Improving Semantic Tasks using Weather Augmentation

Ashwin Wariar, Athreyi Badithela, Rajan Singh

1 Introduction

Land cover classification is a significant task in understanding the effects of climate change and other shifts in the environment over time. Land cover classification is a difficult task due to differences in features based on location, scale issues, and image resolution limitations (Foody, 2002). Remote sensing has been established as a method to obtain environmental data and many datasets have been built using satellite data since Landsat 1 was launched in 1972 (Goward et al., 2006).

In recent years, deep learning techniques have become more prominent in land cover classification due to advances in remote sensing technology and the ability of deep learning to handle and learn from large datasets. Some of the common techniques include convolutional neural networks (CNNs) and recurrent neural networks (RNNs) such as long short term networks (LSTMs) (Ienco et al., 2017). While there are many benefits, there are still issues with ground truth, resolution, and the overall nature of this data (Vali et al., 2020).

In this work, we used a UNet model and a hybrid UNet model with an LSTM layer to classify patches from the Sentinel-2 dataset. The dataset has thirteen spectral bands at 10m, 20m, and 60m resolution. These include visible (VNIR), near infrared (NIR), and short wave infrared (SWIR) bands. Our method is designed to learn from the different channels. The model was compared with a baseline random forest classifier model to show the differences between traditional machine learning techniques and deep learning.

2 Related Work

2.1 Deep Learning for Land Cover Use

In Alshari et al., an artificial neural network (ANN) with random forest method for land-use classification is introduced. This model utilizes the Sentinel-2 dataset as well. In their method, they aim to limit the number of layers to avoid the complexity and expense of deep learning. The ANN gives output and those samples are used to create the trees in the random forest. This hybrid model has a higher accuracy compared to the normal ANN when trained on the Sentinel-2 dataset (Alshari et al., 2023).

In Kussul et al., a deep learning technique is introduced based on an ensemble of CNNs to classify crops. The architectures of the CNNs are built for spectral and spatial features to utilize the dataset as effectively as possible. This ensemble method results in a 5.9 percent increase over a random forest model, exemplifying the benefits of deep learning when working on land classification tasks with large datasets. Discrimination between closely related crops was also done with more accuracy using the CNNs (Kussul et al., 2017).

2.2 Semantic Segmentation for Land Cover Mapping

Semantic segmentation is a task in the field of computer vision where each pixel in an image is classified into a specific category or class. The goal of semantic segmentation is to partition the image into meaningful regions based on the objects or areas they represent, assigning a label to every pixel. This task is crucial to the problem of land cover classification, where machine learning or deep learning models can take advantage of satellite imagery in order to accurately classify large areas of land.

Talha et al.'s work introduces an extended UNet architecture designed for semantic segmentation of satellite imagery in land cover classification tasks. The model, ADU-Net, incorporates dense decoder connections and an attention mechanism in order to enhance pixel-wise classification performance. Experimental results on the Gaofen Image Dataset used for testing showed a 4% increase over the original UNet in terms of mIoU and F1 score, highlighting the effectiveness of semantic segmentation and its applicability to land cover classification. (Talha et al., 2023)

In other work, Boonpook et al. propose LoopNet, a deep learning semantic segmentation algorithm for automatic land use classification using the Landsat 8 satellite image set. The proposed image architecture combines a convolutional loop and convolutional block in order to address the degradation problem in small land use features often seen with deep convolutional networks. Results showed that LoopNet outperformed baseline UNet and other advanced architectures, achieving an overall accuracy of 89.84% accuracy. These findings demonstrate LoopNet's superiority as well as the potential to improve medium spatial resolution imagery. (Boonpook, 2023)

In the vein of datasets for this specific problem, Johnson et al.'s work introduces OpenSentinelMap, a large-scale land use dataset designed for semantic segmentation of satellite imagery. (Johnson et al., 2022) The dataset combines Sentinel-2 multispectral images with labels derived from OpenStreetMap (OSM) annotations, providing per-pixel semantic labels for 137,045 globally distributed spatial cells. The labels span 15 categories, including land use, water, roads, and buildings, offering a comprehensive resource for land cover mapping tasks. The importance of this dataset lies in its scale and flexibility. Unlike prior datasets focused on single-label image classification or limited geographical scope, OpenSentinelMap emphasizes dense per-pixel semantic segmentation with broad, global coverage. This makes it particularly valuable for training deep learning models in supervised, semi-supervised, and unsupervised settings. By incorporating a variety of land use categories and leveraging high-resolution satellite imagery, the dataset facilitates research on land cover mapping.

2.3 Weather Inclusion in Models

While semantic segmentation has been shown to yield high accuracy when it comes to classifying land cover, researchers in the field have explored whether additional features can be concatenated with image data to yield higher accuracy when it comes to classification of land cover. This type of work comes under the umbrella of a type of machine learning known as knowledge-guided machine learning, where scientific knowledge is combined with black-box machine learning or deep learning models to induce higher accuracies in certain tasks.

