neural network architecture varies across geographic locations. In addition, we provide a taxonomy of SVANN types and a physics inspired interpretation model. Experiments with aerial imagery based wetland mapping show that SVANN-I outperforms OSFA and SVANN-E performs the best of all.

CCS Concepts: • **Information systems** → *Data mining*; • **Computing methodologies** → **Neural networks;**

Additional Key Words and Phrases: Neural networks, spatial variability

ACM Reference format:
Jayant Gupta, Carl Molnar, Yiqun Xie, Joe Knight, and Shashi Shekhar. 2021. Spatial Variability Aware Deep Neural Networks (SVANN): A General Approach. *ACM Trans. Intell. Syst. Technol.* 12, 6, Article 76 (November 2021), 21 pages.
https://doi.org/10.1145/3466688

1 INTRODUCTION

Deep learning techniques have resulted in significant accuracy improvements in image-based object recognition [18, 36] and semantic segmentation [16] tasks. The use of multiple layers in these techniques allows approximate modeling of all continuous functions [4]. Unlike traditional machine learning, which requires manual feature engineering, deep learning models interpret the data and automatically generate features [6]. The current deep learning literature [10, 24, 42] follows a spatial **one-size-fits-all (OSFA)** approach in which deep neural networks are trained with

This material is based on work supported by the National Science Foundation under Grant No. 1737633.
Authors' addresses: J. Gupta, C. Molnar, and S. Shekhar, Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 4-192 Keller Hall, 200 Union Street SE, Minneapolis, MN 55455; emails: {gupta423, molna018, shekhar}@umn.edu; Y. Xie, Department of Geographic Sciences, Center for Geospatial Information Science, University of Maryland, College Park, 1124 Lefrak Hall, 7251 Preinkert Dr., College Park, MD 20742; email: xie@umd.edu; J. Knight, Department of Forest Resources, University of Minnesota, Twin Cities, 1530 Cleveland Ave. N, St. Paul, MN 55108, USA; email: jknight@umn.edu.

2157-6904/2021/11-ART76 $15.00
https://doi.org/10.1145/3466688

(a) Igloo in Liyama City, Japan (b) Apartments in Shanghai, China (c) Houses in a U.S. suburban neighborhood (d) Ancient cave houses in Petra, Jordan

Fig. 1. Spatial variability in houses and their surroundings.

(a) Snow cover in polar region (b) Salt flats in Rann of Kutch, India

Fig. 2. Location knowledge provides context to distinguish between otherwise visually similar features.

no consideration of spatial variability. However, geographic properties differ across different areas giving rise to varied geophysical and cultural phenomena. Failure to account for this variability results in inconsistent models (e.g., object detection, segmentation) across geographic areas.

Knowledge of spatial variability is necessary to understand the spatial patterns of events and objects over an area [38]. Machine learning models can be affected by two types of spatial variability: variability in the objects of interest themselves, which may differ in shape, size, or both, and variability in the surroundings of an object of interest. For example, a computational model that is trained to find residential housing in the United States may have difficulty with the task of finding houses in other countries where housing construction is adapted to different local climates or other conditions (cave houses in Petra, igloos in polar regions, etc.), and where the neighboring surroundings may differ as well. Figure 1 shows examples of spatial variability in houses and their surroundings across the globe.

Location knowledge provides relevant context that is necessary to distinguish between visually similar imagery features. For example, the scenic views in Figure 2 appear almost identical. Once the geographic location of each image is known, however, they are easily differentiated. Similarly, knowledge of spatial neighborhood such as spatial co-location of a dense tree canopy with a river can help highlight mangrove forests.

Spatial variability is a prominent feature of many geographic phenomena, including climate zones, USDA plant hardiness zones [26], and terrestrial habitat types (e.g., forest, grasslands, wetlands, and deserts). Differences in climate zones affect the plant and animal life of a region. Similarly, knowledge of plant hardiness zones helps gardeners and growers assess appropriate plants for a region. Furthermore, laws, policies, and culture differ across countries and even states within some countries. Spatial variability is considered as the second law of geography [22] and has been adopted in regression models (e.g., **geographically weighted regression (GWR)** [8]) to quantify

Table 1. Application Domain and Use Case of Spatial Variability

Application Domain	Example Use Cases
Wetland mapping	Wetlands in Florida (e.g., mangrove forest) are different from those in Minnesota (e.g., marsh), which affects their water filtration, habitat suitability, and carbon sequestration.
Cancer cell identification	Cancer cells in pathology tissue samples are known to be spatially heterogeneous [12].
Vehicle detection	Vehicle types differ across India (e.g., auto-rickshaw) and the United States (SUVs).
Residence detection	House types and design (e.g., igloos, huts, flat-roof) differ across geographic areas.
Urban agriculture	Detection of urban gardens: Urban gardens designs may vary across rural (large ones), suburban (small backyard gardens), and urban (e.g., container gardens, community gardens) areas due to differing space availability and risks (e.g., deer, rabbit).
Precision agriculture	Soil properties may vary from one location to another due to various factors.

the relationships among variables across a study area. In this work, we assess the effect of spatial variability on pixel-level image segmentation built using deep learning techniques. Spatial variability is challenging to quantify due to many geophysical factors that influence it. For example, soil scientists are often interested in understanding soil characteristics (e.g., carbon content) at a location to determine yield. They have observed that soil samples (which are) collected in an area of 100 m^2 can vary significantly depending on factors such as tillage, soil composition, vegetation, land management practices, and topography [2].

This article examines the effect of spatial variability on different neural network architectures and their weights. Previously, we proposed a spatial variability aware neural network (SVANN-I, formerly called *SVANN*) approach where weights are a function of location, but the neural network architecture is location independent [11]. We also described two types of training and prediction methods and evaluated them using the aerial imagery from two geographic areas for mapping urban gardens. In this work, we explore SVANN-E where the neural network architecture varies across geographic locations. We evaluate SVANN-E with aerial imagery-based wetland mapping. Broadly, the idea of SVANN-I is similar to interpolation and the idea of SVANN-E is similar to extrapolation.

We also provide a taxonomy of SVANN types and an analogous interpretation using physics-inspired models. For example, we can find an analogy between partial differential equations and neural network models, where the set of (partial) derivatives is analogous to neural network architecture and the coefficients are analogous to the weights in the neural network. This analogy can be extended to the training of models using spatial data from two locations and modeling the different phenomena using differential equations. An appropriate modeling of the variability across the two locations (or two phenomena) will help improve the efficiency of models to understand, interpret, and make predictions.

Application domains and example use cases where spatial variability is relevant and needs to be considered include mapping human residences from satellite imagery (Figure 1), wetland mapping, cancer diagnosis, and many others. Additional examples are listed in Table 1.

Table 2. Table of Notations

Symbol	Description
f	Network architecture
K	Number of layers in the network
loc_M	Model location
$w^i_{loc_M}$	Weight of the i^{th} layer at loc_M
loc_S	Sampling location
$x(loc_S)$	Sample located at loc_S
$y(loc_M)$	Output of model located at loc_M
d	Distance threshold
η	Learning rate
$d(loc_M, loc_S)$	Euclidean distance between loc_M and loc_S
$\Delta y^i(loc_S)$	Back-propagation error at layer i for sample at loc_S
f_{loc_M}	Location-dependent network architecture

Contributions:

(1) We propose a taxonomy of **spatial variability aware deep neural networks (SVANNs)** that classifies networks by the level of spatial variability in their architectures and weights.
(2) We propose a SVANN-E approach where neural network architecture varies across geographic locations.
(3) We evaluate the SVANN-E approach using a new two-step extrapolation method, where step 1 is architecture selection and step 2 is weight calibration.

Scope: This article focuses on geographic and other low-dimensional space. Generalization of the proposed approaches to model variability in high-dimensional spaces is outside the scope of this work. We use **convolutional neural networks (CNNs)** for the experimental evaluation and case studies. The evaluation dataset in this work is limited to high spatial resolution RGB imagery. We do not evaluate SVANN against other types of neural networks. Detailed discussion of the subtypes of SVANN-I and SVANN-E is also beyond the scope of this work.

