Refinement of Large-scale Vegetation Height Maps

Taylan Soydan

Abstract

The field of remote sensing concerning vegetation height mapping has increasingly become significant due to its pivotal role in environmental monitoring, ecosystem research, and practical applications. The purpose of this research is to explore the potential refinement of such vegetation height maps through a benchmark of related methods mainly focusing on the one that formulates it as a pixel-to-pixel transformation (Pix-Transform), and it's modified version such that it uses external sources of information for ranking similarity based regularization (RankSim). For comparison, we benchmark related methods: Fast Guided Filter (FGF), Deep Anisotropic Diffusion (DADA), Pix-Transform (PT), and Pix-Transform with RankSim regularization against the global coarse VHM (Source). The results indicate that Pix-Transform with Ranksim provides perceptually superior outputs than the baseline methods. Particularly the best experiment results in improving the Source SSIM from 0.15 to 0.19 and PSNR from 7.50 to 7.55. We also conduct a patch size experiment for the Pix-Transform with RankSim to see the effect of patch size which yields results that are in-different to the patch size.

1. Introduction

The presence of precise Vegetation Height Maps (VHM) is essential for environmental surveillance, ecosystem investigation, and real-world uses. Naturally finer the VHM, the better the potential decision-making and benefits. In fact, the traditional field measurements with the addition of airborne laser scanner (ALS) are able to produce such fine VHMs but not on a global scale as this would be infeasible in terms of time and resources. On the other hand, current advances in spaceborne Light Detection and Ranging (LiDAR) applications, Global Ecosystem Dynamics Investigation (GEDI), (Dubayah et al., 2020) combined with deep learning techniques (Lang et al., 2022) succeeds in obtaining a global VHM with 10-meter resolution quality based on Sentinel-2 satellite (S2) and GEDI data. One downside in this is that the GEDI data which has a 25-meter footprint does not conform with the resolution of the input data which has a resolution of 10-meter. This causes an unwanted drop in the effective output resolution. Therefore the main goal of this work is to improve the coarse VHM such that the details lost by the resolution discrepancy are restored as much as possible. As ALS data is not globally available due to mentioned limitations, this leads to seeking self-supervised approaches to do such restoration such that the problem is defined as a guided super-resolution where the input is current coarse VHM i.e Source, S2 data i.e Guide, Dynamic World Cover Type Product i.e Segmentation to obtain Fine ALS images i.e Target (Figure 1). Therefore the Target images are to be used for only the final evaluation of predictions and not for any kind of supervision throughout model training. In this research, we investigate the potential improvements on the resolution of the current globally available low resolution solution by benchmarking the related methods and implementing RankSim regularization on Pix-Transform model to see if we can improve on the existing frameworks. In the related works, we will be going over the main ideas of the models we are using in the benchmark, followed by Data section introducing the data including how we obtain and process it. We present the methods that we focus on in this work in detail in the Methods section. Experiment section describes the training experiments we perform, including the parameters. Results section goes over the evaluation and interpretation of the results we obtain in the experiments. We finally qualitatively evaluate the model predictions in the Qualitative Model Prediction Analysis section and conclude with the Conclusion.

2. Related Work

There are many frameworks for the task we are dealing with, but as our scope is to benchmark some of the simple as well as complicated models that can take advantage of our Source and Guide images in an unsupervised way, we evaluate the research related to Fast Guided Filter, Deep Anisotropic Diffusion, and Pix-Transform frameworks.

2.1. Fast Guided Filter

Fast Guided Filter (He and Sun, 2015) is an image processing technique that refines or enhances an image while preserving its edges and fine details. It efficiently accomplishes this by using a guidance image, often a low-resolution version of the original image, to guide the filtering process. By adapting the filter's weights based on the guidance image, it ensures that the filtered image maintains the same structure as the guidance image. This technique is particularly useful in applications like image denoising, deblurring, and tone mapping, where it helps reduce noise and artifacts while preserving the important structural information in the image, and it does so with improved computational efficiency compared to traditional guided filter implementations.