Ravirathinam et al. introduces Weather-based Spatio-Temporal Segmentation Network with ATTention (WSTATT), a deep learning model that enhances crop mapping by incorporating weather data through an LSTM model and passing the model outputs via an attention layer. By combining weather information from Daymet and satellite imagery from Sentinel-2, WSTATT goes beyond traditional methods that rely solely on spectral imagery, integrating physical parameters such as weather and soil conditions to improve crop map accuracy. The attention mechanism allows for more effective use of temporal weather patterns, enabling the model to detect crop types up to five months in advance, which aids in food supply projections. (Ravirathinam et al., 2024)

In another paper, Zhou et al. explore the potential of integrating climate and remote sensing data for predicting wheat yield at a national scale in China. By using nine climate variables in addition to remote sensing data and three machine learning algorithms, authors investigate yield prediction from 2002 to 2010. The study demonstrates that integrating climate data throughout the growing season with remote sensing data improves yield predictions, particularly in winter wheat zones, and highlights the potential of scalable machine learning methods for agricultural yield forecasting. (Zhou et al., 2022)

3 Methodology

3.1 Datasets

The image dataset chosen for this problem was the OpenSentinelMap dataset (Johnson et al., 2022). This dataset contains multiple multispectral images from the Sentinel-2 satellite over a four-year period, annotated with per-pixel semantic labels that were obtained from the OpenStreetMap dataset. Originally, the dataset that was chosen for the problem was OpenEarthMap, a benchmark dataset that also contains satellite images, which were associated with manually annotated land cover labels (Xia et al., 2022). OpenEarthMap was originally chosen due to the large amount of data available, with 8000 images, as well as the fact that the resolution of the images were very fine, with the class labels having a 0.25-0.5 meter ground sampling distance. In addition, the manually-annotated labels meant that the accuracy of the labels could be trusted more than other datasets. However, the main downside to using OpenEarthMap came from the lack of temporal information included in the dataset. Since the problem statement was concerned with using weather data to hopefully improve land cover classification accuracy, which is data gathered temporally, it would've been extremely difficult to integrate OpenEarthMap data with weather data. Therefore, the decision was made to use OpenSentinelMap. OpenSentinelMap contains more coarse resolutions for their images, ranging from 10 meters to 60 meters. The OpenSentinelMap dataset provides Sentinel-2 imagery compressed into yearly .tgz files, which, when extracted, organize data by MGRS tiles and spatial cells. Each .npz file contains 32-bit float Bottom-of-Atmosphere imagery data, with spectral bands grouped by spatial resolution (10m, 20m, 60m) and additional metadata such as a Scene Classification Layer ("scl") for pixel quality assessment. Labels are available in PNG format with mappings detailed in a JSON file, and auxiliary metadata includes spatial cell information such as latitude and longitude ranges and a pre-defined train/test split at the MGRS tile level to prevent data leakage. The main benefit of OpenSentinelMap is that it contains limited temporal data for when the images were taken, making it a lot easier to grab the corresponding weather data to concatenate with the image data.

For the weather data, the ERA5-LAND dataset was chosen (Muñoz Sabater et al., 2021). The ERA-5 dataset is a high-resolution global reanalysis dataset for land monitoring applications, which include land cover classification. It spans from 1950 to the present with hourly temporal resolution and a spatial resolution of 9 km, offering detailed insights into land surface characteristics, water and energy cycles, and atmospheric conditions.ERA5-Land improves upon previous datasets like ERA5 by offering better accuracy in soil moisture, lake, and river discharge estimations, with notable enhancements in coastal regions, and is widely used for hydrological studies, climate modeling, and environmental management. The dataset was chosen due to its extensive coverage of data temporally and spaitially. In addition, there are multiple features that can be taken, but for the purposes of the problem, hourly temperature and precipitation were the features chosen.

3.2 Data Preprocessing

While exploring the dataset, it became extremely difficult to download the dataset, as it was around 100 GBs for each year , and it was difficult to utilize MSI resources in order to download the data, meaning that all preprocessing and training was done locally. Because of these setbacks, a subset of the dataset was used for training, specifically one tile from the OpenSentinelMap, 11SKA, for the year 2019. The 11SKA tile is located in California, and contains agricultural, residential, commercial, recreational, and other natural land. The amount of classes makes it a viable tile to use in order to test the generalizability of our models. One issue with the 11SKA tile was that there were only 14 samples for that tile. Given that it would be extremely difficult to train and test with only 14 samples, each image was split into 9 patches of equal size to increase the sample size from 14 to 126. Once each sample was split, each of the three channels were concatenated with each other so that each patch would have information from all of the bands and channels. In addition, the target label image was also cut into 9 patches to match the input patches, and each of the input patches were resized in order to match the resolution of the target label patches.