Organization: The article is organized as follows. Section 2 reviews the details of SVANN-I along with different training and prediction procedures and previous results. Section 3 describes the SVANN-E approach and provides a formal result. Section 4 describes the evaluation framework, giving details on the experiment design, evaluation task, evaluation metric, architecture, and dataset. In Section 5, we present the results and a discussion of the effects of spatial variability. Section 6 provides a SVANN taxonomy, a physics-inspired interpretation of the taxonomy, other observations, and a brief review of the relationship of the ideas to the broader literature. Finally, Section 7 concludes the article with future directions.

2 SVANN-I

To keep this article self-contained (for our readers), we summarize previous results [11] on SVANN-I (formerly called *SVANN*) training and prediction procedures. We also provide a table of notations (Table 2) that lists relevant symbols in the order of their use.

Spatial OSFA. Figure 3 shows the OSFA approach using a CNN with three layers: a convolution layer, a spatial pooling layer, and a fully connected layer. The initial two layers perform feature engineering and selection, whereas the fully connected layer is responsible for output prediction. As can be seen, the approach does not account for the geographic location of training samples.

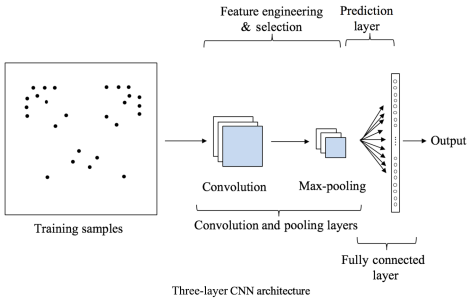


Fig. 3. Spatial OSFA approach using a CNN with three layers: convolution, spatial pooling, and a fully connected layer.

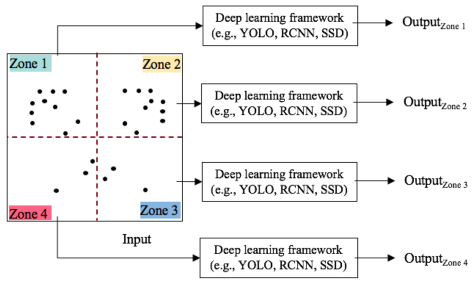


Fig. 4. SVANN-I using fixed-partition-based neighbors. Four distinct models are trained using training samples from each zone.

In previous work, we showed that an OSFA approach underperforms on an object detection task compared to SVANN-I.

SVANN-I. SVANN-I is a spatially explicit model where each neural network parameter (e.g., weight) is a function of the model's location loc_M. The architecture f is composed of a sequence of K weight functions or layers $(w^1(loc_M), \ldots, w^K(loc_M))$ that map a geographic location based training sample $x(loc_S)$ to a geographic location dependent output $y(loc_M)$ as follows:

$$y(loc_M) = f(x(loc_S); w^1(loc_M), \ldots, w^K(loc_M)), \tag{1}$$

where $w^i(loc_M)$ is the weight vector for the i^{th} layer. In this approach, we assume that the architecture is location invariant (i.e., K is constant for all the models). Figure 4 shows the SVANN-I approach where the geographic space has four zones and deep learning models are trained for each zone separately. For prediction, each zonal model predicts the test samples in its zone. We classified SVANN-I by the choice of training and prediction procedures. Here, we describe those procedures.

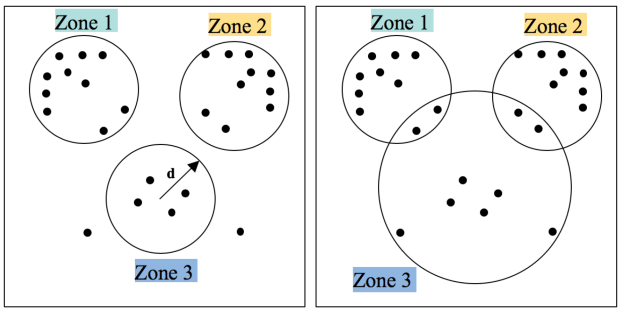
2.1 Training

There are at least two possible training procedures for SVANN-I, namely model-location-dependent sampling for learning and distance-weighted model-location-dependent sampling for learning.

2.1.1 Model-Location-Dependent Sampling for Learning. Model parameters for a location are derived by training the model using labeled samples from nearby locations. There are three types of nearest neighbor techniques that can be considered:

(a) *Fixed-partition-based neighbors*: Partitions (also known as zones) are used when policies and laws vary by jurisdiction, such as countries, U.S. states, counties, cities, and climatic zones. We use administrative, zonal partitions of geographic space to build individual models. This approach is simple but relatively rigid as partitions are usually disjoint and seldom change. Figure 4 illustrates SVANN-I model training using zone-based neighbors, where a sample from each zone is used to train a model for that particular zone. Partitioning the data based on zones can break up natural partitions (e.g., Zone 3 and Zone 4 in Figure 4.).

(b) *Distance-bound nearest neighbors*: In this training regime, a model at location (loc_M) is trained using nearby training samples within distance d. This model assumes that there are sufficient training samples in the vicinity of the model locations. This approach may be more flexible than a fixed-partition-based approach as the training samples can overlap across models and



(a) Distance-bound nearest neighbors (b) K-nearest neighbor

Fig. 5. Model-location-dependent sampling for learning.

the model locations can adapt to the spatial distribution (e.g., hotspots) of learning samples. Figure 5(a) shows training of different models using training samples within distance d.

(c) *K-nearest neighbors*: In this training regime, a model at location (loc_M) is trained using k-nearest training samples in the geographic space. This model does not assume that there are sufficient training samples in the geographic vicinity of model locations. Thus, this approach may be more flexible than distance-bound nearest neighbors. Figure 5(b) shows training of different models using k-nearest training samples.

All the samples selected for learning samples are treated equally in the training phase for model-location-dependent sampling for learning.

2.1.2 Distance-Weighted Model-Location-Dependent Sampling for Learning. In this approach, all training samples can be used to train models at different locations. To address spatial variability, nearby samples are considered more important than further away samples by adapting the learning rate. To update the neural network weights, the learning rate is multiplied by a back-propagation error and a function of the distance between the selected learning sample and the location of the model. This is equivalent to the learning rate being dependent on the distance between the labeled sample and the location for which the model is being trained. The distance function can be thought of as the inverse of the distance squared as follows:

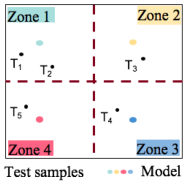
$$w^i(loc_M) = w^i(loc_M) + \frac{\eta}{d^2(loc_M, loc_S)} * x^i(loc_S) * \Delta y^i(loc_S), \tag{2}$$

where η is the learning rate, d is the distance between the location of learning sample (loc_S) and the location of model (loc_M), $x^i(loc_S)$ is the input to the i^{th} layer, and $\Delta y^i(loc_S)$ is the back-propagated error at layer i. This approach is similar to boosting techniques [9] where weak learners or hypotheses are assigned weights based on their accuracy. It is also similar to GWR [8] where regression coefficients and error are location dependent.

In the context of object detection or semantic segmentation in imagery via CNN, we note that CNN may favor nearby pixels over distant pixels (by using convolutional and pooling layers) within a single labeled sample (e.g., a 512 × 512 image), whereas the proposed method favors nearby labeled samples over a set of distant labeled samples.

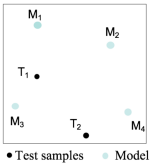
2.2 Prediction

Since multiple models are trained at different locations and a new sample may not be at those locations, we developed two prediction methods, zonal and distance weighted, to combine the predictions from multiple models for the new sample.



Partition	Sample	Prediction
Zone-1	T_1	1
	T_2	1
Zone-2	T_3	1
Zone-3	T_4	0
Zone-4	T_5	0

(a) Zonal prediction using 5 test samples and 4 partitions.



Model	Class T_1	$D(M_i, T_1)$	Class T_2	$D(M_i, T_2)$
M_1	1	2	1	5
M_2	0	3.5	1	4
M_3	1	1.5	0	2.5
M_4	1	5	0	1.5

Assigned Class T_1	Assigned Class T_2
1	0

(b) Distance weighted prediction using 2 test samples and 4 models. For illustration, approximate distances are used.