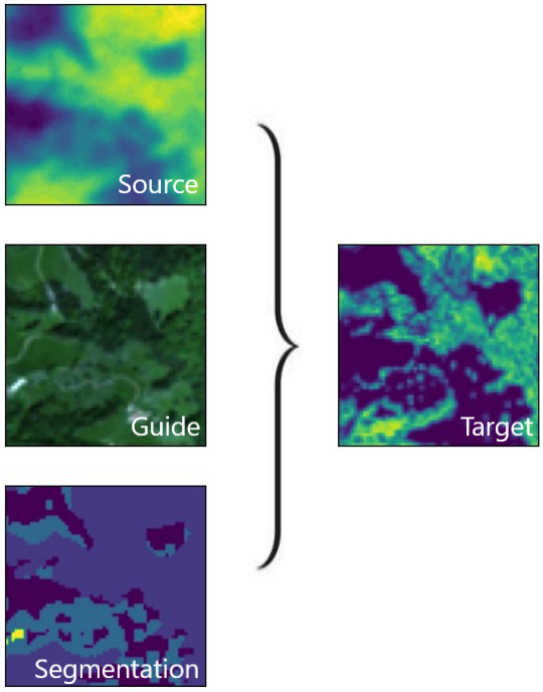

Figure 1: Sample Input Data vs. Target

2.2. Deep Anisotropic Diffusion

Guided Depth Super-Resolution by Deep Anisotropic Diffusion (Metzger et al., 2023) proposes a novel approach which combines guided anisotropic diffusion with a deep convolutional network and advances the state of the art for guided depth super-resolution. The edge transferring/enhancing properties of the diffusion are boosted by the contextual reasoning capabilities of modern networks, and a strict adjustment step guarantees perfect adherence to the source image.

2.3. Guided Super-Resolution as Pixel-to-Pixel Transformation

Guided Super-Resolution as Pixel-to-Pixel Transformation (de Lutio et al., 2019) proposes to turn the classic super-resolution upside down and see it as a pixel-to-pixel mapping of the guide image to the domain of the source image. The mapping is parametrised as a multi-layer perceptron, whose weights are learned by minimizing the inconsistency between the source image and the downsampled target image. The formulation also makes it possible to regularise only the mapping function, avoiding regularisation of the outputs, ending up with natural-looking images. The method is unsupervised, using only source and guide images to fit the mapping. Details on Pix-Transform will be covered in the methods section.

3. Data

3.1. Source

Source data is the coarse predictions obtained by fusing GEDI with Sentinel-2, via a probabilistic deep learning model. In practice the used data in the project contains 3 tiles of 10980×10980 10-meter resolution .tif files that cover the blue area in Figure 3.

3.2. Guide

Guide data is Sentinel-2 (S2) data obtained from the European Space Agency's (ESA) Sentinel-2 satellites, which are part of the Copernicus Earth observation program. These satellites are equipped with multispectral optical sensors that capture high-resolution imagery of the Earth's surface. In practice, the available data for this contains multiple bands of S2 tiles of 10-20-60 meter resolution. We resize these into 10-meter resolution in this work. S2 data originally has 12 channels, but for the project we use only the RGB, therefore 3 channels.

3.3. Segmentation

We obtain Segmentation data from the Dynamic World via the Google Earth Engine using a script. The script mainly involves a start and end date, and a country name filter variable as well as the data source. It first imports country boundaries, defines date variables, then filters the data according to the time and place and exports the desired label. In practise, GOOGLE/-DYNAMICWORLD/V1 dataset was chosen as the image collection data, Switzerland was chosen as the country to define the area of query, a start date of 2020.01.01 and an end date of 2020.05.01 was used so that it conforms with the timeline of S2 images. It is important to feed a wide time range to ensure that the query spans the desired location. For example, a date range of 1 day likely will not span the entire country provided. The query export returns a Segmentation tile with cover class labels from 0 to 10, covering the desired area which is the entire Switzerland in this case. Then using QGIS we take the intersection of the Segmentation tile with the Source tiles and keep for memory efficiency and convenience. This Dynamic World data is also 10-meter resolution and is aligned with the rest of the data tiles.