For the weather data, the coordinates for the 11SKA were grabbed from OpenSentinelMap spatial cell information. Once the coordinate ranges were obtained, we used Google Earth Engine in order to import the ERA5-Land dataset and use the temperature and precipitation features. Specifically, the temperature feature measures the air temperature at 2 meters above surface level in Kelvin, and the precipitation measures the total precipitation. Both of these features were available hourly. After establishing a rectangle of land using the coordinate ranges for the 11SKA tile, the relevant weather data was collected and aggregated to reflect daily values. As stated before, each sample in the 11SKA tile contained some temporal information, including when the image was taken. This information was used in order to grab the corresponding weather data for that day as well as the previous 30 days in order to be fed into an LSTM model. In order to account for the mismatch in weather samples and patch samples (14 vs. 126), we simply repeated the weather data 9 times for each patch so there would be an equal amount of weather and image samples.

3.3 Models

Three models were used to train and test our dataset. The first model is a decision tree classifier. A decision tree classifier is a machine learning algorithm that makes predictions by splitting data into subsets based on feature values, creating a tree-like structure where each node represents a decision rule, and the leaves represent the final predictions. (Breiman, 2001) This model was chosen because it was simple in its implementation, as well as having a relatively simple architecture compared to neural networks and LSTMs, making it easier to explore where the model is learning. Specifically, the model is trained for multiple max-depths ranging from 5 to 25.

The second model used is a UNet model. The UNet is a convolutional neural network (CNN) architecture designed for image segmentation tasks, where the goal is to label each pixel of an image with a corresponding class. (Ronneberger et al., 2015) The UNet has a symmetric encoder-decoder structure that resembles the letter "U." The UNet was chosen due to its generalizability in many use cases including land cover classification, as well as its high performance on limited data, which was especially useful for our problem and dataset. The purpose of this model was to test only the image data as a baseline against the concatenated weather and image data. Specific to this problem, the baseline UNet used had three encoder blocks, a bottleneck, and three decoder blocks. Each encoder block consists of two convolutional layers, batch normalization, and ReLU activation, while the decoder blocks follow a similar structure, combining upsampled features with corresponding encoder features using skip connections. The upsampling is achieved with transposed convolution layers, and the bottleneck layer has two convolutional layers for the most abstract feature representation. Finally, a fully connected layer transforms the output of the decoder structure into a classification vector per each pixel. The softmax of this classification vector gives a one-hot encoding classification for that pixel.

The final model used is a modified UNet model combined with an LSTM layer. An LSTM model is a type of recurrent neural network (RNN) architecture designed to learn and make predictions on sequential data. (Hochreiter and Schmidhuber, 1997) The LSTM layer was added to account for the weather, which was represented as time-series data. Similar to the baseline UNet model, it has three encoder blocks, each consisting of two convolutional layers, batch normalization, and ReLU activation, followed by a bottleneck layer with two convolutional layers. The key difference lies in the addition of an LSTM-based pathway, which processes temporal weather data into a fused feature representation using a single-layer LSTM followed by a fully connected layer. This output is reshaped and concatenated with the bottleneck features, enabling the model to combine spatial and temporal data for segmentation. With this modified model, it is possible to pass in both image and weather data to the model and output a prediction on the pixel-wise classification of an image patch.

In terms of evaluating the performance of the three models, two metrics were used: mean accuracy and mean intersection over union (mIoU). Mean accuracy measures the average classification accuracy across all classes, providing an explanation of how well the model predicts each class independently. This metric is useful for determining whether the model predicts well across all classes, especially for datasets with class imbalances, which fits T11SKA. mIoU, on the other hand, evaluates the overlap between the predicted and ground truth regions for each class, averaged across all classes. IoU is calculated as the ratio of the intersection (common area) to the union (total area) of the predicted and actual pixels for a given class. This metric is widely used in semantic segmentation tasks because it provides a more nuanced measure of spatial accuracy, capturing both false positives and false negatives in the predictions

3.4 Training and Hyperparameters

The OpenSentinelMap dataset contains 18 classes, but these classes are not uniformly distributed across all tiles. To identify tiles with a balanced distribution of vegetated and non-vegetated pixels, the scene classification layer was utilized, which also provides information on the annotation quality for each pixel. The Californian region was selected due to its favorable balance of annotation quality and pixel distribution. However, accessing the scene classification layer requires extracting data from the 100 GB set, a process that is computationally intensive on locally available machines. As a result, the selection of tile T11SKA was also informed by the tiles used in (Ravirathinam et al., 2024).