Fig. 6. Prediction methods in SVANN-I.

2.2.1 Zonal Prediction. Given a fixed partitioning of the geographic space (e.g., counties), prediction results from the model within the same partition will be used for prediction. If there are multiple models within a partition, voting (e.g., majority, mean) can be used for prediction. Here the votes from all models within the partition are treated equally. In addition, samples located at zone boundaries are disjoint and are assigned to a single zone. Zonal prediction is suitable for models trained on model-location-dependent learning samples. Figure 6(a) shows an example with five test samples (T_1 through T_5) and four partitions, where each model in a partition is a binary classifier representing classes as (0, 1). The Zone 1 model is used to make predictions for test samples T1 and T2. The Zone 2 model makes predictions for T3 and so on.

2.2.2 Distance-Weighted Prediction. Given a test sample and distances from all models, we weight the predictions from each model as an inverse function of the distance. The highest weighted prediction is assigned as the class of the test sample. Distance-weighted prediction is suitable for models trained using distance-weighted model-location-dependent learning samples. Figure 6(b) shows an example with two test samples and four models where each model predicts sample class (0 or 1). Assume that the adjacent (top right) table shows the predictions and distance ($D(M_i, T_i)$)) of each model from a set of test samples that are used to calculate class weights and assign class. All models are used to make a prediction for each test sample. For T_1, the nearest models (M_1, M_3) predict its class as 1, whereas for T_2, the nearest models (M_3, M_4) predict its class as 0. Therefore, the final assigned classes (shown in bottom right table) for the two test samples are 1 and 0, respectively.

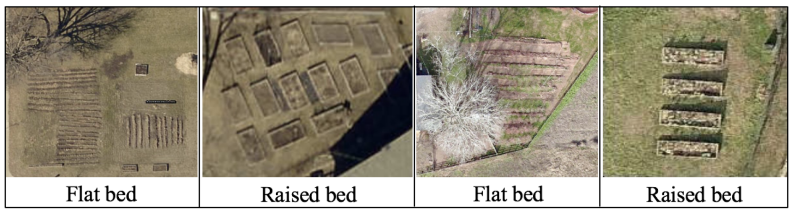
2.3 Validation Results

We evaluated SVANN-I on an urban garden detection task. Given aerial images from different places and an object definition (for urban garden), we built a computational model to detect the object having high precision and recall. The problem was challenging because of spatial variability, the large size of the geographic area (order of 1,000 km^2), and ambiguous annotations due to low distinction of the gardens from their background. We trained individual models for two disjoint and distant geographic regions (i.e., Hennepin County, Minnesota, and Fulton County, Georgia). Table 3 shows that SVANN-I performed better than OSFA on the task, achieving a 14.34% higher F1-score overall.

Characteristic-based interpretation showed that Fulton County had a significantly higher proportion of raised beds to flat beds compared to Hennepin County. This may suggest different gardening practices in the two regions. Further, this difference may explain the higher measure values for SVANN-I models trained in Fulton County compared to the models trained in Hennepin County, because detection of raised beds is less challenging due to their distinct boundaries. In terms of spatial variability, we found that gardens differed in their texture across the two regions. In particular,

Table 3. Comparison Results Between SVANN-I and OSFA

Approach	Training Area	Test Area	Precision	Recall	F1-Score
SVANN-I	Hennepin	Hennepin	**0.794**	**0.419**	**0.549**
OSFA	All	Hennepin	0.713	0.341	0.461
SVANN-I	Fulton	Fulton	**0.924**	**0.674**	**0.779**
OSFA	All	Fulton	0.886	0.618	0.728
SVANN-I (Two zones)	All (trained by zone)	Hennepin + Fulton	**0.836**	**0.485**	**0.614**
OSFA	All	Hennepin + Fulton	0.771	0.412	0.537



(a) Sample urban gardens in Hennepin County, MN

(b) Sample urban gardens in Fulton County, GA

Fig. 7. Spatial variability in the dataset. As shown, the backyard urban gardens in Fulton County, Georgia, have greener surroundings compared to the backyard urban gardens in Hennepin county, Minnesota.

gardens in Fulton County, Georgia, had a higher green cover as compared to Hennepin County, Minnesota. Figure 7 shows the spatial variability in the urban gardens across the two counties. In summary, SVANN-I was better in modeling spatial variability due to differing garden beds and surroundings across the two regions.

3 SVANN-E: A GENERAL APPROACH

All the object models in the urban garden evaluation were built using the same neural network architecture. Neural network architectures are highly flexible in their ability to learn object characteristics; however, spatial variability may require learning location-based characteristics for better understanding. Further, different architectures may be suitable for different learning tasks. Thus, we propose a more general SVANN approach (SVANN-E) where the model architecture can vary at different locations.

SVANN-E is a spatially explicit model where neural network architecture is a function of model location loc_M. The architecture f_{loc_M} is a sequence of K weight functions or layers mapping a geographic location based training sample $x(loc_S)$ to a geographic location dependent output $y(loc_M)$ as follows:

$$y(loc_M) = f_{loc_M}(x(loc_S); w^1(loc_M), \ldots, w^K(loc_M)), \tag{3}$$

where $w^i(loc_M)$ is the weight vector for the i^{th} layer. Further, as the architecture varies across locations, it implies that the weights are also a function of location. Lemma 3.1 formally describes the relationship between SVANN-I and SVANN-E.

Lemma 3.1. *SVANN-I is a special case of SVANN-E.*

Proof. When f_{loc_M} in SVANN-E is location invariant, the approach reduces to SVANN-I. □

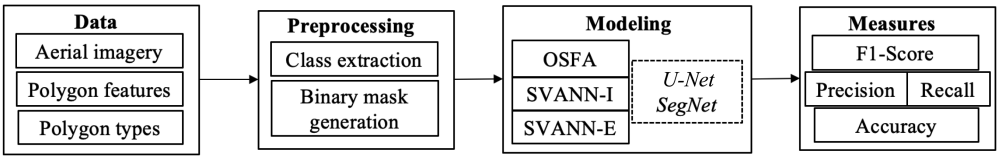


Fig. 8. Experiment design used in the evaluation.

Corollary 3.1. *Classification accuracy of SVANN-I can never be greater than the classification accuracy of SVANN-E.*

Proof Sketch. The corollary follows from the fact that SVANN-I is a special case of SVANN-E.

4 EVALUATION FRAMEWORK

This section details the evaluation framework for the proposed SVANN-E approach. We explain the experiment design, evaluation task, evaluation metric, and network architectures used to build the models. We then describe the dataset and pre-processing steps including the computing resources used for experiments.

4.1 Experiment Design

Since our goal here was a proof of concept, we limited our experiments to the special case of training approaches. We trained individual models for two disjoint and distant geographic regions (i.e., located in Hennepin County, Minnesota, and Miami-Dade County, Florida). Since counties have rigid boundaries, this is a base case of fixed-partition-based neighbors where the number of partitions is 2. Overall, we trained and compared six models categorized by their architecture and training region. Model 1 and Model 2 were composed of the U-Net architecture and were trained separately on the data from Hennepin County and Miami-Dade County. Model 3 and Model 4 were composed of the SegNet architecture and were trained separately for each region. Model 5 (built on the U-Net architecture) and Model 6 (built on the SegNet architecture) were based on a spatial OSFA approach and were trained on imagery data from both areas together. Figure 8 shows the experiment design used in the evaluation.

To evaluate the SVANN-I approach, Model 1, Model 3, Model 5, and Model 6 were evaluated on the Hennepin County imagery. Model 2, Model 4, Model 5, and Model 6 (OSFA) were evaluated on the Miami-Dade County imagery. The best-performing SVANN-I model for each region (Model 1 or Model 3 for Hennepin; Model 2 or 4 for Miami-Dade) were selected based on their validation accuracy. These models then became the testing models used to evaluate the SVANN-E approach against OSFA on the same datasets. We also evaluated combined models 1 and 2, combined models 3 and 4, and OSFA (i.e., models 5 and 6) on the complete dataset. Table 4 shows the set of comparisons to assess spatial variability. Further detail on the architectures is provided in later sections (Section 4.4).