3.4. Target

Target data is the fine ALS measurements obtained by traditional measurements combined with LiDAR. In practise, same as the source, target data covers the same 3 10980×10980 tiles shown with blue in Figure 3. It also has 10-meter resolution and is intended for evaluation purposes only, meaning there is no supervised training using this target.

3.5. Processing

We merge the three tiles for each of the data sources individually and patchify to 64×64 images with no overlap and filter them according to the validity of the center pixel of the corresponding validity mask pixel that comes from a shapefile shown in Figure 3. We filter the remaining valid samples such that we remove the samples where the Target has missing values or more than 50 percent of the values of the Target sample consists of zeros. The shapefile used for validation indicates area of the fine ALS data captured on the relevant time range, so that the data sources are aligned in time.

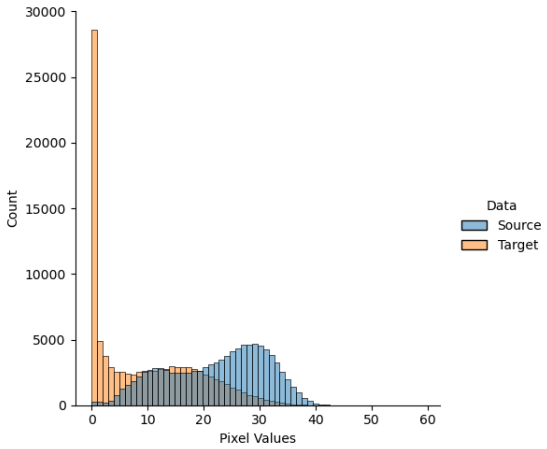

Figure 2: Pixel value histograms of Source and Target

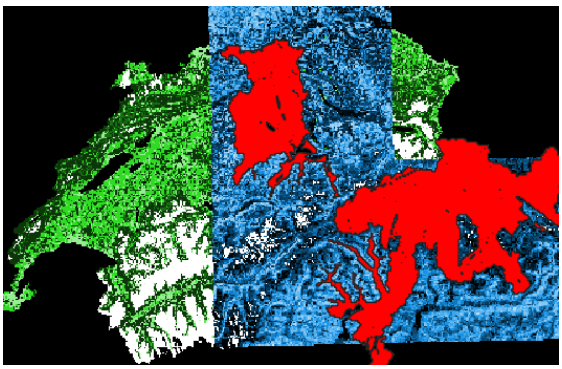

Figure 3: Alignment of the Source, Guide, and Target in blue vs. Segmentation in green/white vs. Validity Mask in red

4. Methods

4.1. RankSim Regularization

RankSim regularizer (Gong et al., 2022) encodes an inductive bias that enforces that samples closer in label space should also be closer in feature space. In contrast to distribution smoothing-based approaches, RankSim has the ability to capture not only nearby but also distant relationships. For a given data sample, RankSim encourages the sorted list of its neighbors in label space to match the sorted list of its neighbors in feature space as described in Equation 1.

$$Loss = \sum_{i=1}^{|M|} \ell\left(\mathrm{rk}(Sy[i,:]), \mathrm{rk}(Sz[i,:])\right) \tag{1}$$

Where M is the subset of input-output pairs, ℓ is the similarity function that penalizes differences in inputs which is the mean squared error in our case, $rk(a)$ is the ranking function that returns ranks of elements in vector a, Sy is the pairwise similarity matrix from label space, and Sz is the pairwise similarity matrix from feature space.

4.2. Pix-Transform

Image Sizes: Assuming we are dealing with square-shaped images, the low-resolution source image S, has dimensions $M \times M$, the high-resolution target image T has dimensions $N \times N$, and the high-resolution guide image G has dimensions $N \times N \times C$, where C represents the channels. Using 1-dimensional indices for pixels, m ranging from 1 to M^2 for low resolution and n ranging from 1 to N^2 for high resolution. When needed, these indices can be converted into 2-dimensional pixel coordinates, e.g., $[x_m, y_m] = x_m$.