After obtaining 14 samples from T11SKA and patching each sample with a 3x3 grid, 126 samples were created. Each image was mapped to its corresponding label image from the OpenSentinelMap dataset. The label images have dimensions 3 x 196 x 196 (channels x height x width), where each channel represents a sub-class label: land-use, roads and water, and buildings. The land-use and buildings

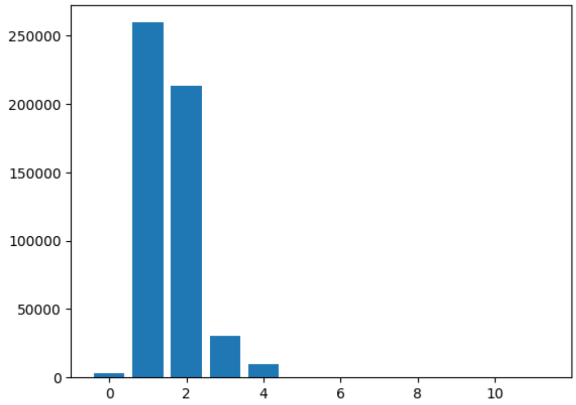

Figure 1: Distribution of class labels used in Training. Labels are as follows: 0 - None, 1 - Agriculture, 2 - Residential, 3 - Commercial, 4 - Recreation. Other class labels are not present in T11SKA.

labels were combined into a single channel, which served as the ground truth classifications. The classification indices were then remapped in sequential order for training with cross-entropy loss. The final label distribution is shown in Fig 1.

The class labels for T11SKA include agriculture, residential, commercial, and recreational use, with a few labels marked as unknown. These unknown labels either lack classification or may belong to the road and water subclasses, which are excluded from this project. A notable issue in this dataset is the very low number of samples given the high resolution (each sample has a size of 12 x 64 x 64 after patching), primarily due to the number of GSD layers used. In addition, there is a significant class imbalance, particularly in the commercial and residential classes. This class imbalance, combined with the simplified nature of the dataset, raises concerns about the potential for overfitting with a UNet model.

The following hyperparameters were used for both the baseline UNet and the modified UNet with LSTM model to ensure consistency and reproducibility. Both models were trained for 300 epochs with a learning rate of 0.0001, using the Adam optimizer for gradient descent, and cross-entropy loss was employed for multiclass classification. A batch size of 8 was found to be optimal for both models. The convolutional layers of the UNet began with a base channel size of 64, increasing by a factor of 2 for each encoding layer up to a bottleneck of 512, and then decreasing back to 64. For the modified UNet, only two encoding layers were used, resulting in a bottleneck of 256, which was concatenated with the weather embeddings to achieve a size of 512. The concatenated embeddings were then reduced using up-convolutions and the decoder to 64 channels. The fully connected layer maps the final 64 channels to 5. To prevent overfitting, dropouts and weight decay were applied in the Adam optimizer. A dropout rate of 0.2 was optimal for the baseline UNet, while 0.3 was optimal for the modified UNet.

To address the class imbalance, we employed two techniques: stratified sampling and class weights. Initially, sklearn's train_test_split was used to partition the dataset into training, validation, and test sets. As shown in Fig 1, the class distributions are highly imbalanced. When T11SKA is patched to create additional samples, some patches consist entirely of the majority class. Since models are saved based on the best validation loss and the number of samples is low, there is a risk that the validation set may be dominated by patches of the majority class, causing the model to overfit and predict only the majority class. To improve training stability, a custom splitting function was developed. This function ensures that each dataset (train, validation, and test) contains a proportionate representation of each class. For instance, if there are 14 images containing agriculture and residential labels, they are split into 8 for training, 3 for validation, and 3 for testing. This allowed the model performance to be consistent across multiple runs with same hyper-parameters. The datasets are also normalized based on the train dataset's channel-wise mean and standard deviation.

Class weights based on label distribution were attempted to prevent the model from neglecting minority classes. However, due to the low sample size and small batch size, the majority classes were given more focus, as the disproportionately large weights assigned to the minority classes were exploited. Combined with the model being saved based on the best validation loss, class weights could not be effectively utilized. It was observed that the model performed well without class weights, so the decision was made to proceed without them.

A cosine annealing learning rate scheduler was also experimented with to facilitate smoother gradient updates during training. However, this approach caused certain batches to have more influence than others, leading to some classes being ignored. As a result, a constant learning rate of 0.0001 was maintained throughout training.

