# Deep Transfer Learning With Field-Based Measurements for Large Area Classification

Zachary L. Langford
*Oak Ridge National Laboratory*
Oak Ridge, TN USA
langfordzl@ornl.gov

Jitendra Kumar
*Oak Ridge National Laboratory*
Oak Ridge, TN USA
jkumar@climatemodeling.org

Forrest M. Hoffman
*Oak Ridge National Laboratory*
Oak Ridge, TN USA
forrest@climatemodeling.org

***Abstract*—Classification of Electro-Optical (EO) datasets using Deep Neural Networks (DNNs) has lead to high-performance supervised learning algorithms. However, DNNs require a large amount of labeled data for training which often have limited availability in ecological studies. Up-to-date classification of vegetation in sensitive Arctic ecosystems continue to be a challenge. In an ecosystem undergoing rapid change, capturing the dynamics of vegetation requires the existing maps of vegetation (such as Alaska Existing Vegetation Type (AKEVT), circa 2000) to be updated based on frequent field based observation of vegetation. A method is needed to transfer the knowledge gained from field-collected observations to a larger area using DNNs. Transfer learning is a machine learning technique where a model trained on one task is re-purposed on another similar task. This paper seeks to train DNNs using field-collected observations and apply transfer learning to apply the knowledge gained to update existing vegetation map at larger scale. We test two DNN methods, (1) a deep multilayer perceptron (MLP) model and (2) siamese MLP network that uses a structure to rank similarity between inputs and can be used for training datasets with few samples and show good performance with limited datasets (e.g. few-shot learning). The results show $\sim$90% accuracy (using the field observations for evaluation) when transfer learning is applied to a siamese network, compared to $\sim$45% accurate when a MLP is trained on the AKEVT and evaluated on the field observations. The approach show promise for improving and update the existing vegetation maps over large areas using limited field-based observations.**

***Index Terms*—transfer learning, deep learning, vegetation classification, Arctic**

## I. Introduction

Land cover information plays an important role in understanding the impact of climate change on Arctic. Understanding of spatio-temporal dynamics of Arctic vegetation is important to understand and predict warming climate on Arctic ecosystem [1]. With vast spatial coverage and frequent temporal revisitation, remote sensing allows assessment of vegetation across the Arctic at a variety of spatial and temporal scales. Additionally, remote sensing can also help understand patterns of landscape geomorphology and parameterize land surface models [2], [3]. However, most large-scale vegetation maps for the Arctic are coarse resolution (e.g. 30 m), and not frequently updated to reflect the Arctic vegetation dynamics. Methods to incorporate new field measurements of vegetation to continually update large–scale vegetation maps are needed.

Deep neural networks (DNNs), which learn the representative and discriminative features in a hierarchical manner from the provided data, are becoming increasingly adopted and applied for image classification in the remote sensing community [4]. However, most existing DNN algorithms have several potential issues, such as requiring large amounts of data for training [5] and tending to overfit to particular classes for imbalanced datasets [6], [7]. Recent approaches in few–shot learning, which builds accurate models based on only a few samples, show great promise especially when data is limited and noisy.

One algorithm for few-shot learning is a siamese network, that hold promise for overcoming the conventional drawbacks in classification of remote sensing datasets [8], [9]. They accept a pair of inputs to measure their similarity, in order to learn a discriminative feature embedding and a similarity measurement. Langford et al. [10] presented a siamese architecture based on convolutional networks for signal classification and found significant performance increase for noisy datasets. Siamese networks has been applied to hyperspectral remote sensing to mitigate the impact of noise in data datasets and with only a few samples per training class [11].

Transfer learning aims to extract the knowledge from one or more source tasks and applies the knowledge to a similar target task [12]. Transfer learning can be an important tool to alleviate some of the limitations of deep learning and extend its application to problem where data is insufficient or noisy [13]. Rostami et. al. [14] applied transfer learning to transfer knowledge from Electro-Optical (EO) remote sensing data based classifier to classify Synthetic Aperture Radar (SAR).

In this study, we seek to transfer knowledge from field collected vegetation datasets to improve classification at larger scale where only existing coarse resolution dataset are available for training. In summary, our main contributions are as follows:

- use field-based vegetation samples for transfer learning to a larger area; and
- compare a deep multilayer perceptron (MLP) model and deep siamese network for vegetation mapping for a large extent using limited field-based data.

## II. Study region and Datasets

### A. Study Region

The study region is located on the Seward Peninsula on the western coast of Alaska (Fig. 1). The Seward Peninsula is

2375-9259/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDMW.2019.00047

at transition zone of vegetation from boreal forest to tundra [15] witnessing stressors from warming climate, making this part of Alaska an important region for characterizing and understanding vegetation distribution. We focused our analysis at a watershed in Kougarok region of the Seward Peninsula (Figure 1), where intensive field campaigns to study hydrology, biogeochemistry, and vegetation are being conducted by the US Department of Energy's Next Generation Ecosystem Experiments (NGEE) – Arctic project.