Upsampling Factor: The relationship between N and M is related to the upsampling factor D. Specifically, $N = D \cdot M$. Therefore each low-resolution pixel in S covers a block of $D \times D$ high-resolution pixels in T.

Pixel Values: The value of a low-resolution pixel s_m is determined by taking the average of the corresponding underlying high-resolution pixels. So each low-resolution pixel is the result of averaging D^2 high-resolution pixels.

Formulation: Pix-Transform formulates the problem as finding a function, $f_\theta : \mathbb{R}^C \rightarrow \mathbb{R}$, parameterized by θ, that maps guide pixels, g_n, to target pixels, $\hat{t}_n = f_\theta(g_n)$.

This mapping function, f_θ, is optimized according to a source consistency loss meaning the consistency between the source image S and the estimated high-resolution image T based on the mentioned averaging and comparison. For comparison, L1 loss is used, leading to the minimization problem from Equation 1. In order to beat the ill-posed part of this problem which allows many different predictions to be consistent and therefore allows the conditions that a perfect solution can always be found via selecting a complex function, an L2 regularization term is utilized that enforces the model to stay simple that is controlled by λ. However, strictly forcing an exact one-to-one pixel-to-pixel mapping between the guide and target is also problematic and should be avoided. Otherwise, two pixels of the same guide value will always be mapped to the exact same

target depth restricting prediction power. For this, using the spatial location of the pixels is introduced, which is the x_n term. This brings flexibility to the formulation.

$$\hat{\theta} = \arg\min_{\theta} \sum_{m} \left| s_m - \frac{1}{D^2} \sum_{n \in b(m)} f_\theta(g_n, x_n) \right|^2 + \lambda \cdot ||\theta||^2 \quad (2)$$

Where $\hat{\theta}$ represents the parameters of the multilayer perceptron that minimizes the inconsistency between the low-resolution source image and the downsampled high-resolution image prediction. The authors found it useful to separate the spatial and color processing heads and add them and feed it all together to the head network such that the regularization amounts can be separately assigned.

4.3. Combining Pix-Transform with RankSim

The idea is to calculate the RankSim loss between the combined color/spatial embeddings of the model and an informative target for Ranksim that can regularize the model output according to ranking similarity. This loss term is to be added to the default source inconsistency loss. There are a few implementation challenges for this task. The first important detail is that the segmentation classes are originally arbitrarily labeled, so we rearrange them such that the cover with the lowest mean depth corresponds to label 0 and the highest to label 8. This introduces ordinality to labels necessary for RankSim to be able to rank the labels. Second, RankSim requires continuous labels, as the method samples unique values from the label space and the corresponding embeddings. Categorical labels in best case can yield the unique number of labels which is not sufficient in our case where we have only 9 classes. Third, RankSim works with the average pooled embeddings, flattened embeddings do not directly work. Therefore we average pool both the embeddings and the target patches of RankSim.

5. Experiment

5.1. Model Benchmark

Following the preprocessing steps, we obtain 10000 samples of 64×64 images. We apply the models mentioned in related works including Pix-Transform model with RankSim regularization to each sample and try various parameter values including different label spaces for RankSim. To go over these parameters: The RankSim Weight represents the assigned weight for the RankSim loss, influencing its contribution to the overall training objective. The RankSim Target is the target variable (i.e. label space) used to calculate the ranking similarity loss with respect to the feature space. Among the RankSim Targets we try are: Segmentation corresponds to the DynamicWorld product image, Source relates to the corresponding source image, Rich Segmentation is the product of the corresponding segmentation image and the max scaled source image, Model Output is the output of the model at each iteration, and Binned Target involves discretizing the target image into the number of bins equivalent to those in the corresponding Segmentation image. The Positional Encoding parameter indicates the presence of positional encoding instead of the default spatial encoding of Pix-Transform. Additionally, Batch Size determines the size of each training batch, while Downscaling signifies the downscaling factor applied to the source image.

