

Mapping Conservation Practices via Deep Learning: Improving Performance via Hillshade Imagery, Sampling Design, and Centerline Dice Loss

Charles J. Labuzzetta & Zhengyuan Zhu

To cite this article: Charles J. Labuzzetta & Zhengyuan Zhu (2024) Mapping Conservation Practices via Deep Learning: Improving Performance via Hillshade Imagery, Sampling Design, and Centerline Dice Loss, *Statistics and Data Science in Imaging*, 1:1, 2401756, DOI: [10.1080/29979676.2024.2401756](https://doi.org/10.1080/29979676.2024.2401756)

To link to this article: <https://doi.org/10.1080/29979676.2024.2401756>



© 2024 The Author(s). Published with license by Taylor & Francis Group, LLC.



[View supplementary material](#)



Published online: 11 Oct 2024.



[Submit your article to this journal](#)



Article views: 336



[View related articles](#)



[View Crossmark data](#)

Mapping Conservation Practices via Deep Learning: Improving Performance via Hillshade Imagery, Sampling Design, and Centerline Dice Loss

Charles J. Labuzzetta  and Zhengyuan Zhu 

Department of Statistics, Iowa State University, Ames, IA

ABSTRACT

The Iowa Best Management Practice (BMP) Mapping Project is a GIS database of conservation practices installed in Iowa's fields as seen in aerial imagery from 2007 to 2010. In this study, we explore the feasibility of using convolutional neural networks (CNNs) to automate the process of image segmentation for several conservation practices in the database: grassed waterways, pond dams, terraces, and water and sediment control basins (WASCOBs). We experiment with imagery sources, sampling methods, transfer learning, neural network architectures, and loss functions to optimize segmentation performance. Our results demonstrate that lidar-derived hillshade imagery is important for identifying structural BMPs such as pond dams, terraces, and WASCOBs. Additionally, we show that a probability-proportional-to-size random sampling method for selecting training imagery outperforms CNNs trained on imagery sampled by systematic random sampling. We also find evidence that the centerline dice loss function helps to preserve the connectedness of linear BMP features. The results of this study show it could be feasible to develop an automated method of identifying BMPs from remote sensing imagery to monitor the adoption of BMPs across the Midwestern United States.

ARTICLE HISTORY

Received May 2024
Accepted August 2024

KEYWORDS

Aerial imagery; Erosion; Nutrient runoff; Probability-proportional-to-size sampling; Semantic segmentation

1. Introduction


The 2008 Gulf Hypoxia Action Plan calls for the states along the Mississippi River to reduce nitrate and phosphorus runoff from agricultural fields. These nutrients contribute to low oxygen concentrations (hypoxia) in the Gulf of Mexico, threatening marine life and the health of an increasingly fragile ecosystem (EPA, 2008). Best management practices (BMPs) for soil and water conservation are key strategies for reducing soil erosion and nutrient runoff due to industrial agriculture (Lowrance, Dabney, and Schultz 2002). Such BMPs include structural, vegetative, and managerial practices such as grassed waterways, terraces, cover crops and strip cropping, among others (USDA 2012). Adoption of BMPs has become increasingly practiced in the Midwestern United States in order to reduce nitrate and phosphorus runoff into the Mississippi River (Schulte et al. 2008; Arbuckle 2013; Rundhaug et al. 2018). The Iowa Nutrient Reduction Strategy highlights Iowa's continued commitment to the use of BMPs in order to reduce negative environmental effects of agriculture (ISU 2012).

Despite widespread adoption of BMPs in the Midwestern United States, there are few geographic records of the use of these practices. In order to monitor compliance with the 2008 Gulf Hypoxia Action Plan, it is necessary to track the patterns of the use of BMPs over both space and time (McNeely et al. 2017). By geographically mapping the use of soil and water conservation BMPs, scientists and agricultural stakeholders may be able to

more effectively study the influence of BMPs on water quality and prioritize locations to install additional BMPs. Studies have shown that conservation efforts have the greatest influence when BMP use is targeted to address sites with disproportionately negative environmental effects (Schulte et al. 2008). These sites can be targeted with precision conservation tools such as the Agricultural Conservation Planning Framework, which can compare existing conservation efforts to full potential (Rundhaug et al. 2018). Creating an up-to-date database of existing conservation practices may assist with targeted conservation efforts to more effectively improve water quality.

The Iowa BMP Mapping Project has taken an initial step toward this goal by creating a geographic database of several BMP types used throughout the state (McNeely et al. 2017). The BMPs mapped in the geographical information system (GIS) database include terraces, water and sediment control basins (WASCOBs), grassed waterways, pond dams, contour strip cropping and contour buffer strips. LiDAR-derived products as well as natural and color-infrared aerial imagery from 2007 to 2010 were used to hand digitize the BMP layers. The creation of this baseline BMP database began in 2015 and concluded in 2019. However, continuing to rely on human labor to hand-digitize BMPs is unsustainable. There is a need to develop automated methods to increase the efficiency of monitoring BMPs in Iowa and the Midwest in order to understand the trends of the use of these practices over time.

CONTACT Charles J. Labuzzetta  clabuzzetta@gmail.com  Department of Statistics, Iowa State University, Ames, IA.

 Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JSDI.

© 2024 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

Deep learning methods, such as convolutional neural networks (CNNs), have been achieving high performance on computer vision tasks over the last decade (Krizhevsky, Sutskever, and Hinton 2012; Long, Shelhamer, and Darrell 2015). The U-Net was developed to assign a class label to each pixel in an image, segmenting images for tasks such as locating cells in biomedical imagery (Ronneberger, Fischer, and Brox 2015). Computer vision approaches have also been used for remote sensing segmentation (Mountrakis, Im, and Ogole 2011; Belgiu and Dragut 2016; Cheng and Han 2016). Deep learning approaches have become very popular in the last decade (Zhu et al. 2017). Image segmentation via CNNs has been used for remote sensing tasks such as mapping roads (Zhang, Liu, and Wang 2017), sea-land coastlines (Li et al. 2018), and land cover (Stoian et al. 2019), among many others (Isikdogan, Bovik, and Passalacqua 2017; Ji, Wei, and Lu 2019; Wu et al. 2019; Yang et al. 2019; de Albuquerque et al. 2020). In this article, we use the U-Net to segment images into BMP and background classes. Image segmentation via CNNs could help automate the process of mapping BMPs throughout Iowa and the Midwest. A previous study attempted segmentation of grassed waterways and terraces using U-Net, but several relevant and promising techniques, especially the use of LiDAR-derived products as a remote sensing data source, were not explored (Martins 2020). In this article, we present experiments exploring additional methods to improve the performance of segmenting images into BMP and background classes for four BMP types: grassed waterways, pond dams, terraces, and WASCOBs.

We study the task of mapping BMPs via CNNs through several experiments. Soil and water conservation BMPs often include both vegetative and structural components, therefore information from several remote sensing data sources could be useful to recognize these features (McNeely et al. 2017). In this article, first we experiment with incorporating color-infrared and LiDAR-derived products in our model. Transfer learning is a standard technique applied in deep learning tasks in order to efficiently train and boost model performance using the weights trained from independent modeling tasks such as ImageNet (Deng et al. 2009; Oquab et al. 2014). However, these weights come from models trained on three-channel RGB image data, so addressing the applicability to remote sensing products including both infrared and LiDAR-derived channels needs to be explored. Finally, we also experiment with a probability-proportional-to-size sampling technique compared to systematic sampling to select training data and a centerline dice loss function compared to standard dice loss (Milletari, Navab, and Ahmadi 2016; Shit et al. 2020).

The experiments presented in this article demonstrate U-Net is a promising deep-learning approach to segmenting BMPs in remote sensing imagery. Our results show that LiDAR-derived products and sampling schemes such as probability-proportional-to-size sampling improve segmentation performance. The centerline dice loss function helps to preserve the connectedness of linear features, such as grassed waterways, terraces, and WASCOBs. These results may be applicable to other vegetative and/or structural features that are identifiable in high-resolution aerial or satellite imagery, especially in highly unbalanced segmentation problems. Our models could contribute to a framework for automating the process of mapping

soil and water conservation BMPs in remote sensing imagery. Automating this process could help scientists more efficiently understand the trends of the use of these practices over time and prioritize future conservation efforts.

This article is organized in the following sections. [Section 2](#) details the datasets used in this article including the Iowa BMP Mapping Project dataset and remote sensing imagery sources. [Section 3](#) presents the sampling methods, model architecture details, data augmentation techniques, computational specifications, loss functions and metrics used in the experiments presented in [Section 4](#). Finally, [Section 5](#) discusses the results of our experiments.

2. Data

In this section, we describe the data derived from the Iowa BMP Mapping Project and provide a description of the four BMPs that were segmented by our models. Additionally, we describe the remote-sensing image layers used to identify these BMPs.

2.1. Iowa BMP Mapping Project

The Iowa BMP Mapping Project is a statewide GIS database of several BMP types that were present in fields across Iowa between 2007 and 2010 (McNeely et al. 2017). The BMPs mapped in this GIS database include terraces, water and sediment control basins (WASCOBs), grassed waterways, pond dams, contour strip cropping and contour buffer strips. LiDAR-derived products as well as natural and color-infrared aerial imagery were used to hand digitize the BMP layers. Over 1.3 million instances of grassed waterways, pond dams, terraces, and WASCOBs are included in the Iowa BMP Mapping Project database.

The polygon and line features that record the locations of these BMPs in this database can be rendered to generate reference layers of binary pixels capturing the BMP class (1) and background class (0) at 1 m resolution. For the pond dam, terrace, and WASCOB BMPs, a buffer region of 5 m. was applied around these line features to more accurately represent these practices in a raster format. These reference layers were used to train the deep learning models to recognize pixels representing BMPs according to both vegetative and structural characteristics.

2.2. BMPs

The four BMP types that we study in this article include grassed waterways, pond dams, terraces, and WASCOBs. These BMPs vary in vegetative and structural properties as well as function. See [Figure 1](#) for an example of how each of these BMPs appear in remote sensing imagery.

Grassed waterways: A grassed waterway is a graded channel ideally formed as a shallow, rounded depression between hills or in other locations where water collects as it runs off fields during rain events. Permanent grass or other appropriate vegetation helps to slow runoff in these channels as a preventative measure against gully erosion (Lowrance, Dabney, and Schultz 2002; Keep and McLoud 2012). Grassed waterways can be recognized

by their permanent vegetation in early-season color-infrared imagery, often appearing red in color compared to barren fields (McNeely et al. 2017).

Pond dams: A pond dam is either an excavated pit or dammed pool of water that helps to prevent gully erosion by collecting runoff. These ponds fill with water during rain events, which helps to protect water quality by storing runoff nutrients as well as preventing erosion (Lowrance, Dabney, and Schultz 2002; Renfro 2012). Pond dams with structural embankments are often recognizable in LiDAR-derived products. However, searching for pools of water in the color-infrared imagery before identifying the embankment with the LiDAR-derived hillshade may be a more efficient method of identification (McNeely et al. 2017).