4.2 Evaluation Task Definition

For evaluation, we built neural network models for wetland mapping using aerial data from two different locations. A wetland refers to a flooded area of land having a distinct ecosystem based on hydrology, hydric soils, and vegetation adapted for life in water-saturated soils [15]. Wetland inventory maps are essential for their management, protection, and restoration. However, development of highly accurate wetland inventories can be expensive and technically challenging. Further, they require periodic updates due to seasonal changes, land use change, and climate change. Decreases in funding to programs such as the **National Wetlands Inventory (NWI)** have led to

Table 4. Assessment of Spatial Variability

		Training Area	Test Area
SVANN-I (U-Net)	Model 1	Hennepin, MN	Hennepin, MN
	Model 2	Miami-Dade, FL	Miami-Dade, FL
	Model 1, Model 2		Hennepin, MN + Miami-Dade, FL
SVANN-I (SegNet)	Model 3	Hennepin, MN	Hennepin, MN
	Model 4	Miami-Dade, FL	Miami-Dade, FL
	Model 3, Model 4		Hennepin, MN + Miami-Dade, FL
SVANN-E	Model 1 or Model 3	Hennepin, MN	Hennepin, MN
	Model 2 or Model 4	Miami-Dade, FL	Miami-Dade, FL
	Model 1 or Model 3, Model 2 or Model 4		Hennepin, MN + Miami-Dade, FL
OSFA (U-Net)	Model 5	Hennepin, MN + Miami-Dade, FL	Hennepin, MN
			Miami-Dade, FL
			Hennepin, MN + Miami-Dade, FL
OSFA (SegNet)	Model 6	Hennepin, MN + Miami-Dade, FL	Hennepin, MN
			Miami-Dade, FL
			Hennepin, MN + Miami-Dade, FL

Comparison 1
Comparison 2
Comparison 3

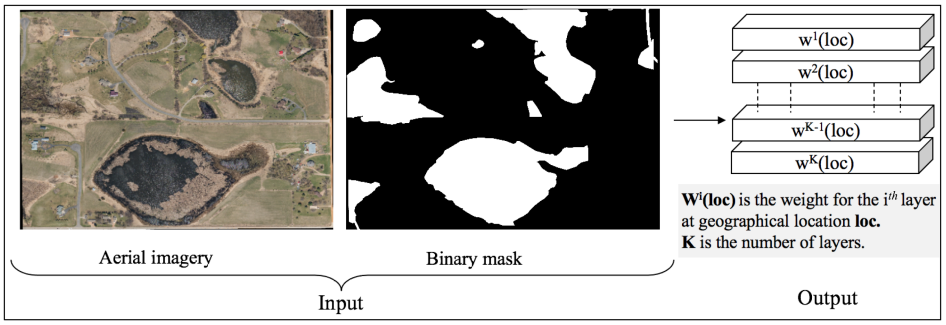


Fig. 9. Example input and output for the SVANN-E evaluation task.

changes in the procedure of wetland mapping from costly manual photo-interpretation [37] to multi-fusion semi-automated approaches [17]. In semi-automated approaches [17], the maps are produced through a combination of image segmentation and random forest classification along with aerial photo interpretation. In this work, we are not evaluating the accuracy of wetland maps constructed with a semi-automated approach; instead, we leverage the wetland maps from previous work to evaluate our approach.

We defined the task of wetland mapping as a pixel-level image segmentation process where we label the pixels of an image as a class. Similar adjacent pixels are then grouped together into larger image objects. In this work, we build binary classifiers useful for classifying a pixel as a wetland or not. Figure 9 shows the input and output of the system, where the input data is RGB aerial imagery along with a corresponding binary mask. The white region in the binary mask represents the pixels that belong to the class. Both the imagery and mask are used to train the final semantic segmentation models.

4.3 Evaluation Metrics

We used the F-1 metric and accuracy [23] to evaluate the results of the pixel-level image segmentation. The F-1 metric is a function of precision and recall, where precision is the ratio of true pixels detected to the total number of pixels predicted by the classifier, and recall is the ratio of true pixels

detected to the total number of pixels in the dataset. Overall, precision, recall, and accuracy can be written as a function of true positives, true negatives, false positives, and false negatives.

Due to inherent differences in spatial variability and class imbalance between regional datasets, as well as differences in features learned by the various CNN architectures, trained models make predictions with varying confidence levels. To accommodate these differences, we use the ROC (receiver operating characteristic) curve, similar to threshold moving, as an effective way to select an optimal prediction threshold when evaluating each model [19]. The ROC curve observes the trade-off between **true-positive rate (TPR)** and **false-positive rate (FPR)** across a complete range of threshold values for a trained model, where the TPR is the ratio of true-positive pixels to total number of positive pixels and the FPR is the ratio of false-positive pixels to the total number of positive pixels. An optimal threshold maximizes the TPR and minimizes the FPR. Since there is a trade-off between TPR and FPR, an optimal threshold is the shortest distance from a perfect predictor that minimizes the trade-off cost [5].

4.4 Architecture

We built the learning models using two architectures: U-Net [29] and SegNet [1]. U-Net is an established technique for image segmentation in biomedical images that has been adapted for aerial imagery [13]. SegNet was motivated by scene understanding applications and has also been used for aerial imagery based land cover classification [21]. Both architectures follow an encoder-decoder framework that combines local pixel information with its context. To achieve this, high-resolution features from the contracting set of layers (i.e., context) are concatenated with the output from the up-sampled images (i.e., localization). In the following, we first describe the two architectures briefly and then highlight the difference between the two.

4.4.1 U-Net. U-Net has two paths: a contracting path and an expansive path. Each step in the contracting path consists of two 3×3 convolutions (with padding), each followed by a **rectified linear unit (ReLU)** and a 2×2 maxpooling operation with a stride of 2 for down-sampling (which reduces each image dimension by half). Each step in the expansive path consists of up-sampling (i.e., up-convolution), which doubles the image dimensions, followed by a 2×2 convolution that reduces the number of feature channels by half. This is followed by concatenation with the corresponding feature map from the contracting path, and two 3×3 convolutions each followed by a ReLU. The final layer consists of a 1×1 convolution layer mapping each image pixel to the class. In addition, we use a dropout layer between each of the convolution layers to avoid over-fitting. Overall, U-Net consists of nine steps: five steps in the contracting path and four steps in the expansive path. Figure 10 shows the U-Net architecture adapted from its original work [29].

4.4.2 SegNet. In SegNet, each step in the contracting path consists of two or three 3×3 convolutions (with padding). The initial two steps have two convolutions and the later three steps have three convolutions. This type of architecture allows SegNet to reuse weights trained on the VGG16 network [34]. However, for this work, we do not reuse any pre-trained weights (i.e., transfer learning) and leave that as an exercise for future work. After convolution, features are batch normalized, followed by a ReLU and a 2×2 convolution that reduces the number of feature channels by half. Each step in the expansive path is symmetric to its corresponding contracting path step. The final layer consists of a softmax layer, which is a 1×1 convolution layer mapping each image pixel to the class. Figure 11 shows the SegNet architecture adapted from its original work [1].

The two architectures differ as U-Net transfers the entire feature map from the contracting step to its corresponding expansive path, whereas SegNet transfers pooling indices that are concatenated to the up-sampled decoder feature maps. Due to its use of feature indices over the entire (pooled) feature map, SegNet is memory efficient but incurs a slight loss in accuracy. The two

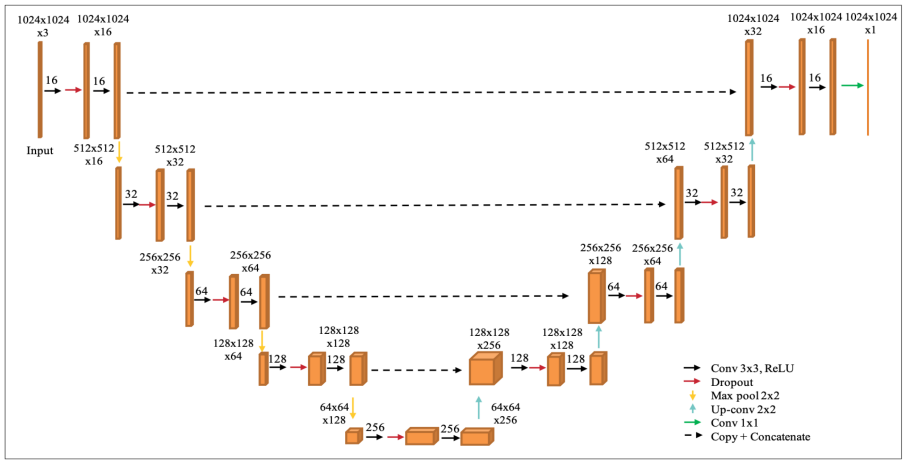


Fig. 10. Architecture of U-Net where arrows represent the operations (i.e., convolution, maxpool, dropout, and copy and concatenation). Feature dimensions are shown above the features, and the number of channels in the convolution operation are shown above the arrows.