We run these experiments to investigate the conditions that can utilize RankSim in Pix-Transform as much as possible. On all Pix-Transform trainings, we use MSE loss, Adam optimizer, a learning rate of 0.001 with 10000 iterations and an early stopping patience of 200 iterations. We use a radius of 3 for Fast Guided Filter, epsilon of 0.001, and sampling factor of 4. For DADA we use a Kappa value of 0.03, use the RGB feature extractor, and use a scaling factor of 4. We determine these common parameter values based on a grid search done on a small sub-sample of the data containing 1000 samples.

5.2. Patch Size Experiment

On a separate experiment, we patchify the original data into 256×256 patches of 211 healthy samples. Then we obtain 128×128 and 64×64 subpatches of each of these and train them with Pix-Transform with RankSim. Then we sample all of the predictions to 64×64 images of 3376 samples and remove samples where the target images contain more than half of their pixels as zeros similar to the preprocessing steps, ending up with 2927 images of patchsize 64×64. Then we run the Pix-Transform with Ranksim with the aforementioned parameters on the samples and compare model predictions, calculate errors and mean-aggregate the errors against patch size to form Table 1.

6. Results

To assess the performance of our results, we employ two widely used error metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE quantifies the average squared differences between predicted and actual values, offering a measure of overall accuracy. Meanwhile, MAE calculates the average absolute differences, providing a robust evaluation of prediction accuracy. Additionally, we utilize two perceptual metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). PSNR assesses the image quality by considering the ratio of maximum signal power to noise, while SSIM evaluates the visual similarity between predicted and actual values, incorporating factors relevant to human perception such as luminance, contrast, and structure. These metrics collectively offer a comprehensive evaluation of both numerical accuracy and perceptual quality in our results.

We show the model benchmark experiment results in Table 2 which indicate an advantage of RankSim regularized Pix-Transform over other methods in terms of SSIM and PSNR, although no advantage in terms of MSE and MAE. This indicates that although the outputs have perceptually better quality, they are not numerically closer to the Target on average.

RankSim Target parameter appears to play the biggest role in determining RankSim performance (Table 2). Experiments

with the use of Binned Target image as RankSim Target yield the best results as expected. This is due to the fact that this use of the target image provides leaked ground-truth depth information to RankSim and it's only shown to see the outcome when RankSim is directly provided the discretized depth information. As the actual task is non-supervised, this kind of performance can not be produced in practice and should be used for benchmarking purposes. We can also consider this use of the Binned Target as imagining the case where Segmentation perfectly captures the cover-depth relationship. It is also clear that using the model output as RankSim Target yields superior results. This approach enforces the model outputs to stay Ranking-Similarity-wise consistent throughout training with respect to the embeddings and results in smooth learning, producing robust predictions despite the fact that it does not use external segmentation information. Then comes the rich segmentation which performs better than both the default segmentation and source images as it provides a more continuous target to RankSim, increasing its utilization. The use of segmentation image as the target seems to be performing relatively poorly as it has discretized value limitations preventing full RankSim utilization. The informative power of Segmentation data is also questionable which will be addressed later on in the section.

A greater batchsize of 256 instead of 32 seems to be working better as it enables a less noisy training and possibly reduces overfitting. Considering the Pix-Transform's framework, the training can be prone to overfitting where a larger batchsize can get handy.

A higher downscaling factor of 8 rather than 4 also seems to be yielding a higher performance as it increases the potential of guidance through RankSim. With a high downscaling factor, the outputs are further differentiated from source images and incorporate more details from the guide and segmentation image, giving more room for RankSim to work, providing extra sharpness.

Positional Encoding does not yield good results in this case as the predictions seem to be overfitted to the guide images in many cases, and the outputs are heavily affected by the positional encoding grid lines, lowering perceptual and numeric performance. In some cases, it has the potential to incorporate details from the guide image although they are clearly nonexistent in the source (See PosEnc in the first sample in Figure 4).

Regarding the patch size experiment results (Table 1), changes in the size of patches do not seem to affect the outputs significantly, this can be explained due to the fact that the images from various patchsizes are generally, locally homogenous and therefore do not bring significant additional depth information to be exploited.