Terraces: A terrace is a structural BMP that runs across moderate and steep slopes to intercept runoff. Used in parallel along a hill, these features help to partition long, steep slopes into a series of shorter slopes. When runoff collects on a slope behind the terrace, soil erosion, and nutrient runoff are reduced because these particles can settle before draining. Some terraces may be used in conjunction with grassed waterways (Lowrance, Dabney, and Schultz 2002; Meyer and Bracmort 2012). While sometimes recognizable in color imagery, these features are often easier to identify in LiDAR-derived products, such as a hillshade image or a high-resolution digital elevation model (McNeely et al. 2017).

WASCOBs: Water and sediment control basins (WASCOBs) are similar to terraces, but are often shorter and found perpendicular to a natural channel in a field. These features are ridged structures that trap sediment as runoff drains along a natural shallow watercourse. WASCOBs help prevent gully erosion and slow the drainage of runoff, allowing time for nutrient-rich sediment to settle (Lowrance, Dabney, and Schultz 2002). WASCOBs are best recognized using a LiDAR-derived product (McNeely et al. 2017).

2.3. Remote Sensing Datasets

The remote sensing datasets used in this article were derived from the same sources of color-infrared aerial imagery and LiDAR-derived products that were used to digitize the BMPs in the Iowa BMP Mapping Project. A 2 ft. resolution, leaf-off spring aerial photography raster layer flown between 2007 and 2010 was used to provide color-infrared imagery (ISUGIS-SRF 2018). The spring capture dates and infrared band can help to differentiate between vegetation and bare ground. Unfarmed conservation practices often sprout before crops begin to grow and this can be visualized by the infrared band's ability to highlight actively growing vegetation (McNeely et al. 2017). Additionally, a 1 m. LiDAR-derived hillshade product captured between 2007 and 2010 was used to provide relative elevation information and visualization of structural geographic features across Iowa (ISUGIS-SRF 2018). This hillshade layer was necessary to digitize pond dam, terrace, and WASCOB features (McNeely et al. 2017). It is possible that a 1 m. digital elevation model (DEM) may provide more useful geographic information, but only a 3 m. product was available for Iowa. We decided that

generating a new 1 m. DEM layer would be inefficient compared to using the available hillshade product.

Figure 1 gives an example of how these BMPs appear in the color-infrared aerial imagery, LiDAR-derived hillshade product, and the reference image derived from the Iowa BMP Mapping Project. These images were cropped along the unit boundaries of a 0.5 mi² grid superimposed over the state of Iowa and transformed into 1024 by 1024 pixel images. The color infrared image had three bands: a near-infrared band, a red band, and a green band with values ranging between 0 and 255. The values of the LiDAR hillshade product also ranged between 0 and 255 but this image had only a single grayscale band. The values of both of these images were rescaled to between 0 and 1 before model training. The reference image indicates where BMPs were located in the aerial imagery according to the Iowa BMP Mapping Project database. Pixels belonging to the BMP class were labeled with a 1, while background pixels were labeled 0. The trained models produce pixel-wise 0/1 classifications of the color-infrared and LiDAR-derived hillshade images, predicting the locations of each BMP type from these inputs. The output had the same height and width as the inputs (1024 pixels²) and the prediction performance was directly compared to each reference image.

3. Methodology

In this section, we describe the sampling methods that were used to produce the training, validation, and test sets for our experiments. Additionally, we describe the Resnet50 model architecture, data augmentation, and computational specifications that were common across all of the models in our experiments. Finally, we describe the two loss functions that we tested in our experiments and the metrics that were used to evaluate and compare our models on the test set.

3.1. Sampling

Using GIS, a 0.5 mi² grid layer was superimposed over the state of Iowa to partition the remote sensing imagery and BMP reference layers into units. For each BMP type, the units containing a nonzero amount of the BMP of interest were subset to form study populations. Two training sets (one from systematic sampling and one from probability-proportional-to-size sampling), one validation, and one test set were sampled for each BMP type (Table 1). The validation set for each BMP type was used to monitor the performance of the model during training and model selection. The performance on the test set was calculated only after all final models were selected to evaluate generalization to new data. The test set remained completely independent from all other datasets to prevent data leakage.

For the validation and test datasets, a sample of 500 units was taken separately for each BMP type via standard systematic random sampling (Fuller 2009). Prior to taking the sample, the total area of each BMP type was calculated for all units. These auxiliary data were then used to assist with the sampling procedure to form the four training, validation, and testing datasets. For each sampled unit, the corresponding BMP reference layer, color-infrared imagery, and LiDAR-derived hillshade product were

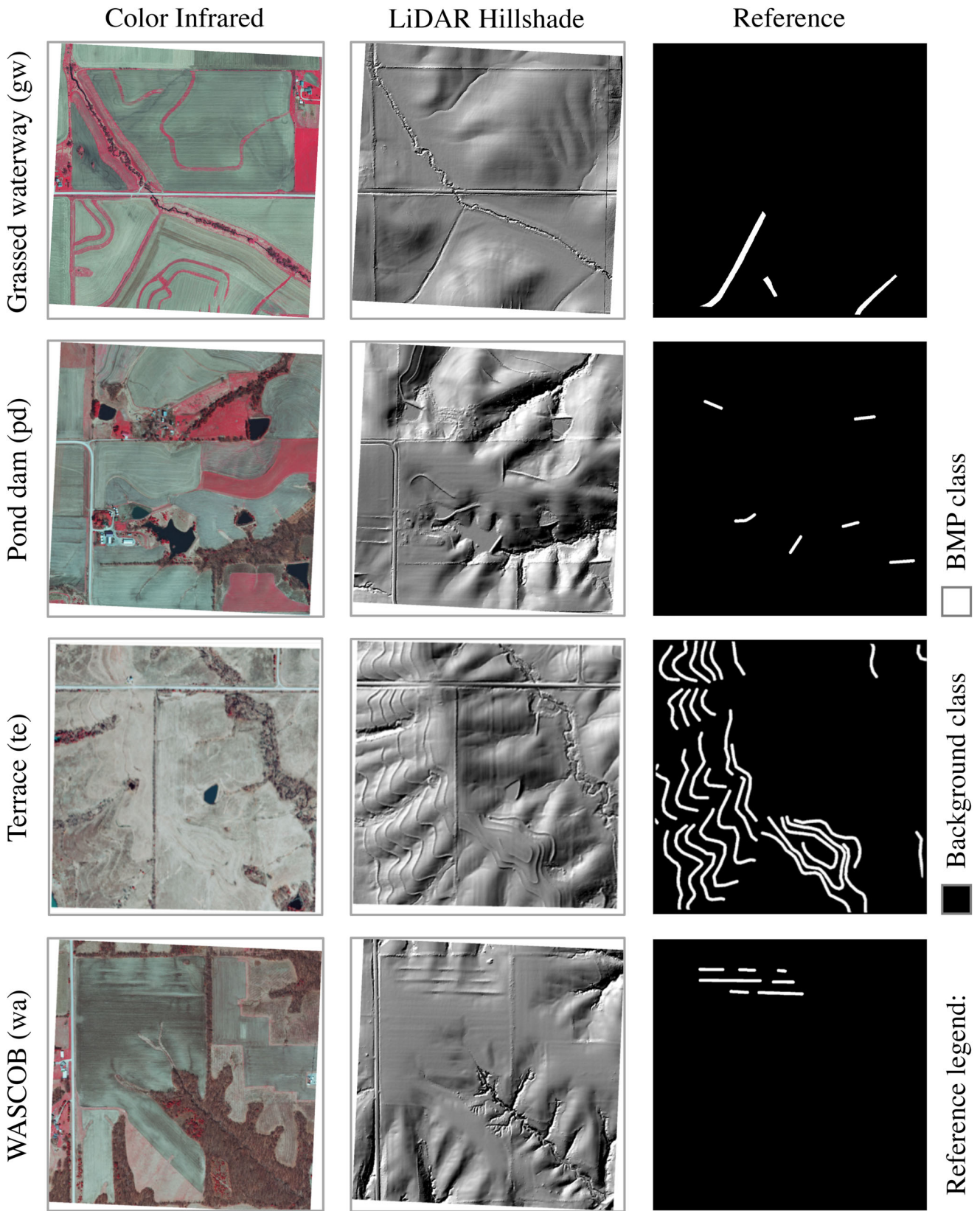


Figure 1. Color infrared, LiDAR hillshade, and reference imagery were used to train the convolutional neural network models in this article. This figure shows how the aerial images relate to the reference image for each BMP type. Each image is 1024 by 1024 pixels sampled from a 0.5 mi² grid superimposed across the state of Iowa.

clipped at the unit boundaries. These systematic samples were designed to accurately represent the population distributions of

the BMP types across Iowa to create representative validation and test sets.

Table 1. For each BMP type, the following datasets were sampled for training, validation, and testing.

Sample	Sampling method	Use	Size (units)
Training set: sys	Systematic	Train “baseline,” “lidar” and “imagenet” models	1000
Training set: pps	Probability-proportional-to-size	Train “pps” and “cldice” models	1000
Validation set	Systematic	Validation set for all models	500
Test set	Systematic	Test set for all models	500

NOTE: Two sampling methods were used to select training dataset samples. Refer to Section 3.1 for a description of these methods. Refer to Section 4 for a description of the experimental models that were trained using these training datasets.

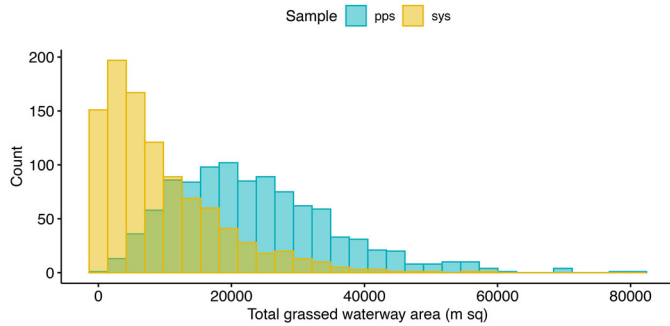


Figure 2. Distribution of total grassed waterway area within each sample of 1000 units for both the systematic (sys) and probability-proportional-to-size (pps) training datasets.

Two sampling methods were used to form experimental training datasets for each BMP type. Exactly similar to the systematic random sampling procedure described above, a systematic random sample of 1000 training units was generated for each BMP type. Additionally, a custom probability-proportional-to-size random sampling technique was used to oversample units that contained larger areas of each BMP type compared to the population.

Fuller (2009) describes probability-proportional-to-size random sampling in detail. For our application, consider a_i to be the total area of a BMP type within the i th 0.5 mi^2 unit, where $i \in 1, 2, \dots, N$. Let the probability of selecting unit i be $p_i = \frac{a_i^2}{\sum_{i=1}^N a_i^2}$. We squared a_i , the area of BMP in each unit i , to further weight the sample selection toward units with larger areas of BMP.

We hypothesized that this sampling technique would select units with more examples of each BMP type across the training sample and therefore improve performance compared to the model trained with units selected via systematic random sampling. We refer to this sampling technique as probability-proportional-to-size, “pps,” throughout this article. Histograms of the total area of grassed waterway for the units in each sample are depicted in Figure 2 to illustrate the effect of these sampling techniques.