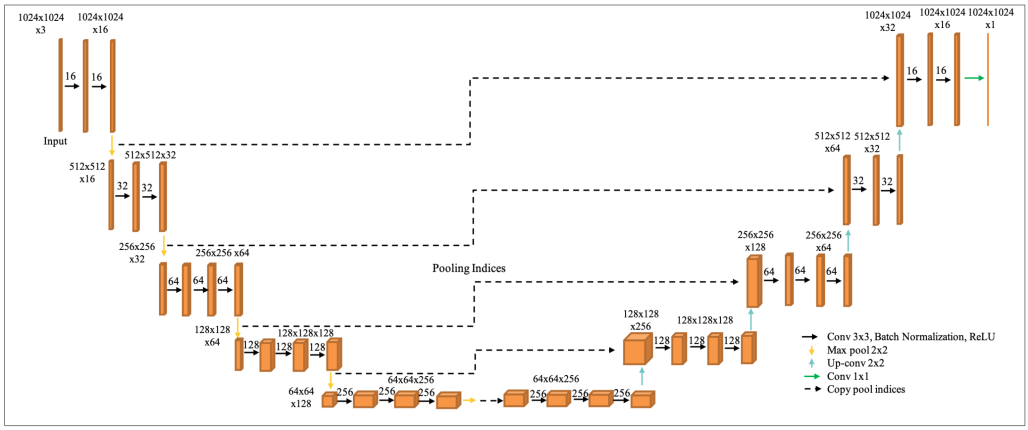


Fig. 11. SegNet architecture where the straight arrows represent the operations (i.e., convolution, maxpool, and copy and concatenation), and the elbow connector arrow shows pooling indices used along with up-sampled features. Feature dimensions are shown above the features, and the number of channels in the convolution operation are shown above the arrows.

architectures differ further in the number of convolution layers (U-Net has 18 and SegNet has 26 convolution layers), use of dropout layers, and batch normalization. Dropout layers help to reduce over-fitting in U-Net, and batch normalization helps improve training stability during SegNet-based model learning.

4.5 Dataset

We used high spatial resolution aerial imagery and part of a set of wetland maps developed by the NWI for the task of wetland mapping [7]. The imagery had red, green, and blue bands and a spatial resolution of 3 inches. The imagery in Hennepin County was acquired between May 4, 2018

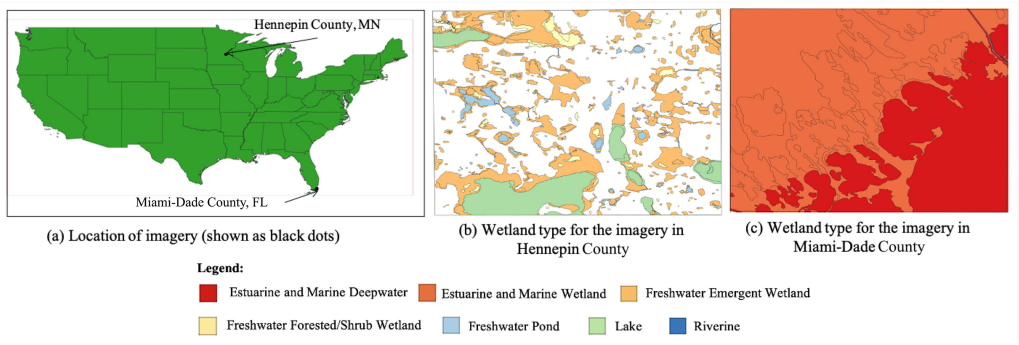


Fig. 12. Location and types of wetland in the imagery.

and May 18, 2018, and the imagery in Miami-Dade County was acquired between March 22, 2019 and March 23, 2019. The wetland maps consist of more than 34 million features that represent the extent, approximate location, wetland types, and surface water habitats in the United States and U.S. trust territories. In this work, we limited our analysis to two study areas of 23.12 and 22.68 km^2 located on the western border of Hennepin County, Minnesota, and the southern border of Miami-Dade County, Florida, respectively. The two areas lie at a straight line distance of around 2,489 km from each other. Figure 12(a) shows the location of study areas marked by black dots.

The initial wetland map dataset from NWI was in a shapefile format with wetlands mapped for Southern Minnesota that have 74,018 features and Florida that have 1,037,009 features (these were the smallest shapefiles that overlapped the area of study). All the features were classified according to the Cowardin classification system [3]. To align and limit the analysis to the study area, the features and imagery were first re-projected to the WGS84 reference system. Next, the shapefile was cropped to the extent of the imagery region. Overall, the study regions had seven type of wetlands: Estuarine and Marine Deepwater, Estuarine and Marine Wetland, Freshwater Emergent Wetland, Freshwater Forested/Shrub Wetland, Freshwater Pond, Lake, and Riverine. Figure 12(b) and (c) show the wetland type in the imagery for the two locations. A later section (Section 5.2) describes the pixel distribution across different wetland types for the two regions to help assess the effect of spatial variability.

4.5.1 Pre-Processing. The default dimensions of the image tiles provided by the counties were 10,630 × 10,280 from Hennepin County and 5,000 × 5,000 for Miami-Dade County. To train and test the models, we partitioned the imagery (e.g., Figure 13(a) and (b)) and its mask into images having the dimensions of 2,048 × 2,048 (Figure 13(c)), which were resized into tiles of size 1,024 × 1,024 (Figure 13(d)). Resizing the images from 2,048 × 2,048 to 1,024 × 1,024 tiles did not have any significant impact on the model accuracy and resulted in a 4× reduction in model training time. These steps resulted in 792 and 624 samples for the Hennepin and Miami-Dade regions, respectively. Partitions at the edge of the imagery were removed, as a majority of the area was empty.

The binary masks for training and testing were created in two steps. First, we used the polygons in the shapefile to extract the wetland from the imagery for each region. We then used the extracted wetland imagery to create the binary mask that would be used for training. Figure 13(a) shows the initial imagery from Hennepin County, Figure 13(e) shows the extracted wetland imagery for each image partition, and Figure 13(f) shows the corresponding binary mask where white pixels represent wetlands. The dataset was divided into training (~80%), validation (~10%), and testing (~10%) datasets to build, validate, and test the models, respectively.

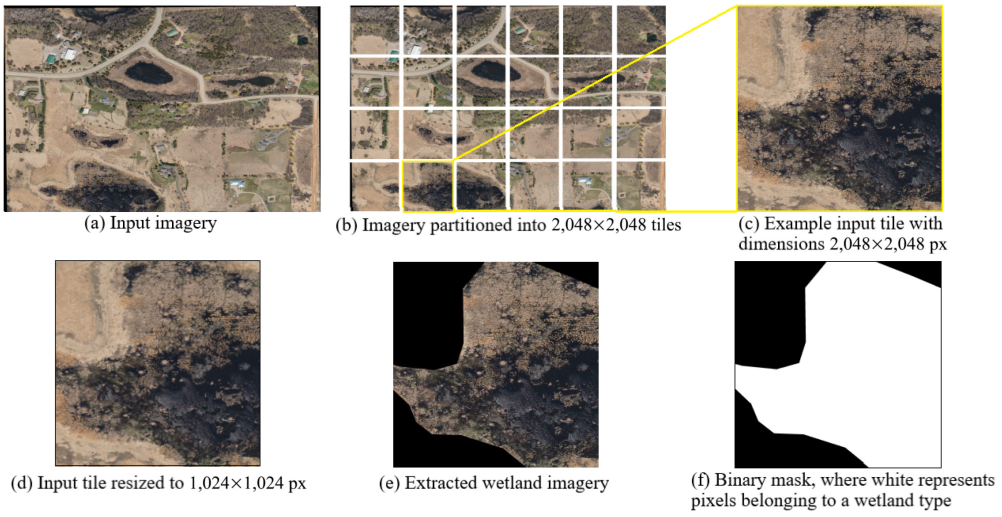


Fig. 13. Pre-processing imagery and polygon-based wetland maps.