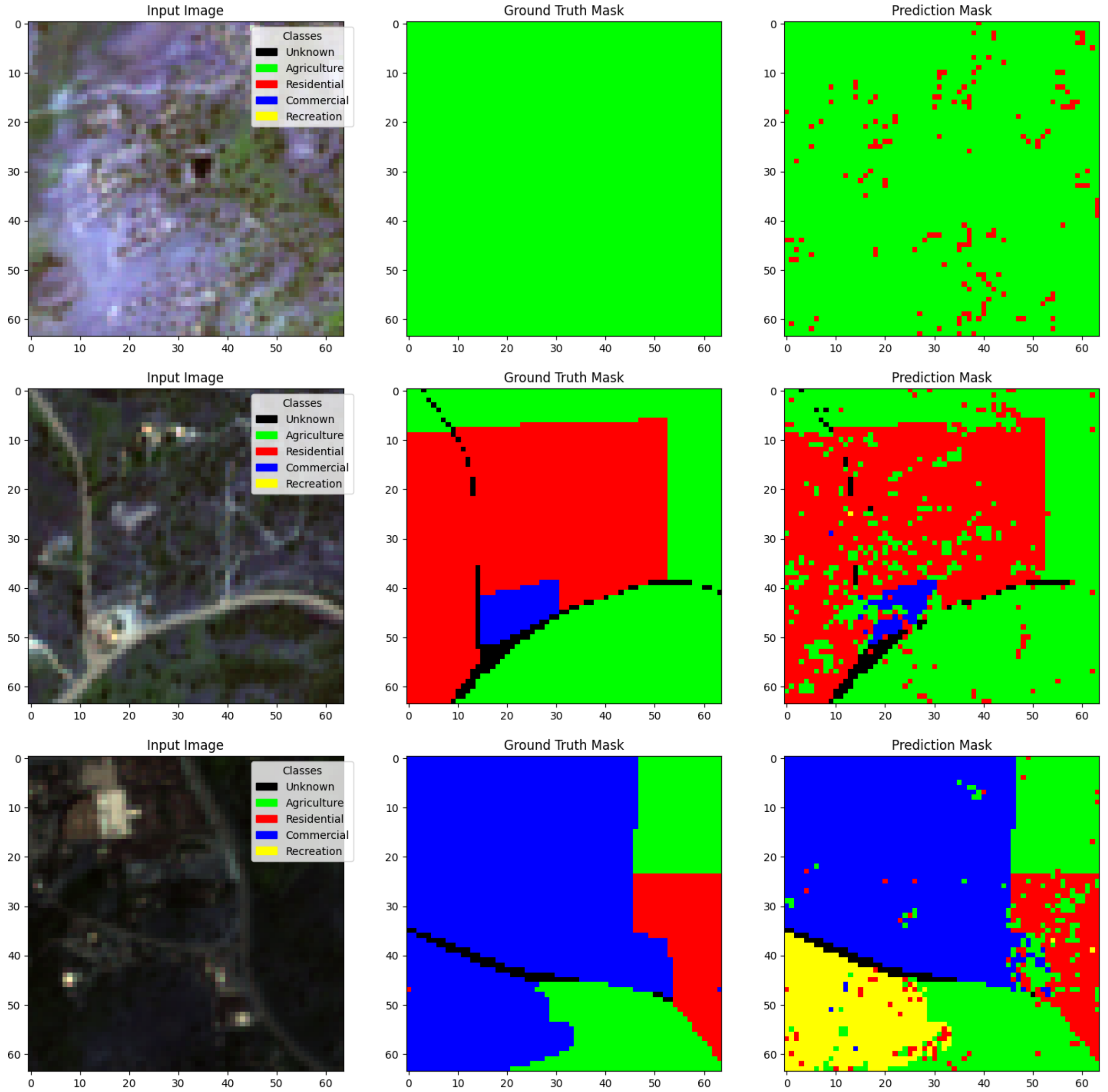

Figure 2: **Decision Tree Predictions**. From left to right, input image, ground truth segmentation map, and decision tree segmentation predictions. From top to bottom, there are three different images showcasing three examples, image with only one majority label, majority and minority label, and all labels.

Sci-kit learn decision tree was used as the decision tree algorithm as a baseline comparison.

4 Results

The trained models are evaluated on a separate portion of the dataset that is kept aside specifically for testing purposes. Evaluation metrics are divided into two categories, label-wise and image-wise. Two metrics are calculated: Intersection over Union (IoU) and accuracy. In total, there are three distinct metrics, label-wise accuracy, image-wise accuracy and image-wise mean IoU.

Label-wise metrics evaluate the performance of the model on a per-class basis, measuring how well each class is predicted relative to the ground truth.

Image-wise metrics assess the performance at the image level, focusing on how accurately the model predicts all images.

In addition to these quantitative metrics, qualitative results are provided by visualizing the model's predictions. These visualizations help illustrate how well the model is able to segment and classify different regions within the images, offering insights into areas where the model performs well or struggles. This combination of quantitative and qualitative evaluation ensures a comprehensive understanding of the performance of the model.