### *B. Remote Sensing Datasets*

We used datasets derived from Satellite for Observation of Earth (SPOT-5) and Interferometric Synthetic Aperture Radar (IfSAR) to map vegetation across the Seward Peninsula. SPOT-5 satellite images used in this study were gathered and made available by the "Alaska Statewide Digital Mapping Initiative", which produced a new statewide orthomosaic that provides complete multispectral coverage of the state at 2.5 m spatial resolution. This satellite image mosaic is the first consistent, high-resolution, high-accuracy, digital orthoimagery base layer ever produced across the entire state of Alaska. Three statewide mosaics are available and were used for this study: color infrared (CIR), psuedo-natural color, and panchromatic (grayscale). Quantum Spatial and Fugro Geospatial, Inc. performed the image processing, orthorectification, and mosaicing of the datasets. The SPOT-5 orthoimage was radiometrically corrected for tone, balance, and geometry quality control along tile edges for terrain and linear features. Table I lists the remote sensing datasets used in this study. IfSAR based digital terrain model for the Seward Peninsula of Alaska was obtained from the Geographical Information Network of Alaska (GINA) (http://ifsar.gina.alaska.edu/).

### *C. Vegetation Datasets*

We acquired the 30 m Alaska Existing Vegetation Type (AKEVT; http://akevt.gi.alaska.edu/) vegetation map and clipped the dataset to our study region (Fig. 1). The AKEVT map was prepared using Landsat 7 ETM+ (30 m spatial resolution) around the year 2000 [16], however, has limited accuracy and has not been updated [17]. The AKEVT map legends are based on Alaska Vegetation Classification [18] and 19 vegetation classes exist within our study region. However, our field vegetation observations focused on three dominant vegetation classes: Mixed Shrub-Sedge Tussock Tundra-Bog (MS), Dryas/Lichen Dwarf Shrub Tundra (DL), and Alder-Willow Shrub (AW). The AKEVT dataset was subset to three classes, to match the field-based observations. Fig. 1(a,b) shows the three main vegetation classes that were focus on our study, while the vegetation classes ignored shown in gray. Three vegetation types show differences in their spectral signatures in remote sensing dataset, which we exploit for their classification (Figure 2).

Sampling of vegetation plots in the field [19] utilized methods recommended for the Arctic region by the authors of the Arctic Vegetation Archive prototype [20]. These vegetation types were crosswalked to the AKEVT map resulting in three main classes of vegetation including Mixed Shrub-Sedge Tussock Tundra-Bog, Dryas/Lichen Dwarf Shrub Tundra, and Alder-Willow Shrub. Each sample was collected in a 5×5 m grid to match the remote sensing datasets. Fig. 1 (b and c) shows the location of the field-based samples collected over the study region. Table II shows the number of samples collected and area distribution of vegetation classes in AKEVT over the study region.

### *D. Preparing Datsets for DNNs*

All remote sensing datasets were processed and put together in a three dimensional ($X \times Y \times bands$) image stack for analysis. While all SPOT-5 datasets were available at 2.5 m resolution, the IfSAR DEM and AKEVT dataset were resampled to 2.5 m using the nearest neighbor interpolation to match the SPOT-5 layers. Additionally, all remote sensing datasets were normalized to between 0 and 1 for consistency before they were input to classification model. This was performed by:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{1}$$

where $x$ = ($x_1$,. . . ,$x_n$) are the DN values of the datasets and $z_i$ is the the normalized data. The images were stacked together to form a multidimensional array, where the final image consisted of a size of $8000 \times 8000 \times 8$, corresponding to $X$, $Y$, and $bands$, respectively. Only the pixels corresponding to the three vegetation classes were extracted based on AKEVT dataset (Table II). Finally, grids of 2×2 around each pixel were extracted to match the field plot sizes (5×5 m) and the dataset was converted from a 3-dimensional array to a 2-dimensional array, such that each row of this array contains the ''flattened" version of the bands respectively. Figure 3 illustrates the image extraction for the field-based model. This consisted of extracting a 2×2 grid around the center for each field sample and using this dataset to train the model. For the labeled dataset, the field sample was converted to a pixel corresponding to the class label (Figure 3(c)).

## III. Modeling Approach

Traditional machine learning techniques try to learn each task from scratch, while transfer learning techniques try to transfer the knowledge learned from a previous task to another similar task [12]. Fig 4 shows an overview of the transfer learning approach developed in this study. First, the Multilayer Perceptron (MLP) and siamese MLP models are trained using the 30 field vegetation observations. Second, we transfer the weights to new MLP and siamese MLP models and freeze the weights. Third, we add extra layers to both models and train with the AKEVT dataset. Finally, we validate both transfer learning models using the 30 field samples. We also build models without transfer learning for comparison, validating on the 30 field vegetation observations. Section III-A discusses the MLP models. Section III-B describes the development of siamese MLP model. Section III-C discusses the transfer learning approach using the field observation trained MLP and siamese MLP models.