Resources. We used Python's geopandas, rasterio, PIL, and numpy libraries for pre-processing the imagery and shapefiles. We used Keras, a high-level deep learning interface for TensorFlow, to implement U-Net and SegNet. Model training and evaluation was done on a 3.70-GHz Intel Core i7-8700K CPU, NVIDIA GeForce GTX 1080Ti GPU, and 16 GB of 2,400-MHz DDR4 RAM. Python-based implementation of SVANN can be found at the following link: https://github.com/jayantgupta/SVANN.

Imagery for Miami-Dade County was provided by the Florida Department of Transportation[1] over email correspondence, and imagery for Hennepin County was provided by the Minnesota Geospatial Information Office that is hosted at USpatial servers[2] accessible over File Transfer Protocol.

5 EXPERIMENTAL RESULTS

This section presents our spatial variability assessment results and spatial variability based interpretation.

5.1 What Is the Effect of Spatial Variability?

To assess the effect of spatial variability on the performance of training models, we conducted three sets of comparisons. As shown in Table 4, the SVANN-E approach that can vary in model architecture demonstrated better performance by choosing better regional models trained using the SVANN-I approach. For example, SVANN-E selected the U-Net model in the Hennepin region and selected the SegNet model in the Miami-Dade region. Further, both SVANN-I approaches performed better than OSFA on all the measures (precision, recall, F1-score, and accuracy) for all three comparisons. The models tested on the Hennepin imagery had lower F1-scores and accuracy than the models trained on the Miami-Dade imagery. Further, the accuracy values in the region for Model 2 and Model 4 are very similar. This can be attributed to the large number of true positives in the region (~99%). A true reflection of the performance could be estimated by observing the

[1] https://www.fdot.gov/gis/aerialmain.shtm.
[2] https://research.umn.edu/units/uspatial/.

Table 5. Comparison Results Between SVANN-E, SVANN-I, and OSFA

Approach	Model (M)	Arch.	Precision	Recall	F1	Test Acc.	Val Acc.
Test Area: Hennepin County							
SVANN-I	**Model 1**	**U-Net**	**0.516**	**0.614**	**0.561**	**65.3**	**66.41**
SVANN-I	Model 3	SegNet	0.402	0.528	0.456	52.74	61.86
SVANN-E	**arg max$_{Val_Acc}$(M1, M3)**	**U-Net**	**0.516**	**0.614**	**0.561**	**65.3**	**66.41**
OSFA	Model 5	U-Net	0.503	0.48	0.491	60.6	62.7
OSFA	Model 6	SegNet	0.361	0.498	0.418	50.06	37.55
Test Area: Miami-Dade County							
SVANN-I	Model 2	U-Net	0.999	0.978	0.988	97.76	99.10
SVANN-I	**Model 4**	**SegNet**	**0.999**	**0.984**	**0.992**	**98.34**	**99.42**
SVANN-E	**arg max$_{Val_Acc}$(M2, M4)**	**SegNet**	**0.999**	**0.984**	**0.992**	**98.34**	**99.42**
OSFA	Model 5	U-Net	0.993	0.599	0.747	59.91	67.51
OSFA	Model 6	SegNet	0.994	0.708	0.827	70.7	72.1
Test Area: Hennepin County, Miami-Dade County							
SVANN-I	Model 1, Model 2	U-Net	0.825	0.863	0.843	79.6	–
SVANN-I	Model 3, Model 4	SegNet	0.765	0.836	0.799	72.83	–
SVANN-E	**Model 1, Model 4**	**U-Net,SegNet**	**0.836**	**0.877**	**0.856**	**80.15**	**–**
OSFA	Model 5	U-Net	0.781	0.560	0.653	61.47	–
OSFA	Model 6	SegNet	0.695	0.642	0.667	59.16	–

false negative results in the region as follows: Model 2 (1,428,316), Model 4 (1,030,699), Model 5 (26,215,973), and Model 6 (19,086,201). We can observe that OSFA (Model 5 and Model 6) has a significantly high number of false negatives, which is also evident from the models' F1 score. More importantly, Model 2 has around 400,000 more false negatives than Model 4, which clearly shows that SegNet models are performing better than U-Net models in the Miami-Dade region.

5.2 Spatial Variability Based Interpretation

The Hennepin and Miami-Dade regions have a different distribution of wetlands (Figure 12). Table 6 shows the distribution of wetland types in the two regions by number of pixels. The study region in Hennepin County is dominated by Freshwater Emergent Wetlands (860,403,782 pixels), which mostly consist of perennial plants, and by Lakes (371,289,247 pixels). In contrast, the Miami-Dade study region is dominated by Estuarine and Marine Deepwater (2,494,742,829) and Estuarine and Marine Wetlands (1,386,268,631). Estuarine refers to deepwater tidal habitats that are adjacent to tidal wetlands, whereas marine refers to open ocean and its associated high-energy coastline [25]. The two regions therefore differ in the composition and density of their wetlands resulting in high spatial variability across the two regions. The greater variability in wetland types in Hennepin County likely contributed to the lower accuracy. With only two classes, the situation in Miami is much simpler. Thus, SVANN-E is a better choice than SVANN-I due to its ability to have location-dependent architectures, whereas OSFA's rigid approach makes it unsuitable for use cases showing such spatial variability.

6 DISCUSSION

6.1 SVANN Taxonomy

The degree of spatial variability can change depending on the underlying phenomenon. For example, soil samples collected in a 100 × 100-m grid to assess their organic carbon content can vary significantly depending on factors such as tillage, soil composition, vegetation, and land management practices [2]. Table 7 shows a taxonomy of SVANNs classified by the spatial variability in their neural network architecture and weights. Since weights from one architecture cannot be

Table 6. Distribution of Pixels Based on Wetland Type

Wetland Type	No. of Pixels
Study Area: Hennepin County	
Freshwater Forested/Shrub Wetland	89,888,186
Freshwater Emergent Wetland	860,403,782
Freshwater Pond	78,883,844
Riverine	4,283,724
Lake	371,289,247
Non-Wetland	2,447,959,167
Total Wetland	1,404,748,783
Study Area: Miami-Dade County	
Estuarine and Marine Wetland	2,494,742,829
Estuarine and Marine Deepwater	1,386,268,631
Non-Wetland	38,935,167
Total Wetland	3,881,011,460

Table 7. A SVANN Taxonomy by Spatial Variability in Neural Network Architectures and Weights

Architecture		Weights	
		No Spatial Variability	Spatial Variability
	No spatial variability	OSFA	SVANN-I
	Spatial variability		SVANN-E

used for another architecture, one out of the four categories in the taxonomy is not relevant (i.e., cases with spatial variability in architecture and no spatial variability in weights). Here we discuss the SVANN taxonomy and provide examples based on the modeling of streamflow in rice fields (Figure 14).

OSFA represents cases with no spatial variability in weights and architecture across geographic region(s), and thus cases where a single model can be used to represent all the region(s). An example is a single streamflow model built using flat rice fields (e.g., rice fields in Texas) but used globally. OSFA assumes that similar factors affect the streamflow across all the regions and cannot address the challenges of spatial variability. However, OSFA models can serve as a common baseline to compare different modeling techniques. In addition, model accuracy can be improved by using richer features and using data and methods for pre-processing and post-processing.