Multiple BMP types could also be classified simultaneously within each image using a multi-class segmentation model. However, we chose to create separate samples for each BMP type because designing a sampling method that could accurately

represent the varying distribution of these BMPs across Iowa in a single training dataset would be more complex. Starting with separate datasets for each BMP type allowed us to carefully control the samples and conclude how the models performed for each BMP type independently. The results from a multi-class segmentation model would be more difficult to interpret on a per-BMP type basis. Additionally, we selected small sample sizes for each BMP type to train and test multiple experimental models in a computationally efficient manner. There are tens of thousands of 0.5 mi^2 units across Iowa that contain instances of each of these BMP types. In the future, more of these units could be sampled to create larger training sets and also produce a multi-class segmentation model. However, these small training datasets and our exploratory analysis provide a starting point that may pave the way for designing these advanced studies more efficiently.

3.2. U-Net and Resnet50 Architecture

The U-Net is recognizable by its U shape (Figure 3). Originally developed for biomedical image segmentation (Ronneberger, Fischer, and Brox 2015), this deep learning network has an “encoder/decoder” structure that encodes spatial patterns at various levels and then decodes this information to produce a segmented map of the original image. CNNs, such as the U-Net, rely on convolutional layers, which functionally combine, that is convolve, trainable sets of parameters called filters over the image to learn spatial patterns relevant to the segmented classes. The U-Net uses connections between the encoder and decoder layers to preserve high-resolution spatial patterns and output precise segmentation maps compared to fully convolutional networks.

In addition to convolutional layers, the U-Net is composed of a series of other layers and functions. After each convolution, a batch normalization layer is applied to accelerate network convergence (Ioffe and Szegedy 2015). Then, a ReLU activation function performs a nonlinear transformation, an identifying characteristic of neural networks. During the encoding process, down-sampling max-pooling layers and striding are applied to reduce the dimensions of the feature maps generated by the convolutions. The original dimensions of these feature maps are then recovered by up-sampling layers during the decoding process and connected by concatenation before additional convolution is performed throughout the decoder. Finally, a sigmoid or softmax transformation function converts the feature maps to the segmented classes.

This U-Net model was built on a Resnet50 encoder backbone (Yakubovskiy 2019). By combining the Resnet50 architecture with the U-Net model, we used a Deep Residual U-Net approach (He et al. 2016; Zhang, Liu, and Wang 2017). A “residual” network connects the original inputs before convolution to the post-convolution outputs via addition before applying the final ReLU activation function in a block. Intuitively, the network learns to minimize the residual, or the difference between the input and output of each block, in order to optimize an identity mapping. Residual networks have helped to solve the problem of vanishing gradients in very deep neural networks to improve model accuracy (He et al. 2016).

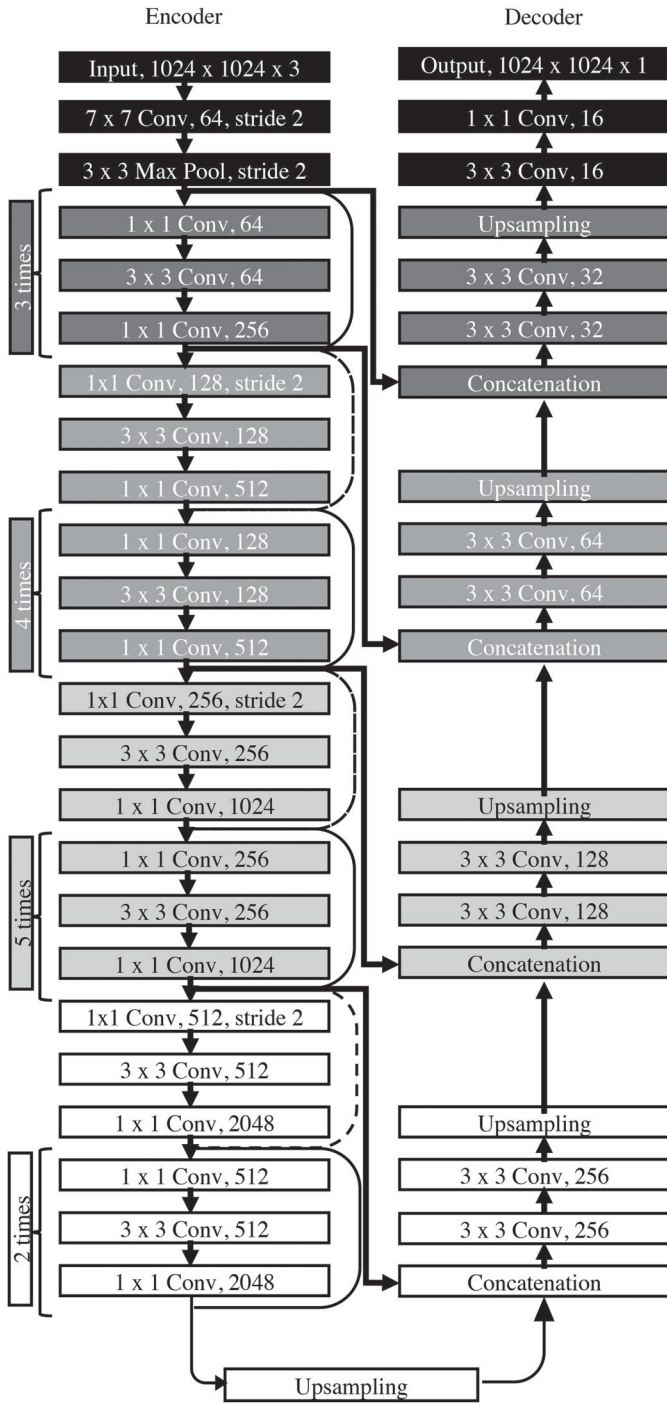


Figure 3. This Deep Residual U-Net model is composed of a ResNet50 encoder architecture and standard U-Net decoder. Residual blocks in the encoder are repeated as indicated and the output feature maps are concatenated at various points to the decoder. Each descending level of residual blocks begins with a stride 2 convolution which reduces the dimensions of the feature map. Upsampling layers expand the encoded version of the image back to its original height and width.

This U-Net model with Resnet50 architecture involves a series of convolutional, max-pooling, batch normalization, upsampling, and activation layers. Figure 3 summarizes the model. Before each convolutional layer, batch normalization and activation layers are applied. Each set of three convolutions is connected as a residual block. After the initial residual block in each level, where the dimensions of the input are halved and the number of channels are doubled, further residual blocks with

these new dimensions are included. The encoder structure of this CNN follows the Resnet50 architecture and the decoder follows the standard U-Net architecture. For up-sampling, 2 by 2 up-convolution layers are included before filters derived at the corresponding level of the encoder are concatenated. Batch normalization and activation layers are applied after each convolutional layer in the decoder. In the final layer of the network, a sigmoid activation function produces the segmentation output with the same height and width as the original input.

3.3. Data Augmentation Scheme

A technique to improve the training of deep learning models is to use data augmentation on the training dataset. Each training image may be randomly augmented via image transformations during each epoch to avoid overfitting the network to the training dataset. This helps to generalize the model to additional imagery. Data augmentation also helps to artificially increase the size of the training dataset, which is especially important in applications where there are few annotated images. The transformations commonly applied for data augmentation include shift, scale, rotation, reflection, and color transformations (Abdelhack 2020; Krizhevsky, Sutskever, and Hinton 2012). It has been shown that horizontal and vertical reflection alone may provide sufficient positional transformations for remote sensing imagery (Abdelhack 2020). Color transformations, such as those described in Wu et al. (2019), are also commonly applied to account for variable sensor and atmospheric conditions.

Following the literature, we applied horizontal/vertical reflections (Abdelhack 2020) and color augmentation (Wu et al. 2019) to the training data for all of our models. Each training image was reflected horizontally and vertically at random, with the probability of each reflection being 0.5 independently. Additionally, using random draws from uniform distributions, the hue of each original image was varied from -30 to $+30$, the saturation varied from -5 to $+5$, and the value varied from -15 to 30 (Wu et al. 2019).

3.4. Computational Specifications

Keras 2.1.0 was used to implement the models described in this article (Chollet 2015). Each model was trained on two NVIDIA Tesla V100 32 GB GPUs. Each image was resized to 1024 by 1024 pixels. The training images were grouped in mini-batches of 5 over 100 epochs. Each model was optimized via ADAM with an initial learning rate of 0.001 (Kingma and Ba 2017). The learning rate was multiplied by a factor of 0.1 when the validation loss reached a plateau with a patience of 10 epochs. Training for each model was completed in less than 8 hr.

3.5. Loss Functions and Metrics

We experimented with two loss functions. The first was the standard batch-wise dice loss function (Milletari, Navab, and Ahmadi 2016). Additionally, we tested the batch-wise centerline dice loss function introduced by Shit et al. (2020). Each BMP class populated only a very small proportion of each image. Dice loss functions have been shown to better handle class imbalance

compared to other standard loss functions such as binary cross-entropy loss (Milletari, Navab, and Ahmadi 2016). These loss functions are derived from the following scores.

Dice score: The standard Dice score (DS) evaluates the harmonic mean of precision and recall, as proposed by Dice (1945). It is considered a standard metric for comparing the similarity of two sets of images. Consider TP to be the number of correctly predicted class 1 pixels compared to the reference pixels. Similarly, TN is the number of correctly predicted class 0 pixels. Finally, FP is the number of incorrectly predicted class 0 pixels, and FN is the number of incorrectly predicted class 1 pixels. The Dice score DS can be defined as follows:

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ DS &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{aligned}$$

Centerline Dice score: The centerline Dice score (cDS) considers the intersection of skeletonized segment centerlines. As a loss function, it preserves the connectedness of predicted features, especially those linear in nature. As described in Shit et al. (2020), cDS is calculated from the reference mask M_T and predicted segmentation mask M_P . The centerline skeletons S_T and S_P are extracted from M_T and M_P , respectively. The proportion of S_P lying within M_T and the proportion of S_T lying within M_P are reported as Topology-Precision ($tPrec$) and Topology-Recall ($tRec$). Accordingly, cDS is the harmonic mean of $tPrec$ and $tRec$.

$$\begin{aligned} tPrec(S_P, M_T) &= \frac{|S_P \cap M_T|}{S_P} \\ tRec(S_T, M_P) &= \frac{|S_T \cap M_P|}{S_T} \end{aligned}$$

$$cDS(M_T, M_P) = 2 \times \frac{tPrec(S_P, M_T) \times tRec(S_T, M_P)}{tPrec(S_P, M_T) + tRec(S_T, M_P)}$$

Loss functions: Loss functions were applied in a batch-wise manner in our experiments. The Dice loss function Λ_{Dice} and $cDice$ loss function Λ_{cDice} were found using differentiable versions of the scores derived above, as detailed in Milletari, Navab, and Ahmadi (2016) and Shit et al. (2020), which we refer to as $soft_DS$ and $soft_cDS$. These two loss functions are defined as

$$\begin{aligned} \Lambda_{Dice} &= 1 - soft_DS \\ \Lambda_{cDice} &= (1 - \alpha)(1 - soft_DS) + \alpha(1 - soft_cDS) \end{aligned}$$

In our implementation, we selected $\alpha = 0.5$ to equally weight the $soft_DS$ and $soft_cDS$ scores in Λ_{cDice} . Shit et al. (2020) explored a variety of weights $\alpha \in [0, 0.5]$. Since $\alpha = 0.5$ is the strongest recommended weighting of the $soft_cDS$ score by Shit et al. (2020), we selected this value to investigate a strong effect of the $soft_cDS$ score on training for this exploratory study, but this value could be tuned to further optimize the results. However, doing so would require training the model separately for each α .