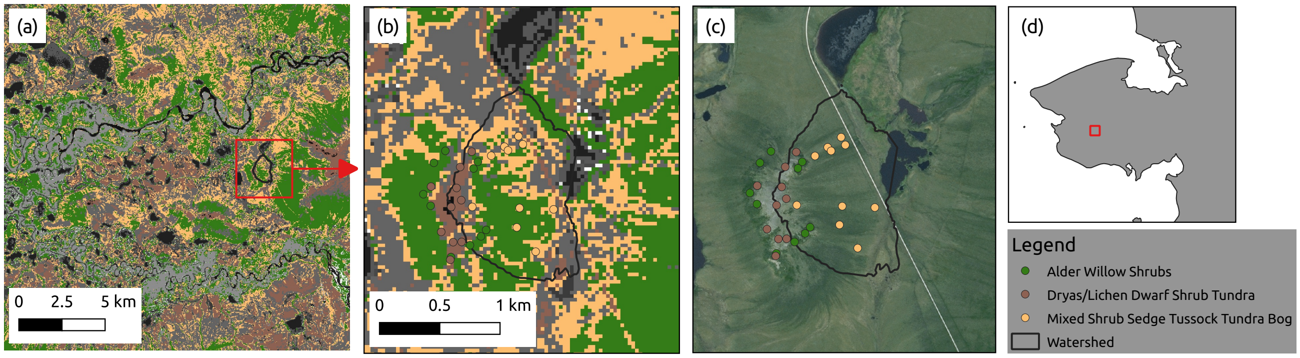



Fig. 1: Study area based on the SPOT-5 footprint, showing (a) AW, DL, MS vegetation types from Alaska Existing Vegetation (AKEVT) map for Seward peninsula with footprint of study region shown as red box; (b) AW, DL, MS types over Kougarok watershed. Shown by colored circles are the location of field vegetation plots; (c) SPOT-5 true color image over the watershed; and (d) location of study region with respect to the Seward Peninsula. AW: Alder Willow Shrubs, DL: Dryas/Lichen Dwarf Shrub Tundra, MS: Mixed Shrub Sedge Tussock Tundra Bog. Vegetation types not included in the study are shown in grey color and non-vegetated areas in black.

TABLE I: Spectral and topographic variables used in the classifications

| Sensor | Predictor Variable | Unit | Date | Bands | Resolution |
|---|---|---|---|---|---|
| SPOT-5 | Green, Red, NIR (0.5–0.9 µm) | DN | August 2010 & June 2013 | 3 | 2.5 m |
| | Blue, Green, Red (0.4–0.7 µm) | DN | August 2010 & June 2013 | 3 | 2.5 m |
| | Panchromatic (0.5–0.7 µm) | DN | August 2010 & June 2013 | 1 | 2.5 m |
| IfSAR | Elevation | m | July 2012 | 1 | 5 m |

TABLE II: Area ($km^2$) of the AKEVT vegetation classes for the study region (SR). The number of samples collected are also presented.

| AKEVT Class | SR Area | Samples |
|---|---|---|
| Alder-Willow Shrub | 19.85 | 10 |
| Mixed Shrub-Sedge Tussock Tundra-Bog | 24.49 | 10 |
| Dryas/Lichen Dwarf Shrub Tundra | 9.33 | 10 |

### *A. Multilayer Perceptron Model*

A deep neural network is an artificial neural network that consists of multiple hidden layers between the input and output layers. In this study we used multilayer perceptron (MLP) model that are trained using feed-forward error back-propagation algorithm [21]. Each hidden layer consists of many units that act in parallel, each representing a vector-to-scalar function [22]. If the dataset $D = \left\{\left(x^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}$, where $x$ is the $n$-dimensional vector and $y$ is the class label associated with the instance $x$, then a feed-forward neural network models the data as a nonlinear function of:

$$p\left(y^{(n)} = 1 \mid x^{(n)}, \theta\right) = \sigma\left(\sum_i \theta_i x_i^{(n)}\right), \tag{2}$$

where $\theta$ represents the parameters of the network (e.g., weights) and $\sigma$ represents the activation function that is used to determine the value at the output node. A MLP learns to optimize the parameters $\theta$ that result in the best functional approximation of the output [22]. This can be represented as

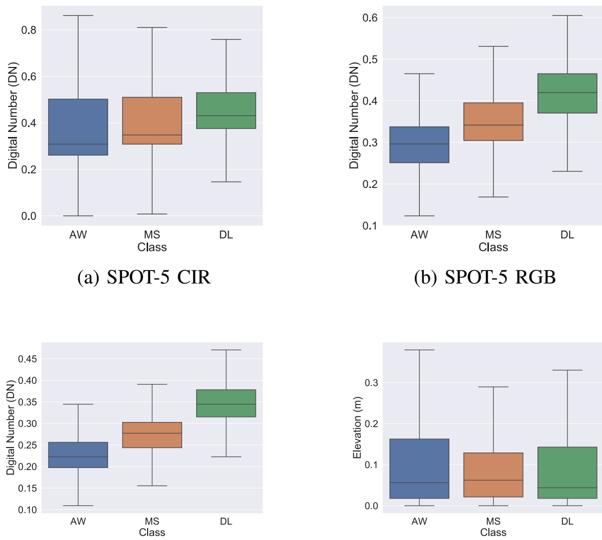
$$y^{(n)} = \sum_j \theta_j^{(2)} \sigma\left(\sum_i \theta_{ji}^{(1)} x_i^{(n)}\right) + \epsilon^{(n)}, \tag{3}$$

where $\epsilon$ represents the learning rate. Learning requires computing the gradients using the back-propagation algorithm, which calculates the direction and magnitude during training that is used to update the network weights. Training also requires making decisions such as choosing the optimizer, cost function, activation functions (which are used to compute the values at the hidden layer), and the form of the output units [22].