The second category, SVANN-I, represents cases where the architecture does not have spatial variability but the weights do vary spatially. Such cases require the use of separate weights calibrated for each spatially varying region (i.e., SVANN-I). The approach is analogous to interpolation (I) of data within each spatially varying region. It is assumed that the phenomenon, function, or object of interest (being modeled) is governed by similar factors across the regions. An example is building separate streamflow models for flat rice fields in Texas, USA, and stepped rice fields in Chiang Mai, Thailand (shown as the dashed green box in Figure 14). The models assume that similar factors (represented by the architecture) affect the streamflow to varying degrees. Additional examples can include the analysis of soil samples at a study area by building different models with similar structure, number of parameters depending on the type (e.g., sand, clay), or the composition of soil.

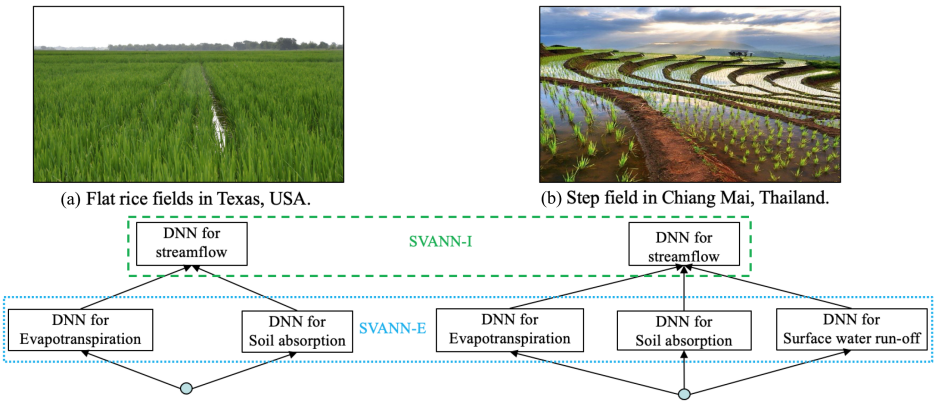


Fig. 14. Pre-processing imagery and polygon-based wetland maps.

Different SVANN-I training (Section 2.1) and prediction (Section 2.2) procedures can be adapted to different levels of variability within the data. For example, a K-nearest neighbor based approach can be used for the study area with smooth transition between two spatially distinct regions. In contrast, fixed-partition-based neighbors and distance-based nearest neighbors can be used for regions that show a sharp transition across the regions. Neural networks with a distance-weighted learning rate utilize the flexibility of neural networks to address different levels of spatial variability (low to high) across the regions.

Finally, SVANN-E represents cases where both the architecture and weights vary spatially. These cases require selection of architecture suitable for each spatially distinct location. SVANN-E is analogous to extrapolation (E) of data outside each spatially varying region. It is a highly flexible approach that can model a phenomenon governed by different factors at different locations. For example, a streamflow model for flat rice fields may consider factors such as evapotranspiration and soil absorption, whereas a streamflow model for stepped rice fields may consider additional factors such as surface water run-off.

As is evident from the preceding discussion, SVANN-E is the most flexible approach to model spatially varying phenomena, functions, or objects of interest compared to SVANN-I and OSFA. The flexibility comes at a higher cost in terms of computational complexity, time, and storage resources. However, with the advances in cost-efficient computing and memory and time-efficient software libraries, SVANN-based approaches can be the preferred option over OSFA approach.

6.2 Physics-Inspired Interpretation

We also provide a physical science based interpretation of the SVANN taxonomy to help relate different types of SVANN techniques with existing physical science techniques. Instead of referring to neural network architecture and weights, we use the analogy of model and parameters and classify the methods used under varying uncertainty. Table 8 shows a classification of physical science methods based on the model-parameter levels of uncertainty. The table shows three relevant categories based on the level of uncertainty in the model and parameters.

When there is no uncertainty in a model or parameter, the result is location-independent models such as gravity, or Newton's laws assuming that the model relates to processes on Earth at greater than microscopic scale. Other examples include wind tunnel experiments and scaling techniques used in aircraft design. From spatial statistics, spatial auto-regression (SAR) is also an example where regression-based analysis is performed using a linear combination of weights.

Table 8. Model-Parameter Uncertainty Based Classification of Physical Science Methods

Model		Parameter	
		No Spatial Variability	Spatial Variability
	No spatial variability	Location independent	Interpolation
	Spatial variability		Extrapolation No two places on Earth are alike

Interpolation methods are used to model processes when there is no model uncertainty and low or high parameter uncertainty. For example, kriging-based methods assume uniform sampling and a Gaussian process, and thus are suitable where a parameter has low uncertainty, whereas nearest neighbor methods are useful where parameters have high uncertainty. In the case of low uncertainty for both model and parameter, we can partition or cluster the data where each partition represents a similar process and then interpolate or recalibrate the model within each cluster.

Estimation of cases with high uncertainty in both the model and parameter require the use of extrapolation methods. Examples are computation of R0 in the SEIR model or inverse problems to find the coefficients of the Navier Stokes equation for estimating wind drag for a new vehicle. In spatial statistics, one example is GWR. More extreme forms of extrapolation would require revision of the set of processes. For example, genetic modification of seeds would require transfer of a trait from one species to another.

6.3 Other Observations

The OSFA model vs. SVANN. Given sufficient training samples and computational resources, SVANN can provide better accuracy over spatial OSFA models. Indeed, extreme cases of training a singular model may exhibit Simpson's paradox [39], where global behavior may differ from local behavior.

SVANN and the number of training samples. SVANNs need more training samples than OSFA models to capture location-specific features. However, spatial big data technologies [32] provide a wealth of spatial data with opportunities to develop SVANN. Furthermore, citizen science [33] provides ways where broader participation from scientists and volunteers can help generate relevant training data.

Computational challenges. The number of weights in a SVANN depends on the size of the network, number of locations, and the number of samples. This adds to the existing high computational cost of deep learning frameworks.

Parametric vs. nonparametric. A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a *parametric model.* In contrast, the number of parameters in non-parametric models is dependent on the dataset [30]. In general, a SVANN can be a non-parametric model if the number of locations is not constrained. However, in special cases, locations may be constrained to a fixed number of zones (e.g., U.S. states, countries) to create parametric SVANN models.

Using SVANN to assess spatial variability in a phenomenon. If OSFA and SVANN have similar performance on a task, then this implies that the phenomenon does not exhibit spatial variability. However, if SVANN outperforms OSFA, then the results support a spatial variability hypothesis in the phenomenon.

Spatial partitioning. The proposed training procedures do not require partitioning of input training samples. In fixed-partition-based neighbors training (Section 2.1.1(a)), partitions are given as input or are part of the application domain. For example, COVID-19 models are built based on political boundaries (e.g., countries). In other situations, the application domain may be willing to explore data-driven (e.g., spatial characteristics) partitioning, or the need to partition may depend on the underlying task. These topics can be explored in future work.

Transfer learning. Transfer learning is the improvement of learning on a new task (target task) through the transfer of knowledge from a related task (source task) that has been learned previously [27]. Transfer learning is used to address the data limitations in training deep neural networks. For example, transfer learning was used to improve land cover classification results, where the weights trained using one dataset were used to improve the results on a well-known UC Merced dataset [31]. Traditional transfer learning methods assume the data from two related tasks are independent and identically distributed. This is particularly relevant to spatial datasets where the independent and identically distributed assumption does not hold due to spatial auto-correlation.

6.4 Relationship to the Broader Literature

Spatial variability has also been discussed as a challenge for detecting other geospatial objects such as trees [41] and buildings [40] using remote sensing datasets. The SVANN-I approach (Section 2.1.2) is similar to GWR [8] where regression coefficients and error are location dependent. However, GWR relies on manual features to calculate model weights. In contrast, we use a multilayer CNN [20], such as YOLO [28] and U-Net [29], where initial layers perform feature engineering and later layers are responsible for prediction.

The SVANN-E approach is also related to a common practice in data mining where we first partition the data, and then develop a separate prediction model for each partition. The partitions are formed in a high-dimensional space, which may mute geographic variability. In contrast, here we use partitions in low-dimension geographic space (Section 2.1.1(a)). A similar approach was followed in the work of Jiang et al. [14], where a spatial ensemble framework was proposed that explicitly partitions input data in geographic space and uses a neighborhood effect to build models within each zone.