Evaluation metrics: Given the predictions from a trained model on a sample of n images, we report the average Dice score as a final metric value for each test dataset:

$$Dice = \frac{1}{n} \sum_{i=1}^n DS_i$$

We also report the average centerline Dice score ($cDice$) as a final metric value for each test dataset:

$$cDice = \frac{1}{n} \sum_{i=1}^n cDS_i$$

As additional metrics, we also refer to the image-wise average precision and image-wise average recall of the test datasets while evaluating the models in our experiments. All of these metrics by model and BMP type are reported in Table 2.

Table 2. The test set results summarized for each BMP type by model name and specification.

Experiment		Design				Results			
BMP	Model name	LiDAR	ImageNet	Sampling	Loss	Dice Score	cDice score	Precision	Recall
gw	baseline	no	no	sys	dice	0.494	0.538	0.567	0.507
gw	lidar	yes	no	sys	dice	0.520	0.573	0.596	0.532
gw	imagenet	yes	yes	sys	dice	0.515	0.569	0.580	0.536
gw	pps	yes	yes	pps	dice	0.585	0.648	0.608	0.638
gw	cldice	yes	yes	pps	cldice	0.561	0.665	0.553	0.652
pd	baseline	no	no	sys	dice	0.423	0.503	0.471	0.430
pd	lidar	yes	no	sys	dice	0.603	0.692	0.643	0.609
pd	imagenet	yes	yes	sys	dice	0.613	0.706	0.643	0.624
pd	pps	yes	yes	pps	dice	0.635	0.732	0.643	0.666
pd	cldice	yes	yes	pps	cldice	0.628	0.759	0.605	0.691
te	baseline	no	no	sys	dice	0.334	0.389	0.403	0.345
te	lidar	yes	no	sys	dice	0.548	0.623	0.609	0.557
te	imagenet	yes	yes	sys	dice	0.593	0.670	0.639	0.604
te	pps	yes	yes	pps	dice	0.623	0.704	0.656	0.639
te	cldice	yes	yes	pps	cldice	0.613	0.731	0.618	0.668
wa	baseline	no	no	sys	dice	0.178	0.214	0.228	0.198
wa	lidar	yes	no	sys	dice	0.405	0.471	0.477	0.408
wa	imagenet	yes	yes	pps	dice	0.386	0.450	0.449	0.398
wa	pps	yes	yes	pps	dice	0.485	0.554	0.513	0.535
wa	cldice	yes	yes	pps	cldice	0.475	0.582	0.470	0.556

NOTE: We report the average Dice score, $cDice$ score, precision, and recall for each model on their respective test sets. Refer to Section 4 for experiment abbreviations.

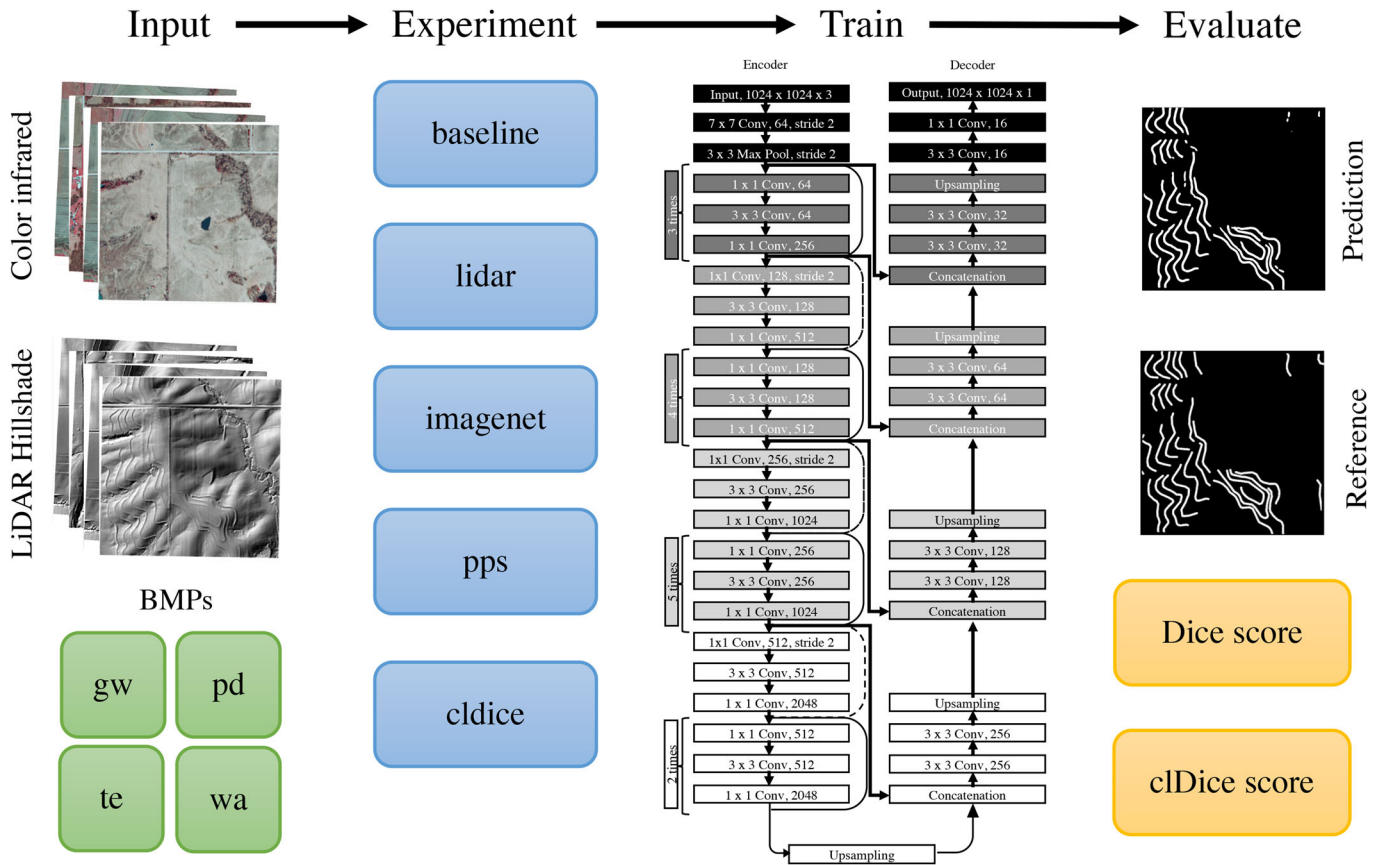


Figure 4. A visual summary of our experimental pipeline. We trained and evaluated five experimental models for each BMP type by calculating the Dice and cDice scores across all images in each training set. Refer to Section 4 for experiment abbreviations.

4. Experiments

For each of the four BMP types, grassed waterways (gw), pond dams (pd), terraces (te), and WASCOBs (wa), we implemented a set of experiments to compare the performance of various U-Net models. We began with a “baseline” model that included only color-infrared remote sensing imagery (Section 4.1). Subsequently, we included the LiDAR-derived hillshade imagery in our “lidar” model (Section 4.2) and then used pretrained weights from Imagenet in the “imagenet” model (Section 4.3). Training data sampled via probability-proportional-to-size sampling rather than systematic sampling was used to train the “pps” model (Section 4.4). Finally, the topology-preserving centerline dice loss function guided the optimization of the “cldice” model (Section 4.5).

To determine whether a model produced segmentations with higher performance than the previous model, we used a sign-test (Conover 1999) to compare the DS_i and cDS_i scores on the test set images $i \in 1, 2, \dots, 500$ from Model A to the respective scores from Model B, that is the tuple (A_i, B_i) . These scores ranged between 0 and 1 and were not necessarily normally distributed. The nonparametric sign-test is appropriate for our application because it only requires that the subjects are randomly sampled from the population and each sample is paired (Conover 1999). By randomly selecting units from the population of 0.5 mi² units across Iowa and comparing the scores of two models on each unit, we satisfied these conditions. The more powerful Wilcoxon signed-rank test (Conover 1999)

could not be used because some of the paired differences did not appear to have symmetric distributions. Under the null hypothesis for the sign-test, we assumed the distribution of the difference of scores $A_i - B_i$ had a median centered at 0. We tested the one-sided alternative hypothesis that the pairs of differences $A_i - B_i$ were centered about some median > 0 , implying that Model A had higher performance than Model B. We used the “rstatix” R package to implement this test with a Bonferroni adjustment between the five sequential models for each BMP type (Kassambara 2019).

Figure 4 summarizes this experimental pipeline visually, including the evaluation step where we compared the Dice and cDice scores on the test datasets for each experiment and BMP type. We chose to evaluate these models sequentially to reduce the number of models that would need to be trained rather than completing a factorial design. We expect that the unexplored interactions in our experimental design would either not be significant or not necessarily more informative than the order presented to merit the computational time required to complete a full factorial analysis. When we compared the models in this sequential design, the new model (Model A) was compared to the previous model (Model B) when evaluating the significance of the sign-test at the 0.05 level with the Bonferroni p -value adjustment. We report the results and significance for each of these comparisons in the supplementary materials. We also include the significance of these tests plotted above the bar for each model in Figure 5, indicating whether each