4.1 Decision Tree

Sci-kit Learn's decision tree is adapted for use in semantic segmentation on this dataset by flattening the input data to a shape of (64 * 64, 12). This transformation allows the decision tree to treat each pixel as an individual sample, with the 12 channels serving as the features on which the decision tree makes splits. As a result, the decision tree effectively operates with a much larger sample size compared to the UNet models, since it considers each pixel separately.

To determine the optimal depth for the decision tree without overfitting, the maximum depth is varied from 5 to 25 in increments of 5. The performance metrics—image-wise accuracy, image-wise IoU, and label accuracy—are plotted against tree depth in Fig 3. A maximum depth of 20 is chosen to avoid overfitting. It is noted that the decision tree is capable of learning the problem effectively with sufficient depth, indicating that the problem is relatively simple. This result supports the hypothesis that the problem is simple after reducing the dataset to T11SKA. Consequently, efforts were focused on preventing overfitting in the UNet models.

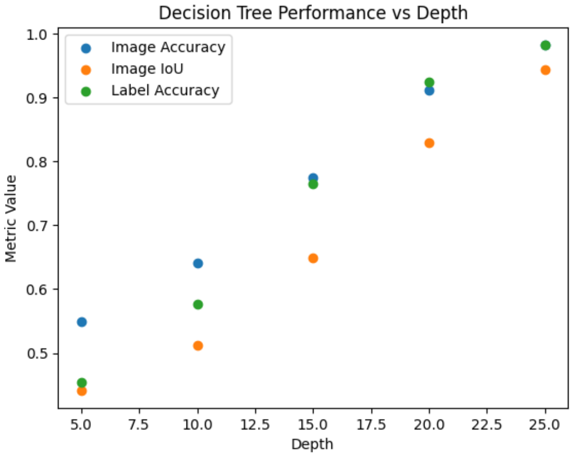

Figure 3: Scatter plot showing image-wise accuracy, image-wise IoU, and label accuracy over the depth of the decision tree.

In Fig 2, the results of the decision tree model are shown for an image containing one label, two labels, and multiple labels. It is important to note that the decision tree model exhibits significant salt-and-pepper noise in its predictions

4.2 UNet

In Fig 4, the qualitative predictions of the UNet model are shown for images containing a single label, two labels, and multiple labels. It can be observed that the UNet tends to expect most images to contain both majority classes, as evidenced by the image with only a single label, where the model still predicts both majority classes. Additionally, the model under-predicts minority classes in some cases, as seen in the image containing two labels.

We also found it challenging to tune the UNet model without overfitting. While the decision tree model had a sufficient number of samples to learn from, the UNet had significantly fewer high-resolution samples, making learning more difficult. Furthermore, the UNet was highly sensitive to class weights and learning rates during training, resulting in a model that was difficult to train and required careful fine-tuning to avoid instability.

Additionally, we experimented with replacing the fully connected layer with a convolutional layer as the head of the model. The convolutional layer was expected to be better suited for capturing local patterns and features, and it would also produce predictions that are less sensitive to salt-and-pepper noise. Coupled with a more focal loss, the model could potentially adapt better to spatial patterns. However, we found that the convolutional layer made the model more difficult to generalize, leading to lower accuracy. As a result, we decided to revert to the linear layer originally implemented.

4.3 UNet with LSTM

Initially, we attempted to add an LSTM layer to the UNet without modifying the structure of the original model. However, this approach led to overfitting, with the model skewing predictions toward the majority class. Despite experimenting with hyperparameter tuning, the best performance was achieved after reducing the model's complexity by removing one layer from both the encoder and the decoder. We also explored changing the base channels in the modified UNet but encountered difficulties in tuning it to match the performance of the baseline UNet.

In Fig 5, we can see that the model performs better for the agricultural class, with predictions showing improvements. However, predictions for other classes are more patchy, and the edges exhibit significant salt-and-pepper noise. This suggests that the agricultural label benefits from the

	DT	UNet	**Unet w/ LSTM**
Image-wise Acc.	**91.19**	80.11	**86.91***
Image-wise mIoU	**82.87**	69.65	**78.28***
Label-wise Acc.	**92.49**	89.34	**89.73***

Table 1: Quantitative results for Decision Tree (DT), UNet and Unet with LSTM. The higher the metrics the better. All metrics are in percentage.