The taxonomy proposed in this article aligns with the organization of the spatial structures in the rainfall–run-off model [35]. For example, the Lumped structure is similar to OSFA, whereas Semi-distributed and Distributed are similar to the SVANN approach.

7 CONCLUSION AND FUTURE WORK

In this work, we investigated a spatial variability aware neural network approach (SVANN-E) that is more flexible than our previously proposed SVANN-I approach. Further, we show that SVANN-I is a special case of SVANN-E. We also provide both a SVANN taxonomy based on the spatial variability in the weights and architecture as well as an analogous physical science based interpretation of the taxonomy. We chose high spatial resolution imagery for the task of wetland mapping using an established pixel-based image segmentation technique. We evaluated the SVANN approach using imagery from two geographic areas. The experimental results show that SVANN-I outperforms OSFA and SVANN-E performs the best of all.

In the future, we plan to explore SVANNs with richer datasets (e.g., multi-spectral imagery, lidar imagery). In addition, we will evaluate the use of various indices (e.g., NDVI, NDWI) to reduce the effect of spatial variability on different models. Future work will also investigate the effect of spatial variability on multi-class classifiers. Variants of U-Net and semantic segmentation techniques

using other frameworks will be explored. Finally, for simplicity of the taxonomy, we divided the cases of uncertainty into low and high. In addition, the taxonomy will be further formalized by linguistic variables in fuzzy set theory.

ACKNOWLEDGMENTS

We would also like to thank Kim Koffolt and the spatial computing research group for their helpful comments and refinements.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2481–2495.

[2] Richard T. Conant, Gordon R. Smith, and Keith Paustian. 2003. Spatial variability of soil carbon in forested and cultivated sites: Implications for change detection. *Journal of Environmental Quality* 32, 1 (2003), 278–286.

[3] Lewis M. Cowardin. 1979. *Classification of Wetlands and Deepwater Habitats of the United States*. Fish and Wildlife Service, U.S. Department of the Interior.

[4] George Cybenko. 1989. Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2 (1989), 183–192.

[5] W. L. England. 1988. An exponential model used for optimal threshold selection on ROC curves. *Medical Decision Making* 8, 2 (1988), 120–131.

[6] Bradley J. Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L. Kline. 2017. Machine learning for medical imaging. *Radiographics* 37, 2 (2017), 505–515.

[7] U. S. Fish and Wildlife Service. 2020. *National Wetlands Inventory: Surface Waters and Wetlands*. Department of the Interior, Fish and Wildlife Service, Washington, DC. https://fwsprimary.wim.usgs.gov/wetlands/apps/wetlands-mapper/.

[8] A. Stewart Fotheringham, Chris Brunsdon, and Martin Charlton. 2003. *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. John Wiley & Sons.

[9] Yoav Freund and Robert E. Schapire. 1996. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning (ICML'96)*. 148–156.

[10] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. 2016. Deep learning for visual understanding: A review. *Neurocomputing* 187 (2016), 27–48.

[11] Jayant Gupta, Yiqun Xie, and Shashi Shekhar. 2020. Towards spatial variability aware deep neural networks (SVANN): A summary of results. In *Proceedings of the Workshop on Deep Learning for Spatiotemporal Data, Applications, and Systems (DeepSpatial'20)*.

[12] Andreas Heindl, Sidra Nawaz, and Yinyin Yuan. 2015. Mapping spatial heterogeneity in the tumor microenvironment: A new era for digital pathology. *Laboratory Investigation* 95, 4 (2015), 377–384.

[13] Vladimir Iglovikov and Alexey Shvets. 2018. TernausNet: U-Net with VGG11 encoder pre-trained on ImageNet for image segmentation. *arXiv preprint arXiv:1801.05746* (2018).

[14] Zhe Jiang, Arpan Man Sainju, Yan Li, Shashi Shekhar, and Joseph Knight. 2019. Spatial ensemble learning for heterogeneous geographic data with class ambiguity. *ACM Transactions on Intelligent Systems and Technology* 10, 4 (2019), 1–25.

[15] Paul A. Keddy. 2010. *Wetland Ecology: Principles and Conservation*. Cambridge University Press.

[16] Ronald Kemker, Carl Salvaggio, and Christopher Kanan. 2018. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (2018), 60–77.

[17] Steven M. Kloiber, Robb D. Macleod, Aaron J. Smith, Joseph F. Knight, and Brian J. Huberty. 2015. A semi-automated, multi-source data fusion update of a wetland inventory for East-Central Minnesota, USA. *Wetlands* 35, 2 (2015), 335–348.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[19] Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning* 51, 2 (2003), 181–207.

[20] Y. LeCun, Y. Bengio, and G. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436.

[21] S. Lee, S. Park, S. Son, J. Han, S. Kim, and J. Kim. 2019. Land cover segmentation of aerial imagery using SegNet. In *Earth Resources and Environmental Remote Sensing/GIS Applications X*, Vol. 11156. International Society for Optics and Photonics, 111561J.

[22] Michael Leitner, Philip Glasner, and Ourania Kounadi. 2018. Laws of geography. In *Oxford Research Encyclopedia of Criminology and Criminal Justice*. Oxford University Press, Oxford, UK. DOI: 10.1093/acrefore/9780190264079.013.325
[23] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
[24] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T. Dudley. 2018. Deep learning for healthcare: Review, opportunities and challenges. *Briefings in Bioinformatics* 19, 6 (2018), 1236–1246.
[25] Sarah M. Nusser and Jeff J. Goebel. 1997. The National Resources Inventory: A long-term multi-resource monitoring programme. *Environmental and Ecological Statistics* 4, 3 (1997), 181–204.
[26] U.S. Department of Agriculture. 2012. *USDA Plant Hardiness Zone Map*. USDA. https://planthardiness.ars.usda.gov/PHZMWeb/.
[27] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2009), 1345–1359.
[28] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7263–7271.
[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. 234–241.
[30] Stuart Russell and Peter Norvig. 2002. *Artificial Intelligence: A Modern Approach* (2nd ed.). Prentice Hall.
[31] Grant J. Scott, Matthew R. England, William A. Starms, Richard A. Marcum, and Curt H. Davis. 2017. Training deep convolutional neural networks for land–cover classification of high-resolution imagery. *IEEE Geoscience and Remote Sensing Letters* 14, 4 (2017), 549–553.
[32] Shashi Shekhar, Viswanath Gunturi, Michael R. Evans, and KwangSoo Yang. 2012. Spatial big-data challenges intersecting mobility and cloud computing. In *Proceedings of the 11th ACM International Workshop on Data Engineering for Wireless and Mobile Access*. 1–6.
[33] Jonathan Silvertown. 2009. A new dawn for citizen science. *Trends in Ecology & Evolution* 24, 9 (2009), 467–471.
[34] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[35] Jan Sitterson, Chris Knightes, Rajbir Parmar, Kurt Wolfe, Brian Avant, and Muluken Muche. 2018. *An Overview of Rainfall-Runoff Model Types*. U.S. Environmental Procection Agency, Washington, DC.
[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.
[37] Ralph W. Tiner Jr. 1990. Use of high-altitude aerial photography for inventorying forested wetlands in the United States. *Forest Ecology and Management* 33, 34 (1990), 593–604.
[38] Monica G. Turner and F. Stuart Chapin. 2005. Causes and consequences of spatial heterogeneity in ecosystem function. In *Ecosystem Function in Heterogeneous Landscapes*. Springer, 9–30.
[39] Clifford H. Wagner. 1982. Simpson's paradox in real life. *American Statistician* 36, 1 (1982), 46–48.
[40] Yiqun Xie, Jiannan Cai, Rahul Bhojwani, Shashi Shekhar, and Joseph Knight. 2020. A locally-constrained YOLO framework for detecting small and densely-distributed building footprints. *International Journal of Geographical Information Science* 34, 4 (2020), 777–801.
[41] Yiqun Xie, Shashi Shekhar, Richard Feiock, and Joseph Knight. 2019. Revolutionizing tree management via intelligent spatial techniques. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 71–74.
[42] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. 2017. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine* 5, 4 (2017), 8–36.

Received December 2020; revised April 2021; accepted May 2021