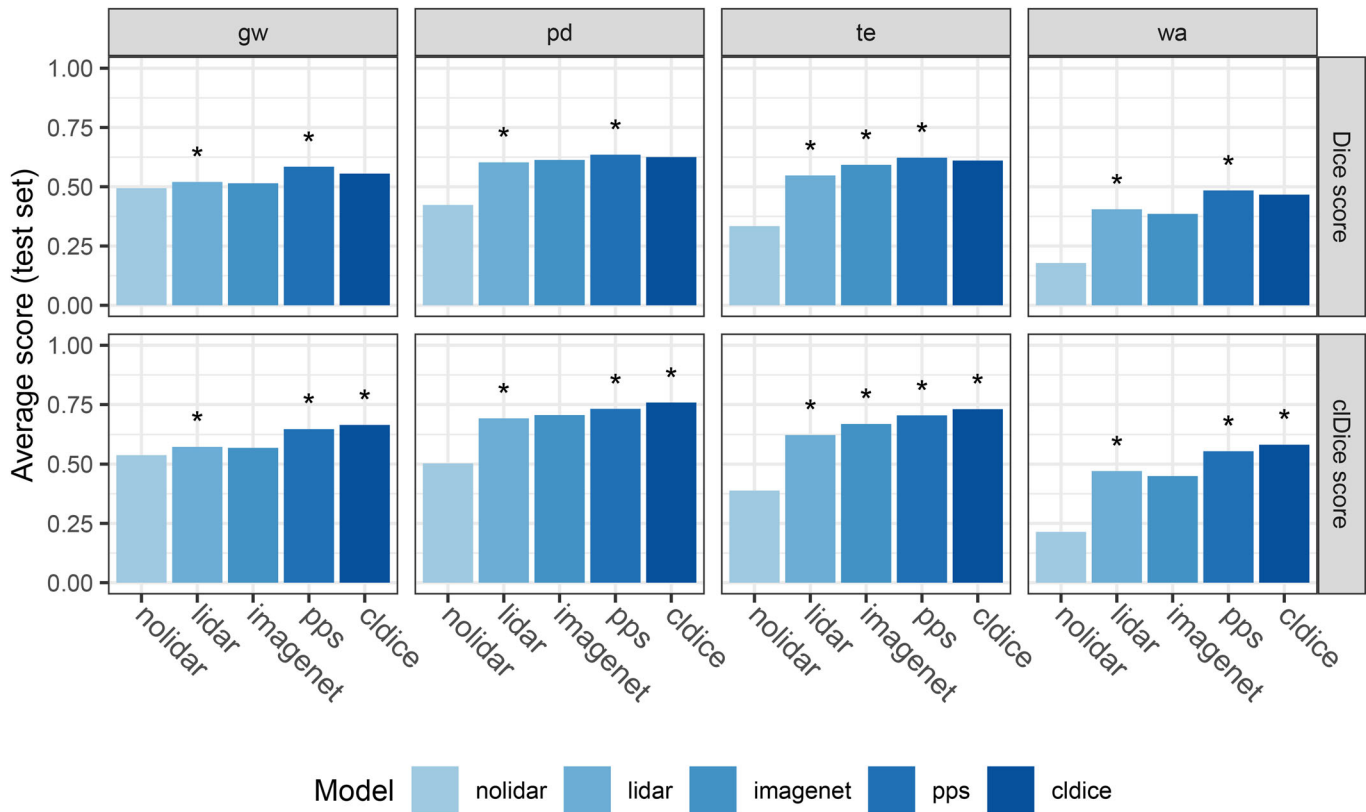


Figure 5. The average Dice and cDice scores across the test set for each model and BMP type. We used a sign-test with a Bonferroni p -value adjustment at the 0.05 significance level to test for significant improvement in model performance between models for each BMP type. An asterisk (*) above the average Dice or cDice score indicates this model had significantly higher performance on the test set compared to the model to its left. Refer to Section 4 for experiment abbreviations and descriptions.

sequential model significantly increased performance on the test set according to a sign-test.

4.1. Color-Infrared Only Imagery: Baseline

The baseline model is comparable to the U-Net model described by V.S. Martins (2020). Only color-infrared remote sensing imagery was included as training data for this model. Systematic random sampling was used to sample 1000 training units for each BMP type. In this experiment, the Deep Residual U-Net model with the Resnet50 encoder architecture was trained without ImageNet pretrained weights using the dice loss function for each BMP type. We hypothesized the baseline model would have the worst performance.

4.2. Incorporating LiDAR-Derived Hillshade: lidar

For the lidar model, we incorporated a LiDAR-derived hillshade product as a fourth channel in addition to the three-channel color-infrared imagery. The same units included in the training data samples for the baseline model were used for each BMP type. Transfer learning from ImageNet pretrained weights was not performed. The dice loss function was used to optimize this four-channel model for each BMP type. We hypothesized the performance would improve for pond dam, terrace, and WASCOB segmentation with the addition of the hillshade channel, given the structural nature of these BMPs, compared to the baseline model.

4.3. Transfer Learning with ImageNet: imagenet

Deep learning models trained on independent segmentation tasks may be applied to new tasks in a process called transfer learning (Oquab et al. 2014; Marmanis et al. 2016). The parameter weights from a previously trained network may be used to initialize the parameters in a new task with the same model architecture. Many remote sensing datasets are not as extensive as other classification datasets such as ImageNet (Deng et al. 2009). These datasets can still provide useful information via transfer learning due to their training on vast sets of images and labels (Igloukov and Shvets 2018).

Our imagenet model used transfer learning, incorporating pretrained weights from a Resnet50 encoder previously trained on the ImageNet dataset. ImageNet includes over 3.2 million images and has been used to train high-performing image classification models (Deng et al. 2009). Using the Segmentation Models Python library with Keras (Chollet 2015; Yakubovskiy 2019), we were able to load the weights from the ImageNet model into the encoder parameters in our model (the parameters on the left side of the “U” diagram in Figure 3). The remote sensing data sources, training sampling scheme, and loss function remained the same as the previously presented lidar model. To incorporate the pretrained weights in this model, which were derived from a three-channel model, we applied an extra 1×1 convolutional layer to convert the four-channel color-infrared and lidar imagery into three channels. We were uncertain whether the performance of this model would improve

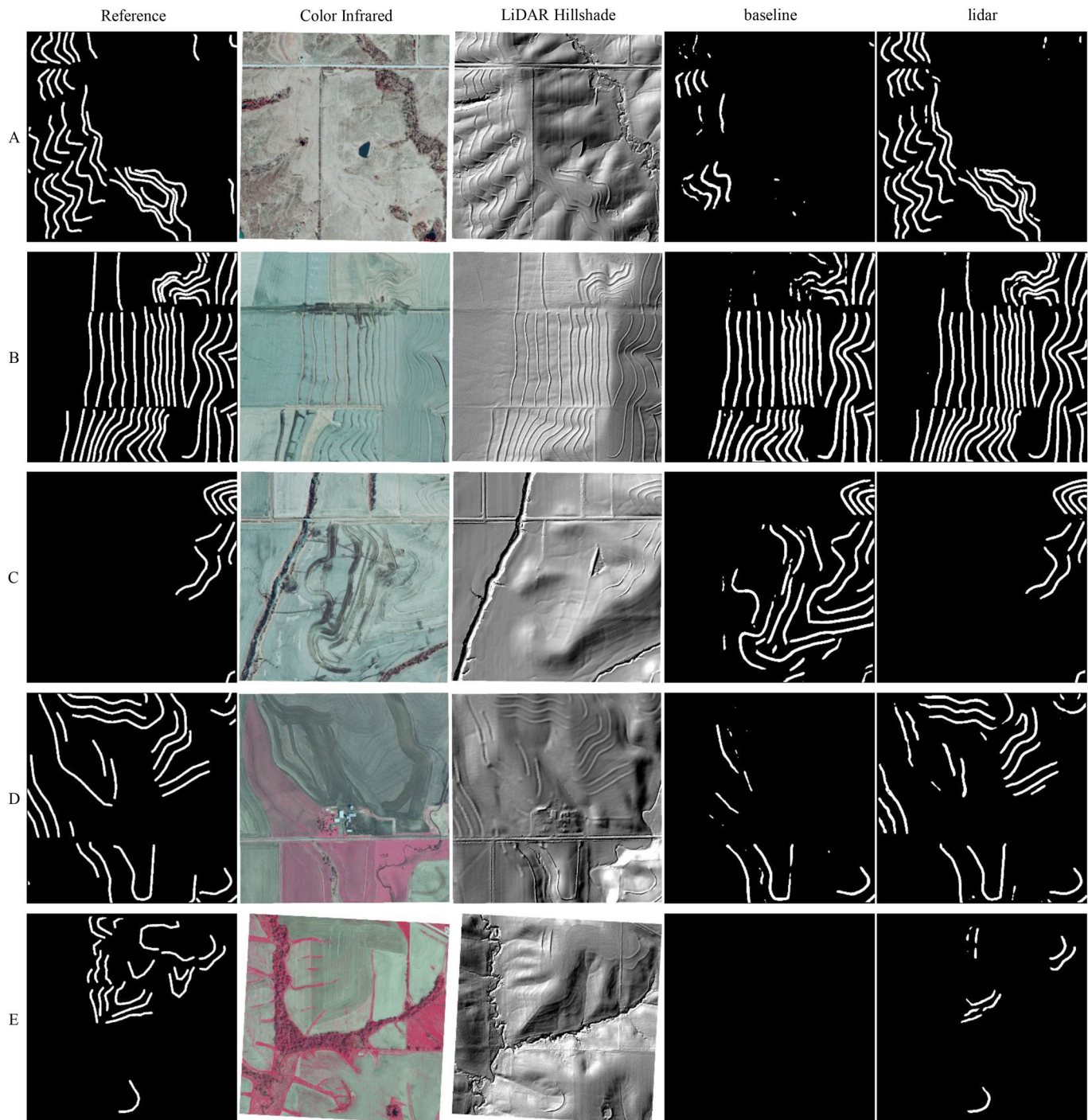


Figure 6. Terrace segmentation results for the baseline and lidar models in comparison to the reference image derived from the Iowa BMP Mapping Project. The color-infrared and LiDAR-derived hillshade images for each example are also provided. These results compare a model trained with only color-infrared imagery (baseline) to one trained also with LiDAR-derived hillshade imagery (lidar).

compared to the lidar model, but we expected the performance would at least remain similar.

4.4. Probability-Proportional-to-Size Training Data Sampling: pps

Rather than using systematic random sampling, for the pps model we used the probability-proportional-to-size random sample as described in Section 3.1 to select the training data

for each BMP type. These 1000 training units were selected to include more examples of each BMP per sampled unit on average compared to the systematic random sample. In this experiment, we included the color-infrared and LiDAR-derived channels, pretrained ImageNet weights via transfer learning, and used the dice loss function. We hypothesized the performance of this model would improve compared to the imagenet model due to the relative increase in the amount of each BMP type included in the training images.

