1 Training details

Here, we report details and settings for machine learning which are necessary to replicate our work.

We used moderate image augmentation for both Centred and Per-pixel models: varying brightness, contrast, saturation and hue by up to 10% at a probability of 75%; scale between $\left[\frac{1}{1.05}, 1.05\right]$; rotation uniformly sampled between $[-180°, 180°]$; and horizontal flipping at a probability of 50%. All necessary pixels were extracted for each example such that there were never any missing pixels in the augmented data, even after random rotations. We did not use any augmentation for the Superpixel models.

The deep learning models were each trained using a single Titan X or GeForce 2080 GPU for 50 epochs. Training for longer resulted in overfitting, so, we used a simple multiplicative learning rate scheduler to stabilise training and diminish changes in validation R^2 between epochs. The MLP was trained for 2000 epochs because it took more training iterations to converge. One epoch for Superpixel methods was defined as seeing each plot once. For consistency, one epoch for both Centred and Per-pixel methods was defined as taking an image crop centred on each plot once. Thus the Centred and Per-pixel methods see many pixels multiple times per epoch as context.

For the Superpixel method, we performed a full grid search across aggregation methods, pixel overlap threshold and combinations of the colour bands and various vegetation indices. The selected hyperparameters performed best on all three of RF, XGB and SVM. The RF-specific hyperparameters were found through random search.

To find the hyperparameters for the Centred and Per-pixel models we used a restricted grid search. We first found a good batch size and learning rate with grid search, then trialled using pretraining or not (Centred only) and different learning rate decays (both). The hyperparameters were determined for each model independently (see Table S1). The Per-pixel output hyperparameters could be tuned without retraining the model, so we used a full grid search of pixel overlap threshold and aggregation function across all models, as for the Superpixel input.

model	**lr**	**batch**	γ	**pretrained**
MLP	$1e^{-3}$	256	0.997	N
VGG-A	$1e^{-4}$	64	0.96	Y
ResNet18	$1e^{-3}$	32	0.93	Y
ResNet50	$1e^{-3}$	8	0.96	Y
DenseNet161	$1e^{-3}$	32	0.96	Y
UNet++	$1e^{-3}$	32	0.93	N
DeepLabv3	$1e^{-4}$	32	0.96	N

Table S1: Learning hyperparameters for each model. Where γ is the multiplicative learning rate decay.