We implemented our approach using the TensorFlow [23] and Keras framework [24] in Python. Table III lists the parameters used for the MLP model. The model consists of three dense (hidden) layers, three dropout layers (which prevent overfitting [25]), loss function (categorical cross-entropy), and the Adam optimizer.

TABLE III: MLP model parameters.

| Parameter | Value | Description |
|---|---|---|
| Dense Layer 1 | 128 Units | ReLU Activation |
| Dropout 1 | 0.1 | Random Selection |
| Dense Layer 2 | 128 Units | ReLU Activation |
| Dropout 2 | 0.1 | Random Selection |
| Dense Layer 3 | 128 Units | ReLU Activation |
| Dropout 3 | 0.1 | Random Selection |
| Loss Function | Categorical Cross-Entropy | Measures Performance |
| Optimizer | Adam | Learning Rate (0.001) |




(a) SPOT-5 CIR (b) SPOT-5 RGB

(c) SPOT-5 Pan (d) DEM

Fig. 2: Distribution of vegetation spectral response across all remote sensing datasets used in the study show distinguishing (with some overlap) signatures.

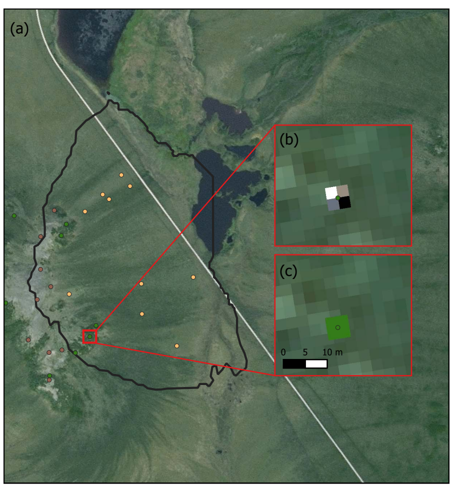



Fig. 3: Schematic of data processing over (a) watershed, (b) where 2×2 grid (i.e. 4 pixels) was extracted from remote sensing dataset, and (c) and the patch was assigned a single vegetation label.

### B. Siamese Network

The siamese network in this study is composed of twin MLP networks that share parameters and weights. Figure 5 illustrates the proposed siamese MLP. The parameters of both linked MLPs are jointly updated through backpropagation by a loss function, which computes a particular metric between the feature representations of each MLP model. Sharing weights causes similar input images to be mapped to similar positions in feature space [11].

*a) Contrastive Loss Function:* A common siamese network learns by minimizing the contrastive loss, which is defined as the distance between the outputs from the two identical inner neural networks [8]. Let $x_1$ and $x_2$ be a pair of image inputs to the MLP and $y$ be a binary label of the pair, where $y = 0$ if the images belong to the same pairs and $y = 1$ if belongs to different pairs. Let $W$ be the shared parameter vector that is subject to learning and let $G_W(x_1)$ and $G_W(x_2)$ be the two points in the low-dimensional space that are generated by mapping $x_1$ and $x_2$. Then the siamese network can be regarded as a scalar metric function $D_W(x_1, x_2)$ to measure the compatibility between $x_1$ and $x_2$, and the Euclidean distance, $D_W$. Designed to minimize $L$ with respect to $W$, it result in low values of $D_W$ for similar pairs and high values for dissimilar pairs [8]:

$$D_W(x_1, x_2) = \|G_W(x_1) - G_W(x_2)\|_2 \tag{4}$$

To simplify notation $D_W(x_1, x_2)$ will be represented as $D_W$. Then the contrastive loss is implemented as:

$$\mathcal{L}\sum_{i=1}^{p} L(W, (x_1, x_2)^i) \tag{5}$$

$$L(W, (y, x_1, x_2)^i) = (1 - y) + L_s(D_W^i) + yL_D(D_W^i) \tag{6}$$

where $(y, x_1, x_2^i)$ is the $i$-th labeled sample pair, $L_S$ is the partial loss function for a pair of similar points, $L_D$ is the partial loss function for a pair of dissimilar points, and $P$ is the number of training pairs. $L_S$ and $L_D$ must be designed such that minimizing $L$ with respect to $W$ would result in low values of $D_W$ for similar pairs and high values for dissimilar pairs [8]. The siamese model architecture is similar to the MLP model (Table III), with the only change being the loss function.