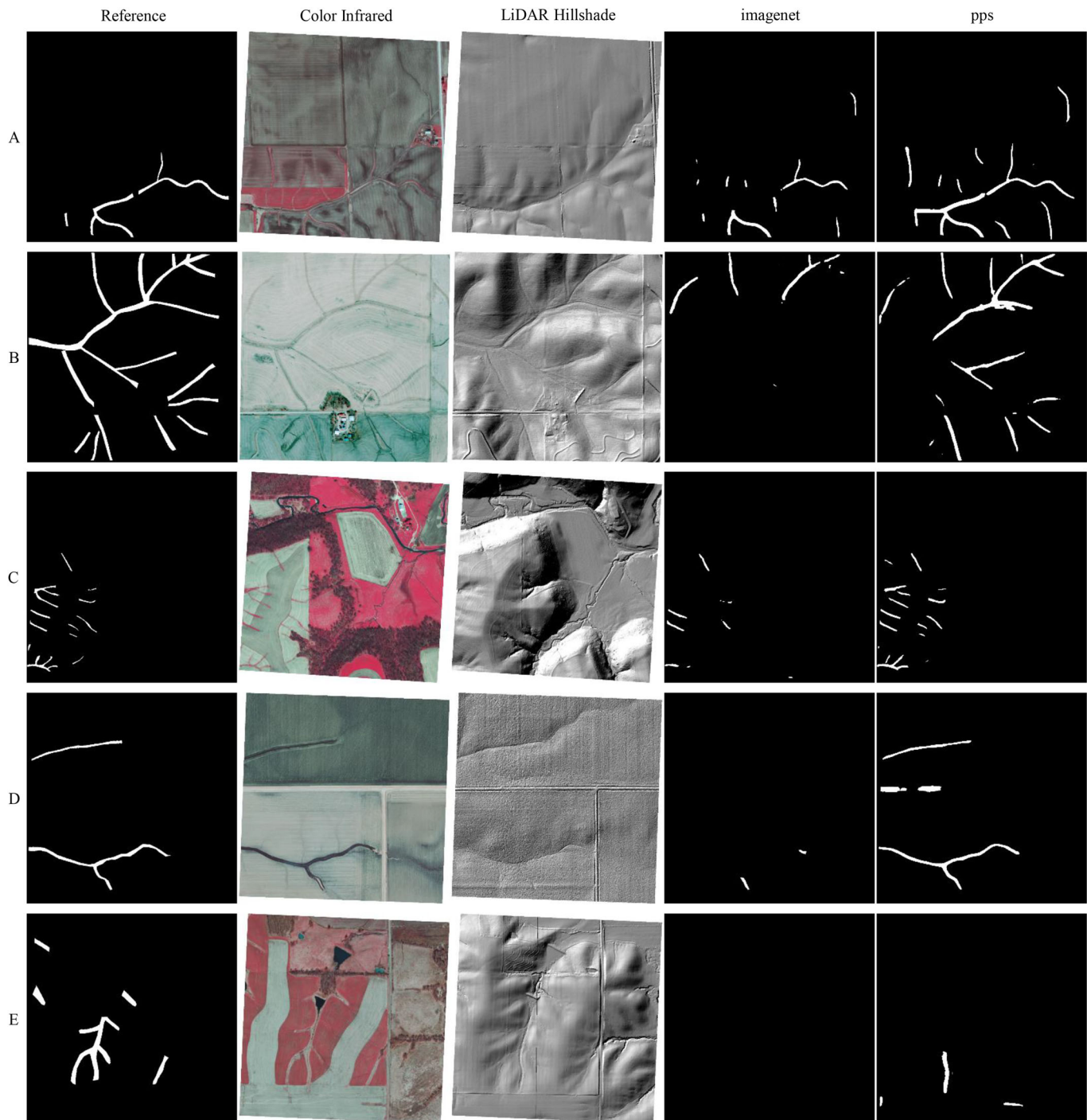


Figure 7. Grassed waterway segmentation results for the imagenet and pps (probability-proportional-to-size sample) models in comparison to the reference image derived from the Iowa BMP Mapping Project. The color-infrared and LiDAR-derived hillshade images for each example are also provided. These results compare a model trained via systematic random sampling (imagenet) to one trained via pps sampling (pps).

4.5. Topology-Preserving Centerline Dice Loss Function: *cldice*

Finally, the *cldice* model was built on the same specifications of the previous model, but we used the centerline dice loss function rather than the standard dice loss function during training. The centerline dice loss function, *cldice*, was designed to preserve the connectedness of linear features, as described in Section 3.5. For this model, we used the pps training sample of color-infrared and LiDAR-derived channels to train the U-Net model with

pretrained ImageNet weights. We hypothesized the *cldice* score would improve compared to the pps model.

5. Results

Here we evaluate the models presented in Section 4 and discuss the conclusions that may be drawn from these experiments. To evaluate these models, an independent set of 500 test units was selected for each BMP type as described in Section 3.1. We evaluated these models on the test units only after all final

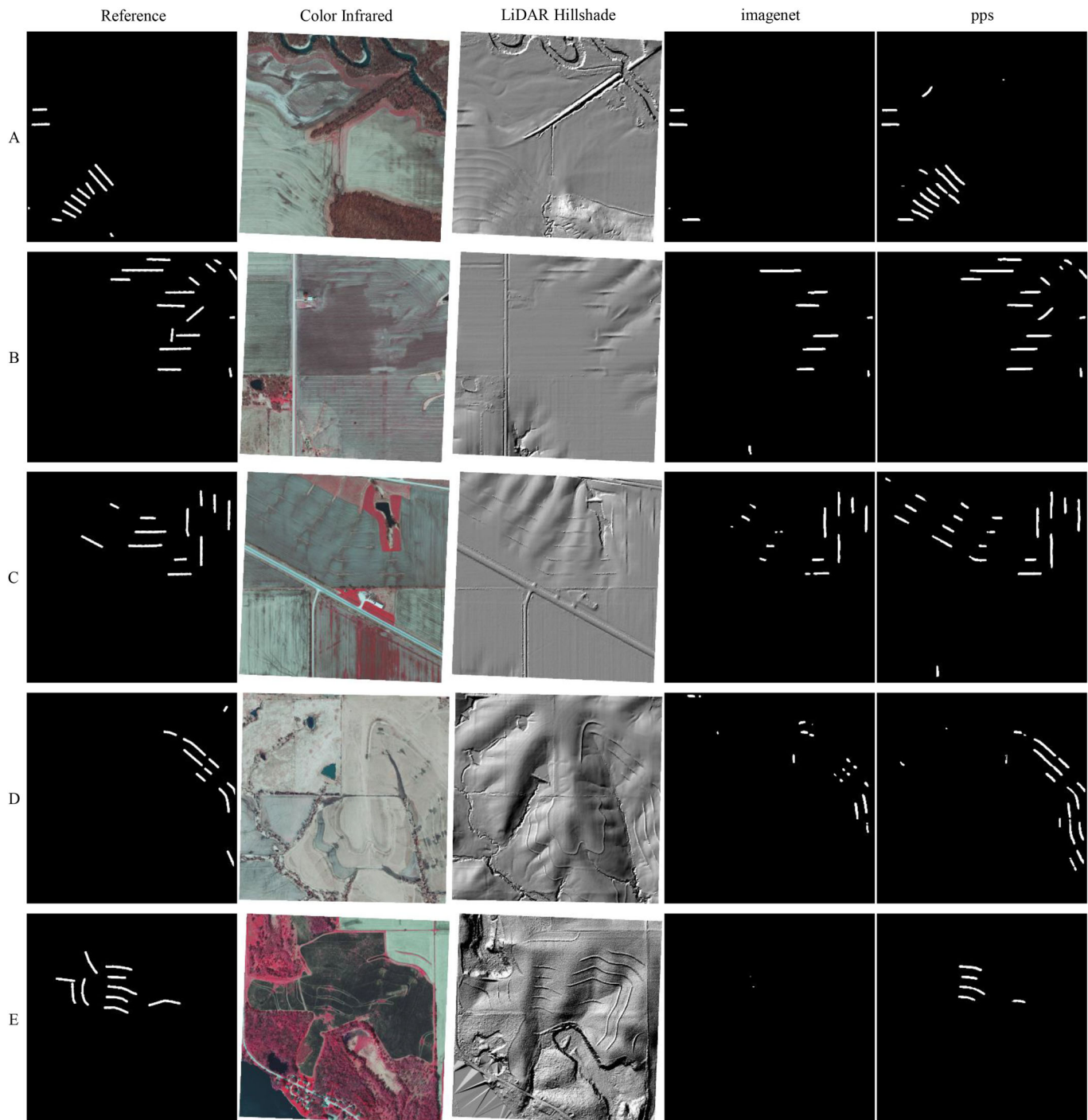


Figure 8. WASCOB segmentation results for the imagenet and pps (probability-proportional-to-size sample) models in comparison to the reference image derived from the Iowa BMP Mapping Project. The color-infrared and LiDAR-derived hillshade images for each example are also provided. These results compare a model trained via systematic random sampling (imagenet) to one trained via pps sampling (pps).

model parameters and experimental settings had been selected. In Table 2, we present the average Dice scores, cIDice scores, precision, and recall for the test datasets by BMP type and model. Refer to the supplementary materials for the standard deviations of the average Dice and cIDice scores. In Figure 5, we plot the average Dice and cIDice scores and report the significance of each sequential model comparison according to pairwise sign-tests as described in Section 4. The exact statistics and p -values for each of these tests are provided in the supplementary materials. Additionally, we report both the validation and test set

average Dice scores in the supplementary materials to demonstrate a lack of evidence of overfitting in these models.

5.1. LiDAR-Derived Hillshade Imagery Improved Performance

Given the significant increase in Dice and cIDice scores between the baseline and lidar models for all BMPs (Figure 5), we can strongly conclude that including the LiDAR-derived hillshade imagery increased segmentation performance across all BMPs.

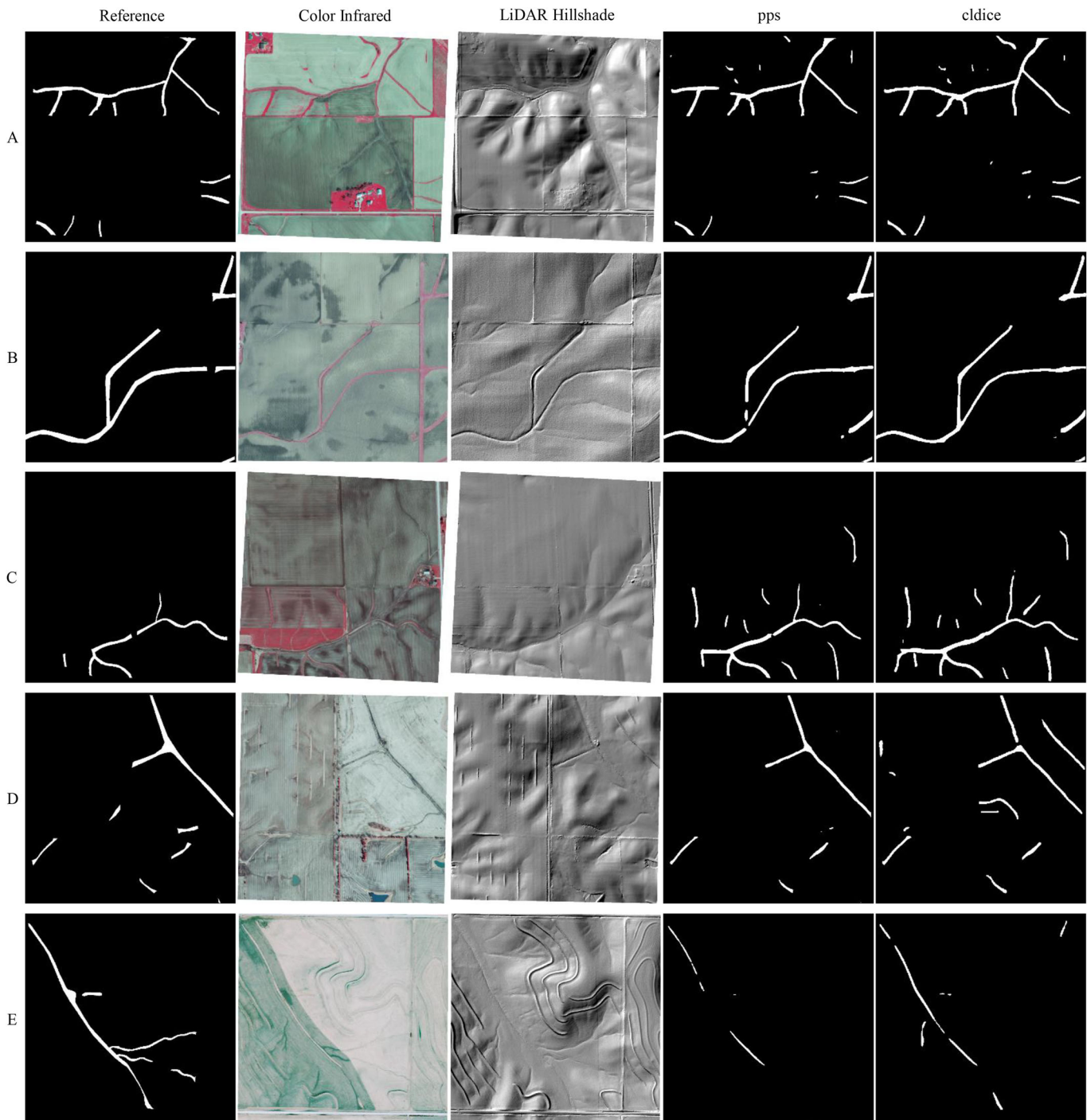


Figure 9. Grassed waterway segmentation results for the pps (probability-proportional-to-size sample) and cldice models in comparison to the reference image derived from the Iowa BMP Mapping Project. The color-infrared and LiDAR-derived hillshade images for each example are also provided. These results compare a model trained via standard Dice loss (pps) to one trained via centerline Dice loss (cldice).