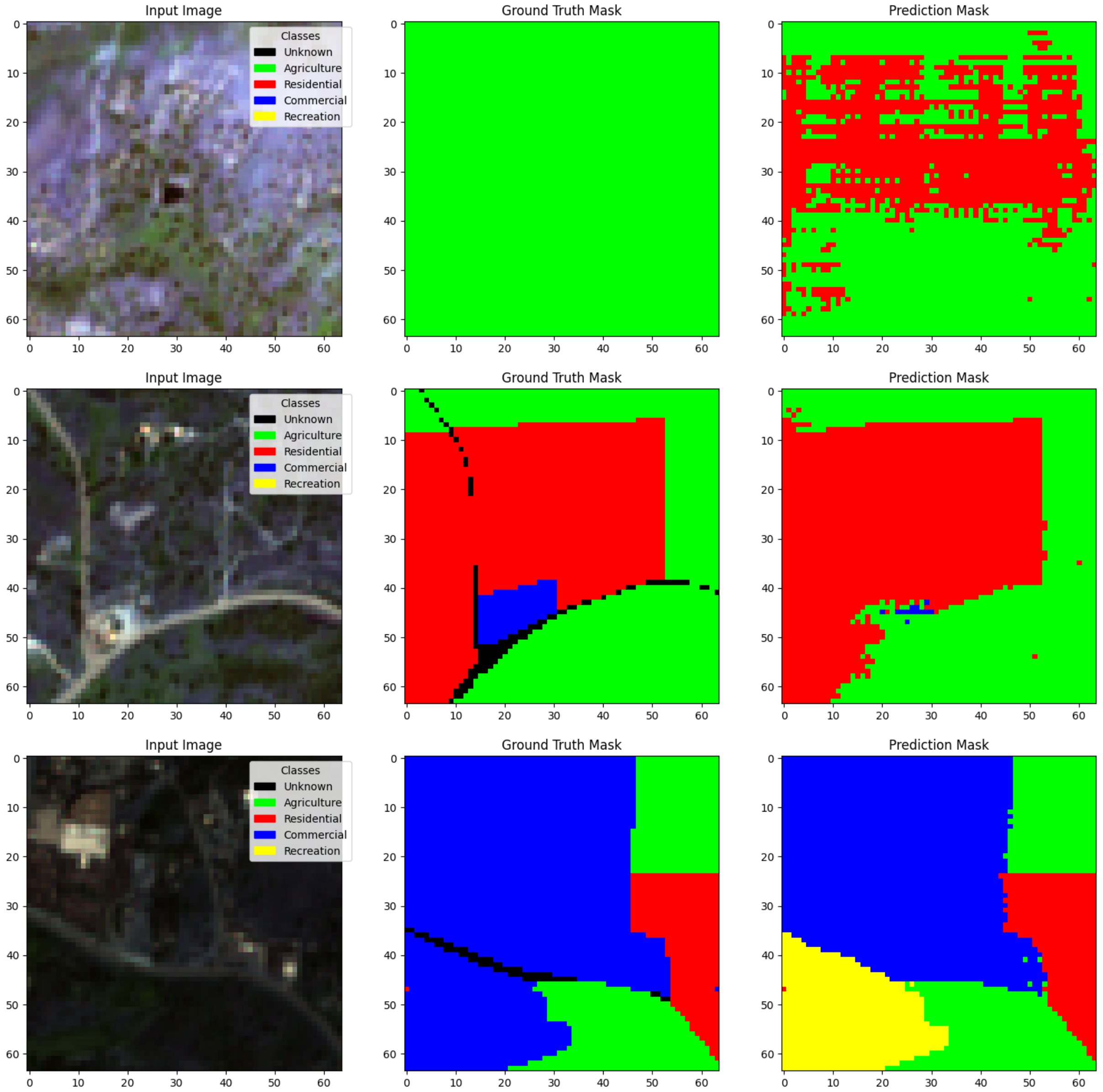

Figure 4: **UNet Predictions**. From left to right, input image, ground truth segmentation map, and UNet segmentation predictions. From top to bottom, there are three different images showcasing three examples, image with only one majority label, majority and minority label, and all labels.

weather data, even though the data is limited to only 30 timestamps prior to the image's timestamp. Nonetheless, the baseline UNet outperforms the modified UNet in predicting the overall shapes of the segmentations. A more detailed discussion of these findings is provided in the Analysis section, alongside the quantitative results.

5 Analysis

In Table 1, we observe that the decision tree performs the best based on our quantitative metrics. However, from the qualitative analysis, it is clear