One important observation to make is to be aware of the limitations of the input data. In this task, Source images are not the exact downscaled versions of Target images. This numerical discrepancy between the Source and Target naturally imposes a limitation on the output performance. We can see from the Table 3 that MSE among the cover types in all experiment outputs is parallel to the MSE between the Source and Target indicating that the discrepancy-related errors are preserved among cover types and carried to model predictions. Another limitation regarding the Segmentation and Source images is that the depth standard deviations are quite high relative to mean depth values (Table 3) which has the potential to not inform the model in the desired way or misguide the predictions. One more limitation is that, in many cases there is a high discrepancy between the Segmentation images and Target images (Figure 4), meaning that guiding the models with these cover types can in fact be misguiding. For example, looking at Figure 4, the Segmentation zones marked by yellow pixels indicate a different cover type than its neighbors and therefore different depth values. However, they do not have a different corresponding depth value than their neighboring regions in Target images.

7. Qualitative Model Prediction Analysis

Here we go over a few of the predictions we observe in model benchmark experiment namely the examples we present in Figure 4. For convenience, we rename the significant experiments such that we name Exp3 as Leak as it uses the binned version of the Target for RankSim. We call Exp24 SelfSim as it uses the model outputs as RankSim Target instead of any external data, we call Exp6 RichSeg as it uses the enriched version of the Segmentation, call Exp10 Seg as it uses the Segmentation, and call Exp25 PosEnc as it uses positional encoding. For all samples, we can see that FGF and DADA provide a limited improvement in resolution and that Pix-Transform without Ranksim provides a slightly greater resolution than the FGF and DADA. When RankSim is used with the Binned Target as in Leak, resolution is significantly increased and the model can capture the details present in the Target, one can especially see the correctly raised pixel values in the third sample at the edges. We can also see how enriching the Segmentation improves the sharpness and guidance by comparing RichSeg and Seg, especially in the third example. As this helps with the utilization of RankSim by providing a more continuous label space and as a result, we see that the predictions are sharper and closer to the Target. We can observe that positional encoding PosEnc has the potential to capture sharp details at the expense of over-capturing details from Guide as well as the unwanted presence of positional encoding lines disrupting the perceptual quality. We can especially see a clear illustration of how positional encoding has the potential of incorporating details from the guide at undesired zones increasing the error in the dark zone in the first example.

8. Conclusion

According to the model benchmark experiment, on average, Pix-Transform with RankSim outperforms other methods in terms of SSIM and PSNR indicating a perceptual advantage. Optimal parameter choices seem to be the ones that favor smooth training and reduce overfitting. All in all, the contribution of this work is providing an investigation for a recipe to improve a self-guided super-resolution framework by utilizing RankSim regularization combined with external data that can guide the model into perceptually superior predictions. For future work, we leave the questions regarding the limitations of

Target-Source-Segmentation discrepancies, and overfitting vs. underfitting trade-off of this specific task.

9. Code

The code with the instructions can be reached at: github.com/SherryJYC/RefineVHM

10. Acknowledgements

I want to thank Prof. Jan Dirk Wegner for the opportunity to work on this project, Yuchang Jiang and Nando Metzger for their support and supervision throughout the project.

References

Dubayah, R., Blair, J.B., Goetz, S., Fatoyinbo, L., Hansen, M., Healey, S., Hofton, M., Hurtt, G., Kellner, J., Luthcke, S., Armston, J., Tang, H., Duncanson, L., Hancock, S., Jantz, P., Marselis, S., Patterson, P.L., Qi, W., Silva, C., 2020. The global ecosystem dynamics investigation: High-resolution laser ranging of the earth's forests and topography. Science of Remote Sensing 1, 100002. URL: https://www.sciencedirect.com/science/article/pii/S2666017220300018, doi:https://doi.org/10.1016/j.srs.2020.100002.

Gong, Y., Mori, G., Tung, F., 2022. Ranksim: Ranking similarity regularization for deep imbalanced regression. arXiv:2205.15236.

He, K., Sun, J., 2015. Fast guided filter. arXiv:1505.00996.