The pond dams, terraces, and WASCOBs have strongly ridged structures that are often easier to identify in the hillshade image compared to the color-infrared. Model performance also increased for segmenting grassed waterways when adding the LiDAR-derived channel, but not as dramatically as the other BMP types.

Figure 6 shows the effect of incorporating the LiDAR-derived hillshade imagery by comparing segmentation results for terraces between the baseline and lidar models. In each example, the lidar model produced a segmentation that more closely

matches the reference image derived from the Iowa BMP Mapping Project. In example A, many more of the terraces were recognized after including the LiDAR-derived hillshade image in the model. In example C, several linear features the model incorrectly labeled as terraces when using the color-infrared imagery alone were corrected. The segmentation results still have room for improvement, as seen in example E, with many of the terraces remaining unidentified, but the positive effect of including the LiDAR-derived hillshade channel was strongly supported by most examples in the test data.

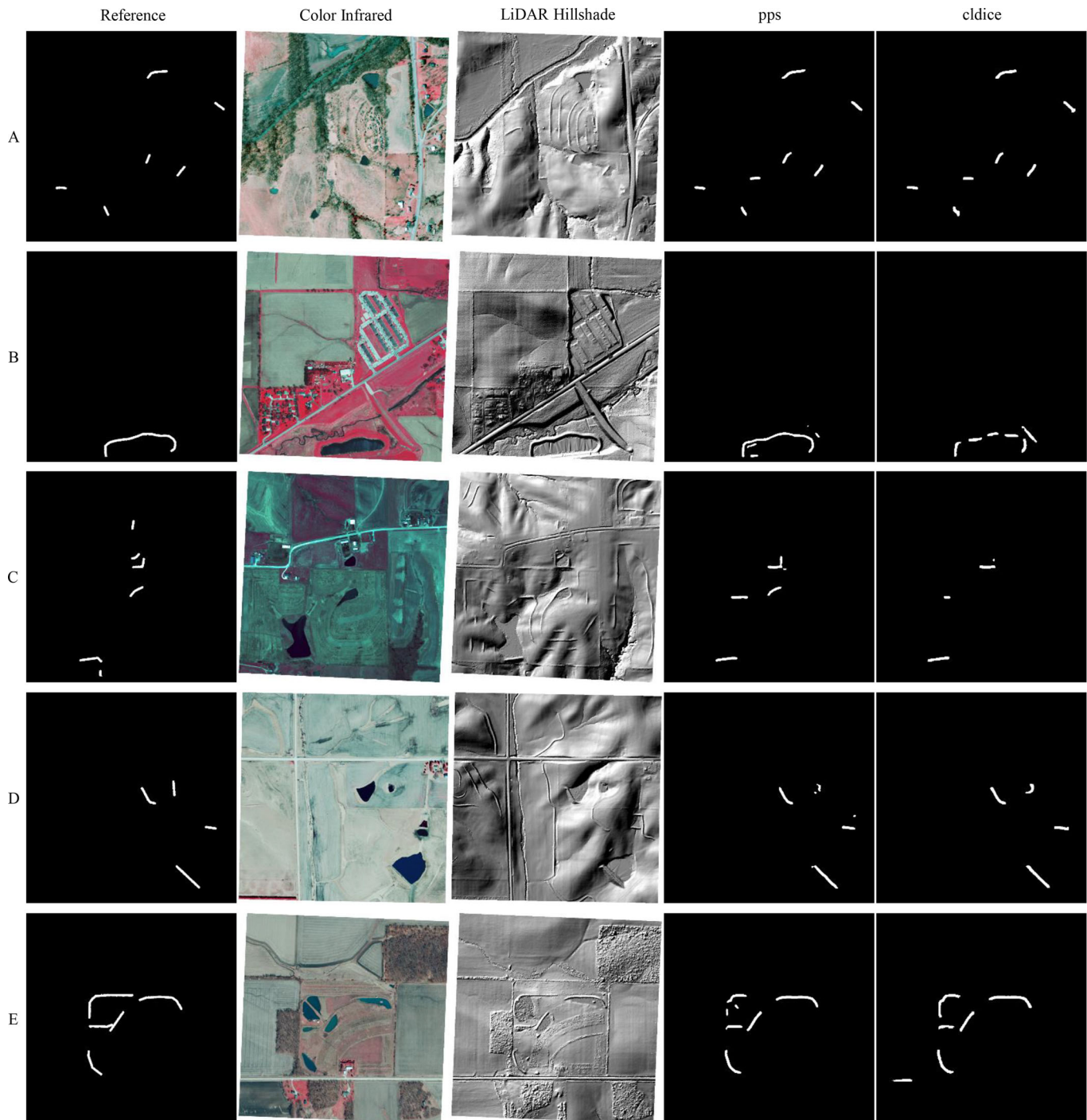


Figure 10. Pond dam segmentation results for the pps (probability-proportional-to-size sample) and cldice models in comparison to the reference image derived from the Iowa BMP Mapping Project. The color-infrared and LiDAR-derived hillshade images for each example are also provided. These results compare a model trained via standard Dice loss (pps) to one trained via centerline Dice loss (cldice).

5.2. ImageNet Transfer Learning Did Not Generally Affect Performance

The results summarized in Figure 5 suggest the inclusion of pretrained weights from ImageNet transfer learning did not largely affect segmentation performance. For the grassed waterway, pond dam, and WASCORB model comparisons, the null hypothesis of the sign-test was not rejected between the lidar and imagenet models. There was a significant improvement between the lidar and imagenet models for only the terrace BMP type. It is possible that the efficiency of model convergence increased

slightly for the imagenet models, in general. Given enough epochs, however, it seems the models performed similarly with or without the inclusion of the pretrained weights for these segmentation tasks.

5.3. Probability-Probability-to-Size Random Sampling Improved Performance

The average Dice and cLDice scores between the imagenet model (trained on a dataset selected via systematic random sampling)

and the pps model (trained on a dataset selected via probability-proportional-to-size sampling) increased significantly for all BMP types (Table 2). By magnitude, the increase in segmentation performance was especially notable between these models for the grassed waterways and WASCOBs. We hypothesized segmentation performance would improve for the pps model due to the relative increase in the amount of each BMP type included in the training images. This increase may have helped to offset the highly imbalanced nature of the data between the BMP and background classes. When we used the pps models on the test datasets, selected via systematic random sampling, we did not see evidence of overfitting. This indicates that the pps sampling method still generalized to the population while increasing performance.

Figure 7 shows examples of grassed waterway segmentation between the imagenet model and pps model. In examples A–C, the total area of grassed waterway correctly recognized by the pps model increased compared to the imagenet model. This is true as well in example D, but there is some evidence that the pps model may have more often over-predicted the amount of grassed waterway compared to models trained with systematic random sampling. This phenomenon is also evident in the relative increases of precision and recall reported for these models in Table 2. The increase in average recall was greater than the increase in average precision indicating that the pps model may have recovered more of the BMP of interest, but in doing so, it may have also included more false positive predictions.

Similarly, Figure 8 gives examples of WASCOB segmentation between the imagenet and pps models. The increase in segmentation performance was especially notable for diagonally aligned WASCOBs among these examples. It is likely that WASCOBs in a variety of orientations were included at a higher frequency in the pps training dataset, increasing the average segmentation performance. The increase in performance was notable in all examples A–E, but the increase in false positive predictions should also be noted. In example C, there were areas predicted as WASCOBs that were not included in the reference image. However, we can see that these areas may be WASCOBs that were simply not included in the Iowa BMP Mapping Project database. This is a problem that often exists in machine learning tasks. There may be many incorrectly labeled features in the reference data, which complicates both the training and assessment of machine learning models.

5.4. Centerline Dice Loss Preserved the Topology of Some Features

It is difficult to come to a definitive conclusion about the performance of the centerline Dice loss (cDice) function compared to the standard Dice loss. In Figure 5, it can be seen that segmentation performance for the cDice model increased significantly compared to the pps model when measured via cDice score, but the average Dice score did not significantly increase. While these metrics are similar, the functions measure different characteristics. By using the cDice loss function, the model is trained to preserve the centerline of the learned features. To evaluate how well this topology-preserving characteristic affected general performance, we considered some example test cases.

Figures 9 and 10 show segmentation results for the cDice model compared to the pps model for grassed waterways and pond dams, respectively. The topology-preserving property was more pronounced for grassed waterway segmentation than pond dam segmentation.

In Figure 9, examples A–C show grassed waterways that had better preservation of topology in the cDice model segmentations compared to the pps model. Each of these grassed waterways had a break in its connectivity in the pps segmentation, whereas the connectivity was preserved by the cDice model. There were cases, but relatively fewer, where the opposite applied, however, as shown in example D. In general, it seems that the cDice loss function may have helped to preserve topology for grassed waterway features and other BMPs that have lengthy connected regions, such as terraces.

In Figure 10, we see that the performance of the cDice loss function for pond dams was less promising. While both the pps and cDice models gave good segmentation results in general (example A), breaks in connectivity may have happened just as often or more often when using the cDice loss function compared to using the standard Dice loss for pond dams (examples B–E). In example B, one long pond dam was broken up into many small, unconnected pieces by the cDice model. These results may have occurred because the majority of pond dams have relatively short, straight embankments. The cDice loss function may have learned these centerline characteristics without the same flexibility as the standard Dice loss function, resulting in pond dam segmentations that did not generalize as well to bent, angled, or lengthy cases.

5.5. Segmentation Performance Varied by BMP Type

From the table of average segmentation results (Table 2) and the figures that demonstrate segmentation for each BMP type, it seems that average segmentation performance was not equal among BMPs. However it is likely that identifying some of these BMPs via segmentation is more difficult than others due to their variable structural and vegetative features, which may also affect the accuracy of the reference data itself. For example, grassed waterways have less pronounced structural properties than other BMPs. Further inspection of the Iowa BMP Mapping Project database revealed that many ditches may have been wrongly classified as grassed waterways and there may have been many that were missed. Furthermore, many of the grassed waterways were not actually grassed when captured, making these grassed waterways harder to recognize using color-infrared imagery from a single time point.