### C. Transfer Learning Models

Fig. 6 shows the overall approach for transfer learning using the field-based models to another network trained on the AKEVT dataset. Model A represents the MLP/siamese MLP model trained on the field dataset and Model B represents the MLP/siamese model trained on the AKEVT dataset (Figure 6). First, Model A is trained using the 30 field vegetation observations and the weights are saved. Second, the weights corresponding to the three dense layers (Table III) are transferred and frozen, i.e. they do not change when training Model B. Finally, additional layers are added (Table IV) to Model B and it is trained using the AKEVT dataset. The approach helps incorporate the field vegetation observations into training the classifier for the larger area, thus improving and updating and updating the information contained in the AKEVT dataset.

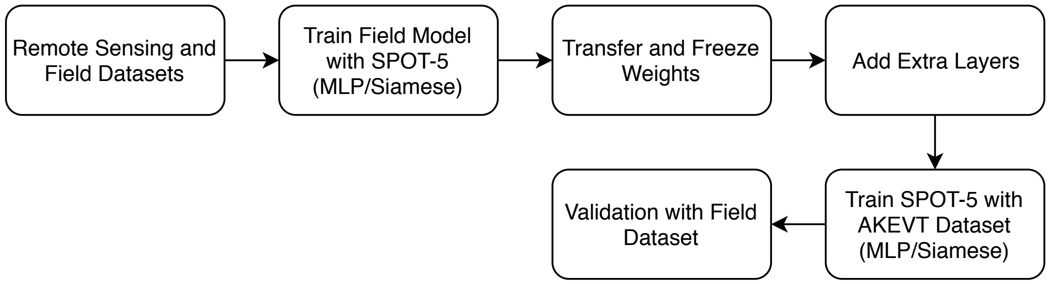



Fig. 4: Overview of developed transfer learning approach.

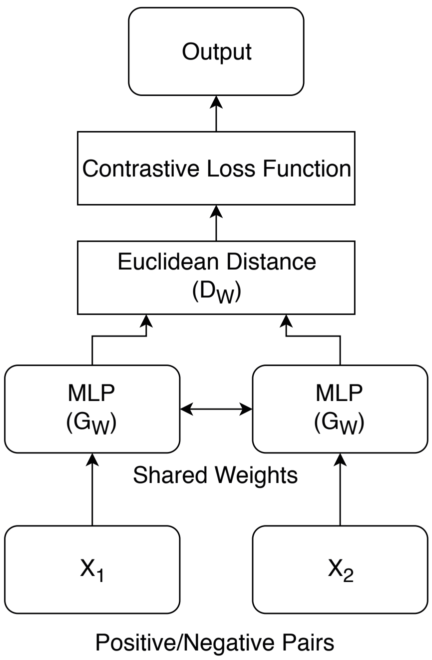



Fig. 5: Siamese MLP architecture.

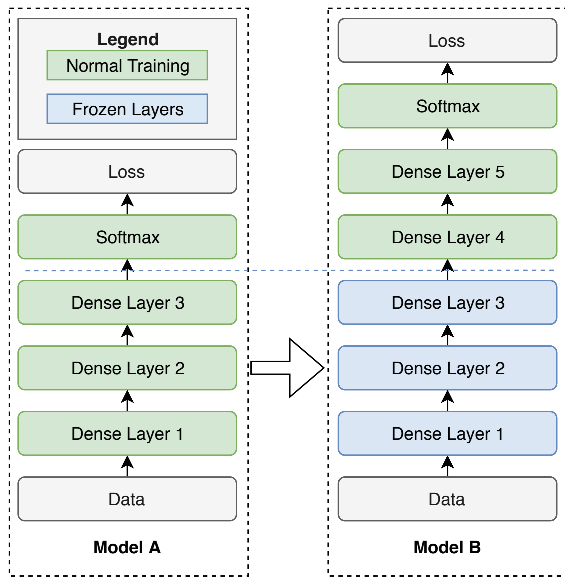



Fig. 6: Transfer learning approach, where model A represents a normal training process using the field observations. Model A weights are transferred to model B and frozen (blue boxes) with additional layers added on when training for the larger study region (Fig. 1).

TABLE IV: Additional layers when performing transfer learning

| Parameter | Value | Description |
|---|---|---|
| Dense Layer 4 | 1024 Units | ReLU Activation |
| Dropout 4 | 0.5 | Random Selection |
| Dense Layer 5 | 1024 Units | ReLU Activation |