Lang, N., Jetz, W., Schindler, K., Wegner, J.D., 2022. A high-resolution canopy height model of the earth. arXiv:2204.08322.

de Lutio, R., D'Aronco, S., Wegner, J.D., Schindler, K., 2019. Guided super-resolution as pixel-to-pixel transformation. arXiv:1904.01501.

Metzger, N., Daudt, R.C., Schindler, K., 2023. Guided depth super-resolution by deep anisotropic diffusion. arXiv:2211.11592.

Patch Size	MSE	MAE	SSIM	PSNR
64×64	216.55	13.13	0.18	7.08
128×128	216.61	13.14	0.19	7.09
256×256	217.20	13.14	0.19	7.09

Table 1: Mean Performance Metrics for Different Patch Sizes

Model	MSE	MAE	SSIM	PSNR	RankSim Weight	RankSim Target	Pos Enc	Batch Size	Downscaling
Exp 14	236.32	12.86	0.12	6.62	0.3	segmentation	1	256	8
Exp 16	229.78	12.75	0.12	6.73	0.3	segmentation	1	32	8
Exp 15	215.78	12.56	0.13	7.05	0.3	segmentation	1	32	4
DADA	192.17	12.28	0.15	7.52	-	-	-	-	-
Source	192.50	12.28	0.15	7.50	-	-	-	-	-
Exp 20	198.82	12.33	0.16	7.41	0.3	source	1	32	4
Pix-Transform	192.41	12.28	0.16	7.51	0.0	-	0	32	4
Exp 13	199.58	12.40	0.17	7.39	0.3	segmentation	1	256	4
FGF	190.60	12.28	0.16	7.56	-	-	-	-	-
Exp 11	191.88	12.28	0.16	7.54	0.3	segmentation	0	32	4
Exp 18	191.95	12.28	0.17	7.55	0.3	source	0	32	4
Exp 7	191.87	12.28	0.17	7.55	0.3	rich segmentation	0	32	4
Exp 4	191.56	12.28	0.17	7.56	0.3	binned target	0	32	4
Exp 17	192.38	12.31	0.18	7.54	0.3	source	0	256	4
Exp 25 (PosEnc)	195.20	12.31	0.18	7.52	0.3	model output	1	32	4
Exp 12	192.98	12.29	0.18	7.53	0.3	segmentation	0	32	8
Exp 9	190.96	12.28	0.18	7.58	0.3	segmentation	0	256	4
Exp 19	192.94	12.29	0.18	7.53	0.3	source	0	32	8
Exp 23	191.38	12.28	0.18	7.58	0.3	model output	0	32	4
Exp 8	192.64	12.28	0.19	7.54	0.3	rich segmentation	0	32	8
Exp 10 (Seg)	192.39	12.28	0.19	7.55	0.3	segmentation	0	256	8
Exp 6 (RichSeg)	193.03	12.30	0.19	7.54	0.3	rich segmentation	0	256	8
Exp 21	191.59	12.30	0.19	7.56	0.3	model output	0	256	4
Exp 22	192.73	12.30	0.19	7.54	0.3	model output	0	256	8
Exp 24 (SelfSim)	**192.33**	**12.29**	**0.19**	**7.55**	**0.3**	**model output**	**0**	**32**	**8**
Exp 2	190.45	12.28	0.19	7.61	0.3	binned target	0	256	4
Exp 5	191.59	12.27	0.19	7.57	0.3	binned target	0	32	8
Exp 3 (Leak)	191.30	12.28	0.20	7.58	0.3	binned target	0	256	8

Table 2: Model Benchmark Experiment Results

Cover Type	MSE Leak	MSE Source	Mean Target	StdDev Target	Mean Source	StdDev Source	Abs Mean Diff
Crops	104.70	113.12	4.01	5.99	13.36	5.58	9.35
Built	119.35	118.53	2.82	4.38	12.84	5.20	10.02
Water	133.59	134.22	0.69	3.38	11.09	6.14	10.40
Bare	142.69	139.19	3.92	5.94	14.10	7.58	10.18
Flooded Vegetation	177.27	173.01	6.95	7.54	19.03	7.17	12.08
Grass	192.14	195.46	4.46	6.13	17.11	6.17	12.65
Trees	204.21	205.69	15.17	8.10	27.42	5.53	12.25
Shrub and Scrub	216.20	215.97	7.09	7.04	20.46	6.41	13.37
Snow and Ice	295.98	294.77	4.53	5.61	20.79	7.61	16.26