It seems that there may also be many errors in the WASCOB dataset. Our best-performing models often recognized features that may truly be WASCOBs but were not labeled in the Iowa BMP Mapping Project. The similarity between terraces and WASCOBs also complicated segmentation. Overall, however, the segmentation results for the pond dams and terraces in this project were especially promising. We have shown that the use of LiDAR-derived hillshade projects may be used to improve segmentation, especially for the predominantly structural BMPs. Our further experiments demonstrated methods to further increase segmentation performance by using the

probability-proportional-to-size sampling method and cDice loss function (at least according to cDice score) across the BMP types generally.

6. Conclusion

In this article, we have presented a series of experiments that demonstrate progress toward an automated method of BMP mapping via deep learning. Given the results for several BMP types, including grassed waterways, pond dams, terraces, and WASCOBs, we have shown that LiDAR-derived hillshade products are an important source of imagery for BMP segmentation. We have also demonstrated that using a probability-proportional-to-size sampling method improved segmentation performance among these highly imbalanced classes. This method of sampling the training datasets improved model performance while remaining generalizable to the population. Finally, we compared centerline Dice loss performance to standard Dice loss and we found the centerline dice loss helped to preserve the connectedness of linear features, particularly grassed waterways. This article expands previous research using deep learning to identify soil erosion and water conservation BMPs in remote sensing imagery and it may further help to generate a framework for automated monitoring of the use of BMPs across the Midwestern United States. Understanding the trends of the use of BMPs over time and identifying locations to target future installations of BMPs could advance conservation efforts in accordance with the 2008 Gulf Hypoxia Action Plan.

Supplementary Materials

The Supplementary Materials include four additional tables of results: test set results with standard deviations, pairwise sign-test results for each model comparison using the Dice score and the cDice score (separately), and a comparison of Dice scores on the validation and test sets.

Disclosure Statement

The authors report there are no competing interests to declare.

Data Availability Statement

Code to reproduce the models described in this article may be found on GitHub at https://github.com/labuzzetta/bmp_unet.

Funding

The authors gratefully acknowledge the Center for Survey Statistics and Methodology at Iowa State University for supporting this research.

ORCID

Charles J. Labuzzetta  <http://orcid.org/0000-0002-6027-0120>
Zhengyuan Zhu  <http://orcid.org/0000-0002-2266-0646>

References

Abdelhack, M. (2020), "An Open-Source Tool for Hyperspectral Image Augmentation in Tensorflow," arXiv abs/2003.13502, pp. 1–4. [6]

- Arbuckle, J. G. (2013), "Farmer Attitudes Toward Proactive Targeting of Agricultural Conservation Programs," *Society & Natural Resources*, 26, 625–641. [1]
- Belgiu, M., and Dragut, L. (2016), "Random Forest in Remote Sensing: A Review of Applications and Future Directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31. [2]
- Cheng, G., and Han, J. (2016), "A Survey on Object Detection in Optical Remote Sensing Images," *ISPRS Journal of Photogrammetry and Remote Sensing*, 117, 11–28. [2]
- Chollet, F. (2015), "keras," available at <https://github.com/fchollet/keras>. [6,9]
- Conover, W. (1999), *Practical Nonparametric Statistics*, Wiley Series in Probability and Statistics (3rd ed.), New York, NY: Wiley. [8]
- de Albuquerque, A. O., de Carvalho Júnior, O. A., Carvalho, O. L. F. d., de Bem, P. P., Ferreira, P. H. G., de Moura, R. d. S., Silva, C. R., Trancoso Gomes, R. A., and Fontes Guimarães, R. (2020), "Deep Semantic Segmentation of Center Pivot Irrigation Systems from Remotely Sensed Data," *Remote Sensing*, 12, 2159. [2]
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009), "Imagenet: A Large-Scale Hierarchical Image Database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. [2,9]
- EPA. (2008), *Gulf Hypoxia Action Plan 2008: For Reducing Mitigating, and Controlling Hypoxia in the Northern Gulf of Mexico and Improving Water Quality in the Mississippi River basin*, U.S. Environmental Protection Agency (EPA), Office of Wetlands, Oceans, and Watersheds [Washington, D.C.]. [1]
- Fuller, W. A. (2009), *Use of Auxiliary Information in Estimation*, pp. 95–180, Hoboken: Wiley. [3,5]
- He, K., Zhang, X., Ren, S., and Sun, J. (2016), "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. [5]
- Iglovikov, V., and Shvets, A. (2018), "Ternausnet: U-Net with VGG11 Encoder Pre-trained on Imagenet for Image Segmentation," CoRR abs/1801.05746. [9]
- Ioffe, S., and Szegedy, C. (2015), "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 of Proceedings of Machine Learning Research, eds. F. Bach and D. Blei, pp. 448–456, PMLR, Lille, France. [5]
- Isikdogan, F., Bovik, A. C., and Passalacqua, P. (2017), "Surface Water Mapping by Deep Learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10, 4909–4918. [2]
- ISU. (2012), *Iowa Nutrient Reduction Strategy: A Science and Technology-based Framework to Assess and Reduce Nutrients to Iowa Waters and the Gulf of Mexico*, Ames, IA: Iowa State University (ISU). [1]
- ISUGIS-SRF. (2018), "Iowa Geographic Map Server," Iowa State University Geographic Information Systems Support and Research Facility (ISUGIS-SRF), Ames, IA: Iowa State University. [3]
- Ji, S., Wei, S., and Lu, M. (2019), "Fully Convolutional Networks for Multisource Building Extraction from an Open Aerial and Satellite Imagery Data Set," *IEEE Transactions on Geoscience and Remote Sensing*, 57, 574–586. [2]
- Kassambara, A. (2019), "rstatix: Pipe-Friendly Framework for Basic Statistical Tests." [8]
- Keep, T., and McLoud, P. (2012), *Terraces*, United States Department of Agriculture Natural Resources Conservation Service, chapter 8. [2]
- Kingma, D. P., and Ba, J. (2017), "Adam: A Method for Stochastic Optimization," ArXiv. [6]
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012), "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems* (Vol. 25), eds. F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Curran Associates, Inc. [2,6]
- Li, R., Liu, W., Yang, L., Sun, S., Hu, W., Zhang, F., and Li, W. (2018), "Deepunet: A Deep Fully Convolutional Network for Pixel-Level Sea-Land Segmentation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11, 3954–3962. [2]
- Long, J., Shelhamer, E., and Darrell, T. (2015), "Fully Convolutional Networks for Semantic Segmentation," arXiv 1411.4038. [2]

- Lowrance, R., Dabney, S., and Schultz, R. (2002), “Improving Water and Soil Quality with Conservation Buffers,” *Journal of Soil and Water Conservation*, 57, 36A–43A. [1,2,3]
- Marmanis, D., Datcu, M., Esch, T., and Stilla, U. (2016), “Deep Learning Earth Observation Classification Using Imagenet Pretrained Networks,” *IEEE Geoscience and Remote Sensing Letters*, 13, 105–109. [9]
- Martins, V. S. (2020), “Deep Learning for Land Cover Classification and Environmental Analysis Using High-Resolution Remote Sensing Data,” PhD Thesis, Iowa State University. [2]
- McNeely, R., Logan, A. A., Obrecht, J., Giglierano, J., and Wolter, C. (2017), *Iowa Best Management Practices (BMP) Mapping Project Handbook*, Ames, IA: Iowa State University (ISU). [1,2,3]
- Meyer, D., and Bracmort, K. S. (2012), *Grassed Waterways*, United States Department of Agriculture Natural Resources Conservation Service, chapter 7. [3]
- Milletari, F., Navab, N., and Ahmadi, S. (2016), “V-net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation,” in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, Los Alamitos, CA: IEEE Computer Society. [2,6,7]
- Mountrakis, G., Im, J., and Ogole, C. (2011), “Support Vector Machines in Remote Sensing: A Review,” *ISPRS Journal of Photogrammetry and Remote Sensing*, 66, 247–259. [2]
- Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014), “Learning and Transferring Mid-Level Image Representations Using Convolutional Neural Networks,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717–1724. [2,9]
- Renfro, G. M. (2012), *Ponds and Reservoirs*, United States Department of Agriculture Natural Resources Conservation Service, chapter 11. [3]
- Ronneberger, O., Fischer, P., and Brox, T. (2015), “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Vol. 9351 of LNCS, pp. 234–241, Springer. (available on arXiv:1505.04597 [cs.CV]). [2,5]
- Rundhaug, T., Geimer, G., Drake, C., Amado, A., Bradley, A., Wolter, C., and Weber, L. (2018), “Agricultural Conservation Practices in Iowa Watersheds: Comparing Actual Implementation with Practice Potential,” *Environmental Monitoring and Assessment*, 190, 695. [1]
- Schulte, L. A., Asbjornsen, H., Atwell, R. C., Hart, C. E., Helmers, M. J., Isenhardt, T. M., Kolka, R. K., Liebman, M., Neal, J., O’Neal, M. E., Secchi, S., Schultz, R. C., Thompson, J. R., Tomer, M. D., and Tyndall, J. C. (2008), “A Targeted Conservation Approach for Improving Environmental Quality: Multiple Benefits and Expanded Opportunities,” in *Agriculture and Environment Extension Publications*, (Vol. 84). [1]
- Shit, S., Paetzold, J. C., Sekuboyina, A., Zhylyka, A., Ezhov, I., Unger, A., Pluim, J. P. W., Tetteh, G., and Menze, B. H. (2020), “cIDice - A Topology-Preserving Loss Function for Tubular Structure Segmentation,” CoRR abs/2003.07311. [2,6,7]
- Stoian, A., Poulain, V., Inglada, J., Poughon, V., and Derksen, D. (2019), “Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems,” *Remote Sensing*, 11, 1986. [2]
- USDA. (2012), *Part 650 Engineering Field Handbook (Amend. 48)*, United States Department of Agriculture (USDA) Natural Resources Conservation Service. [1]
- Wu, M., Zhang, C., Liu, J., Zhou, L., and Li, X. (2019), “Towards Accurate High Resolution Satellite Image Semantic Segmentation,” *IEEE Access*, 7, 55609–55619. [2,6]
- Yakubovskiy, P. (2019), “Segmentation Models,” available at https://github.com/qubvel/segmentation_models. [5,9]
- Yang, X., Li, X., Ye, Y., Lau, R. Y. K., Zhang, X., and Huang, X. (2019), “Road Detection and Centerline Extraction via Deep Recurrent Convolutional Neural Network U-Net,” *IEEE Transactions on Geoscience and Remote Sensing*, 57, 7209–7220. [2]
- Zhang, Z., Liu, Q., and Wang, Y. (2017), “Road Extraction by Deep Residual U-Net,” CoRR abs/1711.10684. [2,5]
- Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. (2017), “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources,” *IEEE Geoscience and Remote Sensing Magazine*, 5, 8–36. [2]