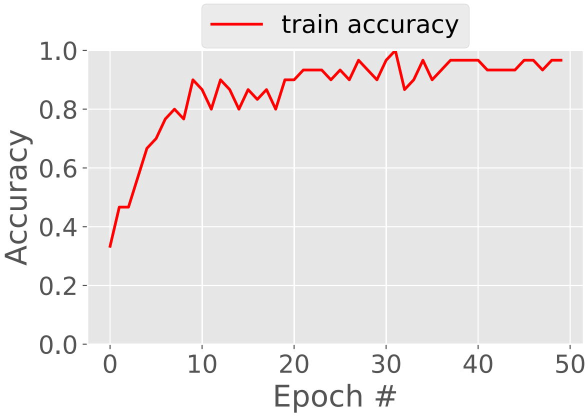
## IV. Results

### A. Field-Based Models

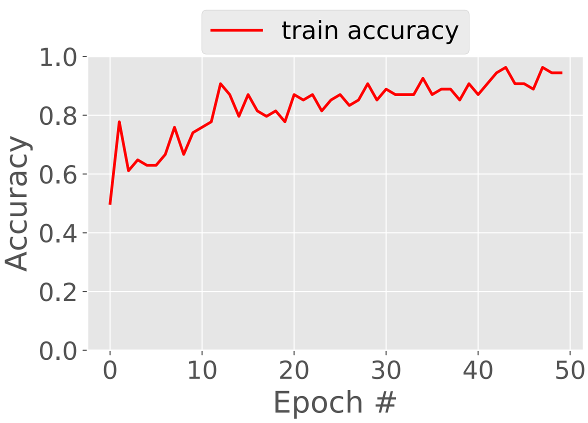
Figure 7 shows the results for the MLP (Figure 7 (a)) and siamese MLP model (Figure 7 (b)) when training with the 30 field vegetation observations. Both models reach 97 % training accuracy within 50 epochs. Additionally, the weights are only saved for the highest scoring model. A test set with the field observations were not selected in order to give the model the most samples during training.

### B. AKEVT-Based Models

Figure 8 shows the results when training with the AKEVT dataset and validating on the field observations. Figure 8 (a) shows the MLP model, where the validation accuracy doesn't reach above 47 % accurate and the training accuracy stays at consistent 76 % after a few epochs. Figure 8 (b) shows the siamese MLP model, that achieves better results for the field observations compared to the MLP model. The siamese MLP model achieves 65% accuracy for the field observations after 6 epochs, and the training accuracy is similar to the MLP (75 % accurate). Both models seem to lose performance when validating on the field samples after a few epochs.

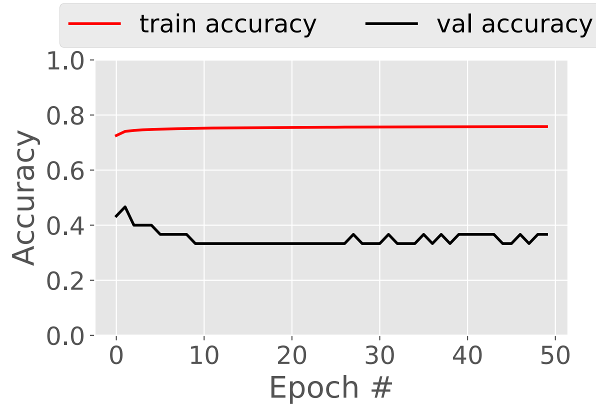



(a) MLP model accuracy over 50 epochs.

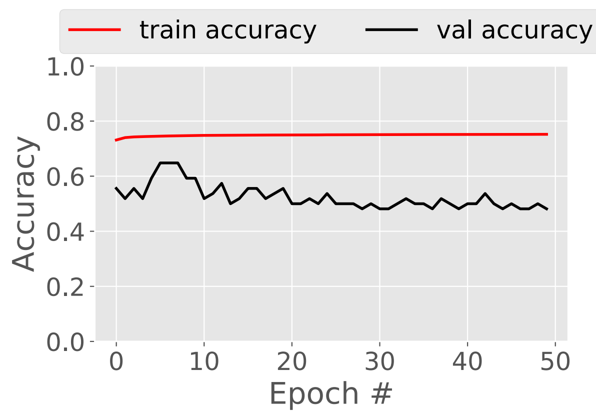



(b) Siamese MLP model accuracy.

Fig. 7: Training accuracy for the 30 field observations for (a) MLP model and (b) siamese MLP model.




(a) MLP model accuracy over 50 epochs.




(b) Siamese MLP model accuracy.

Fig. 8: Training accuracy with the AKEVT dataset for (a) MLP model and (b) siamese MLP model The red line indicates training accuracy and the black line indicates the validation accuracy.

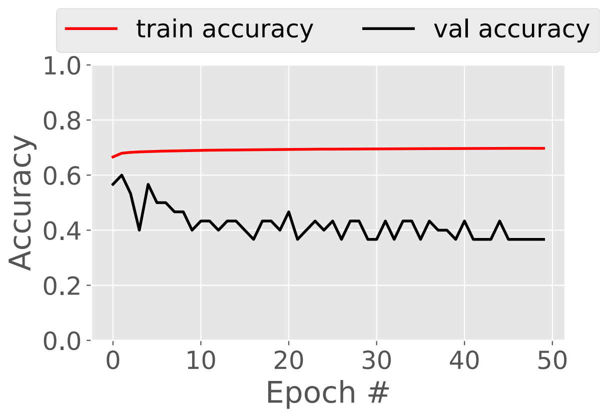
### *C. Transfer Learning Models*

Figure 9 shows the results when applying transfer learning for training with the AKEVT dataset and validating on the field observations. Figure 9 (a) shows the MLP when transfer learning is applied. The results are better compared to when transfer learning is not applied (Fig. 8 (a)), with validation scores reaching 60% accurate. However, this is achieved after the second epoch with performance decreasing for each model run. Similar to original MLP model, the model seem to learn the noise within the AKEVT dataset and performance decreases as the model is trained. Training accuracy is also lower when applying transfer learning, with scores achieving 70% accuracy.