Table 3: Cover Type Statistics of Depth Values

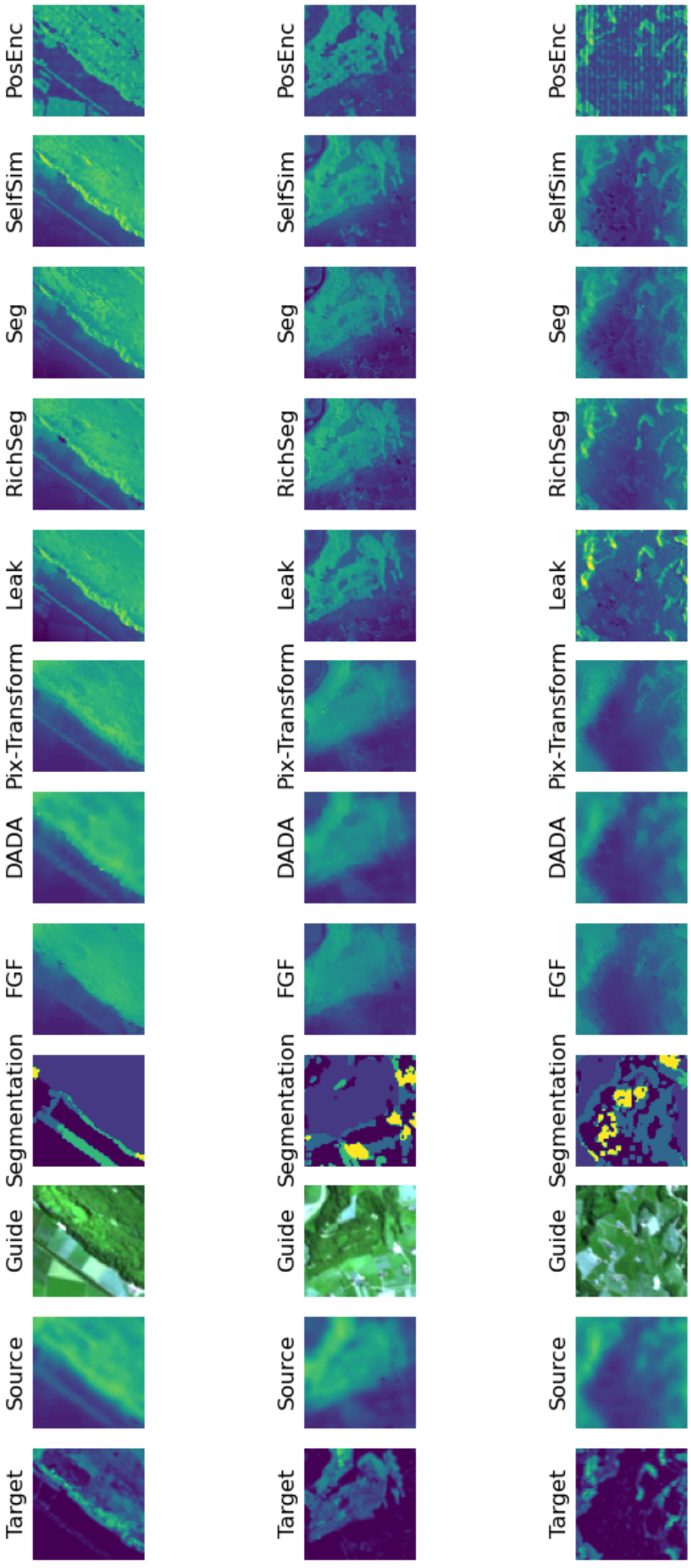

Figure 4: Model Predictions for 3 Samples