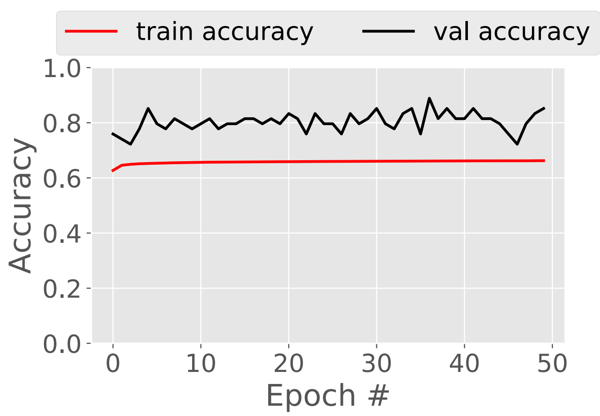
Fig. 9 (b) shows the siamese MLP model with transfer learning. The validation accuracy is always above the training accuracy, with accuracy scores reaching 89 % accurate when using the field observations for validation. Training accuracy reaches 66 % after a few epochs and is consistent during 50 epochs. This shows ideal performance when validating against the field observations and acceptable performance for AKEVT.

### *D. Summary of Model Performance*

Table V shows a summary of results for this study. Overall, the siamese networks show better performance over the MLP model. Applying transfer learning using the field observations-based model weights shows better performance when trained on the AKEVT dataset, with both methods achieving better validation accuracy compared to training solely on the AKEVT dataset. The transfer learning siamese network validation accuracy is much higher than training accuracy, compared to the MLP model.




(a) MLP model accuracy over 50 epochs.




(b) Siamese MLP model accuracy.

Fig. 9: Training accuracy with the AKEVT dataset for transfer learning (a) MLP model and (b) transfer learning siamese MLP model. The red line indicates training accuracy and the black line indicates the validation accuracy.

TABLE V: Summary of results for regular training and transfer learning (TL)

| Model | Train Accuracy | Validation Accuracy |
|---|---|---|
| Field MLP | 100% | N/A |
| Field Siamese | 96% | N/A |
| AKEVT MLP | 76% | 47% |
| AKEVT Siamese MLP | 75% | 65% |
| TL AKEVT MLP | 70% | 60% |
| TL AKEVT Siamese MLP | 66% | **89%** |

## V. Conclusions and Future Work

This study used deep learning on field-based datasets and transferred this knowledge by freezing the weights for other MLP and siamese MLP models applied to larger areas. This is a first approach using deep transfer learning and other sophisticated methods should be tested, such as siamese convolutional neural networks (CNNs) [11]. Additionally, hyperparameter optimization is needed to evaluate the current approach and see how to improve the study. For example, one could optimize the increase of field datasets and associated impacts to model structure.

The siamese architecture shows better performance when validated on the field observations for both regular training and transfer learning, with transfer learning having a much higher validation accuracy than training accuracy. This demonstrated the potential of few-shot learning methods. Some recent approaches in few-shot learning are prototypical networks, which uses the support set to extract a prototype vector from each class and classifies the inputs in the query set based on their distance to the prototype of each class [26]. This method shows state-of-the performance for few-shot learning and an approach using transfer learning could yield better results. Additionally, meta-learning approach could be utilized to adjust the model weights during training that is representative of the clean unbiased dataset (i.e. field samples). For example, Ren et al. [27] applied meta gradient descent step on the current mini-batch example weights to minimize the loss on a clean unbiased validation set.

One issue is the overfitting of models to the AKEVT dataset when transfer learning is not applied. Both models when trained on the AKEVT dataset (without transfer learning) seem to lose performance when validating on the field observations after a few epochs. This is mainly caused by the AKEVT dataset, meaning the model is learning the noise and is not representative of the field samples. Additional methods that could be done by increasing the batch size or lowering the learning rate [28]. Another approach could be batch normalization, a technique that aims to improve the training of neural networks by stabilizing the distributions of layer inputs [29]. These approaches combined with transfer learning could yield better results for all methods in this study.

Another issue in applying this type of method is imbalance classes [7]. The original AKEVT contained 19 classes and we condensed this down to 3 classes to match the field-based dataset, which had a similar distribution. One approach to solve this problem would be to incorporate all classes to increase the field dataset size and apply methods mentioned above related to few-shot learning [26] and meta-learning [27].

We show that the approach of using transfer learning with a siamese network shows great potential for incorporating expert knowledge into another dataset that contains problems, such as being coarse, noise and possibly out of date. Future directions of this work include preventing model overfitting, test out new few-shot learning approaches with transfer learning, and handling class imbalance.

## References


[1] R. J. Norby, V. L. Sloan, C. M. Iversen, and J. Childs, "Controls on fine-scale spatial and temporal variability of plant-available inorganic nitrogen in a polygonal tundra landscape," *Ecosystems*, vol. 22, no. 3, pp. 528–543, Apr 2019. [Online]. Available: https://doi.org/10.1007/s10021-018-0285-6

[2] Z. Langford, J. Kumar, F. M. Hoffman, R. J. Norby, S. D. Wullschleger, V. L. Sloan, and C. M. Iversen, "Mapping Arctic plant functional type distributions in the Barrow Environmental Observatory using WorldView-2 and LiDAR datasets," *Remote Sensing*, vol. 8, no. 9, p. 733, Sep. 2016.
[3] J. Kumar, N. Collier, G. Bisht, R. T. Mills, P. E. Thornton, C. M. Iversen, and V. Romanovsky, "Modeling the spatiotemporal variability in subsurface thermal regimes across a low-relief polygonal tundra landscape," *The Cryosphere*, vol. 10, no. 5, pp. 2241–2274, 2016. [Online]. Available: http://www.the-cryosphere.net/10/2241/2016/
[4] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, June 2016.
[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.
[6] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249 – 259, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608018302107
[7] Z. L. Langford, J. Kumar, and F. M. Hoffman, "Wildfire mapping in Interior Alaska using deep neural networks on imbalanced datasets," in *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW 2018)*, Institute of Electrical and Electronics Engineers (IEEE). Conference Publishing Services (CPS), Nov. 2018.
[8] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 539–546 vol. 1.
[9] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *International Conference on Machine Learning*, 2015.
[10] Z. Langford, L. Eisenbeiser, and M. Vondal, "Robust signal classification using siamese networks," in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, ser. WiseML 2019. New York, NY, USA: ACM, 2019, pp. 1–5. [Online]. Available: http://doi.acm.org/10.1145/3324921.3328781
[11] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu, and Y. Zheng, "Siamese convolutional neural networks for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2019.
[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2009.191
[13] X. Yu, T. Liu, M. Gong, and D. Tao, "Learning with biased complementary labels," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 69–85.
[14] M. Rostami, S. Kolouri, E. Eaton, and K. Kim, "Deep Transfer Learning for Few-Shot SAR Image Classification," *Remote Sensing*, vol. 11, no. 11, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/11/1374
[15] C. Silapaswan, D. Verbyla, and A. McGuire, "Land Cover Change on the Seward Peninsula: The Use of Remote Sensing to Evaluate the Potential Influences of Climate Warming on Historical Vegetation Dynamics," *Canadian Journal of Remote Sensing*, vol. 27, pp. 542–554, 07 2001.
[16] M. Fleming, "Develop an Existing Vegetation Layer for the Western Alaska LCC Region," https://akevt.gi.alaska.edu/data/AKEVT_WALCC_FinalReport.pdf, 2015.
[17] Z. L. Langford, J. Kumar, F. M. Hoffman, A. L. Breen, and C. M. Iversen, "Arctic vegetation mapping using unsupervised training datasets and convolutional neural networks," vol. 11, no. 1, p. 69, Jan. 2019.
[18] L. A. Viereck, *The Alaska Vegetation Classification*, ser. General technical report. USDA Pacific Northwest Research Station, 1992. [Online]. Available: https://books.google.com/books?id=mSU2jwEACAAJ
[19] A. Breen, C. Iversen, V. Salmon, H. V. Stel, and S. Wullschleger, "NGEE Arctic Plant Traits: Plant Community Composition, Kougarok Road Mile Marker 64, Seward Peninsula, Alaska," 2016.
[20] D. A. Walker, A. L. Breen, L. A. Druckenmiller, L. W. Wirth, W. Fisher, M. K. Raynolds, J. Sibík, M. D. Walker, S. Hennekens, K. Boggs, T. Boucher, M. Buchhorn, H. Bültmann, D. J. Cooper, F. J. Daniëls, S. J. Davidson, J. J. Ebersole, S. C. Elmendorf, H. E. Epstein, W. A. Gould, R. D. Hollister, C. M. Iversen, M. T. Jorgenson, A. Kade, M. T. Lee, W. H. MacKenzie, R. K. Peet, J. L. Peirce, U. Schickhoff, V. L. Sloan, S. S. Talbot, C. E. Tweedie, S. Villarreal, P. J. Webber, and D. Zona, "The alaska arctic vegetation archive (ava-ak)," *Phytocoenologia*, vol. 46, no. 2, pp. 221–229, 09 2016. [Online]. Available: http://dx.doi.org/10.1127/phyto/2016/0128
[21] R. Collobert and S. Bengio, "Links between perceptrons, mlps and svms," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 23–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015415
[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.
[23] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
[24] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.
[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html
[26] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *CoRR*, vol. abs/1703.05175, 2017. [Online]. Available: http://arxiv.org/abs/1703.05175
[27] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *ICML*, 2018.
[28] S. L. Smith, P. Kindermans, and Q. V. Le, "Don't decay the learning rate, increase the batch size," *CoRR*, vol. abs/1711.00489, 2017. [Online]. Available: http://arxiv.org/abs/1711.00489
[29] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 2483–2493. [Online]. Available: http://papers.nips.cc/paper/7515-how-does-batch-normalization-help-optimization.pdf