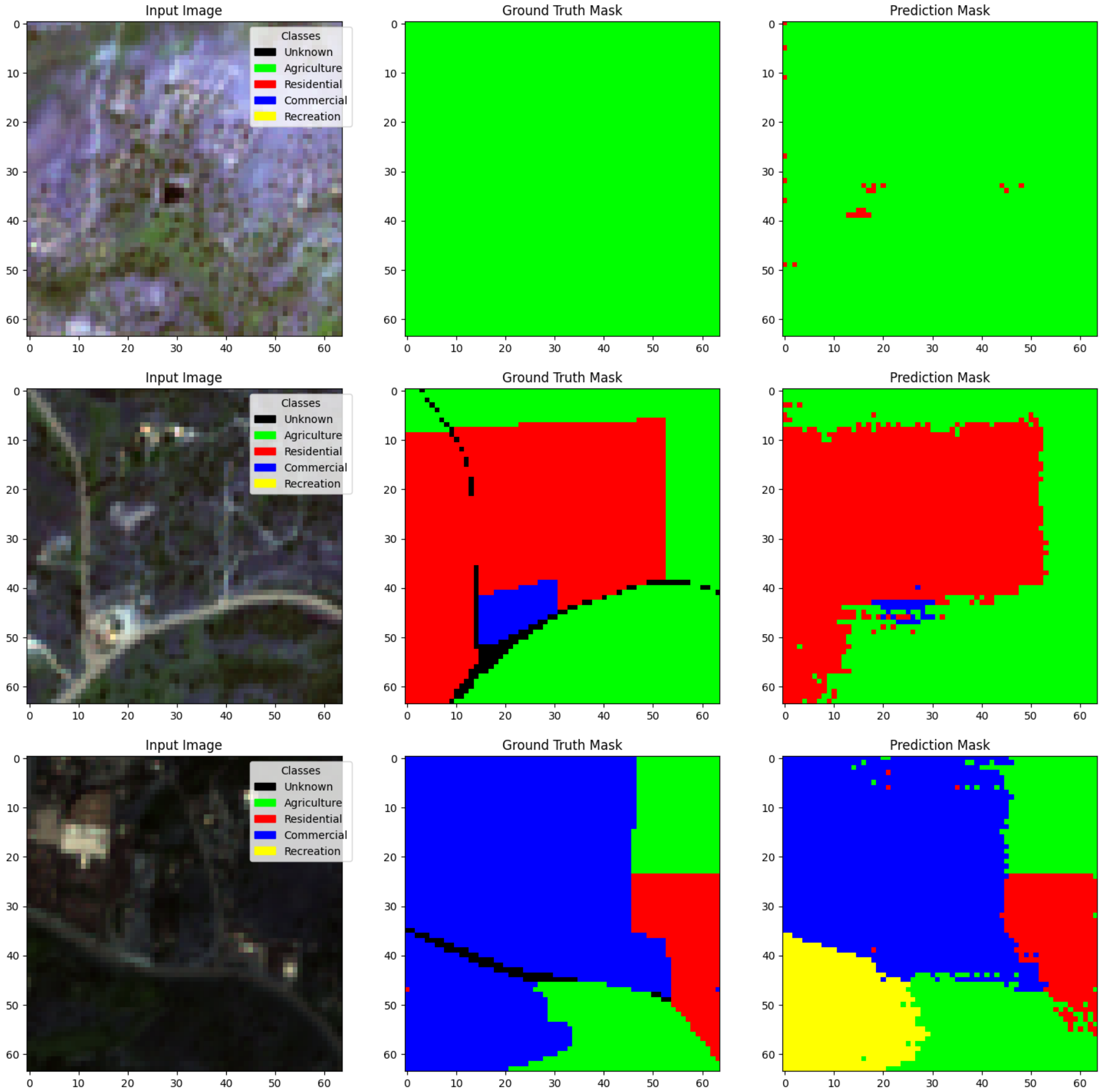

Figure 5: **UNet with LSTM Predictions**. From left to right, input image, ground truth segmentation map, and UNet with Lstm segmentation predictions. From top to bottom, there are three different images showcasing three examples, image with only one majority label, majority and minority label, and all labels.

Labels	DT	UNet	Unet w/ LSTM
Agricultural	**94.94***	72.95	**96.01**
Residential	92.21	**97.33**	85.92
Commerical	**92.49***	**92.78**	89.67
Recreational	**93.33***	**94.31**	87.31

Table 2: Class label wise accuracy for the Decision Tree, UNet, UNet w/ LSTM. The higher the metrics the better. All metrics are reported as percentage.

that the UNet model better captures spatial patterns, despite all three models exhibiting salt-and-pepper noise in their predictions. The UNet and UNet with LSTM models can be easily adapted to reduce this noise by replacing the final linear layer with a convolutional layer, though tuning this adaptation can be challenging.

Interestingly, Table 2 reveals that the UNet performs well for most classes, except for the agricultural class, where it tends to underpredict in comparison to the other classes. Despite not using any class weights, the UNet model performs well in predicting the minority classes. The UNet with LSTM, however, outperforms the other models in predicting the agricultural class, suggesting that weather data immediately benefits the model in identifying agricultural pixels. Even with a small sample size and limited images containing only agricultural pixels, the model can predict most of the image as agricultural land. Additionally, only one month of weather data and a single satellite image were used, which may not fully capture seasonal variations relevant to the crops being grown.

In all the qualitative analysis figures (2, 4, 5), it is noticeable that all models misclassify a portion of the commercial area as recreation in images containing multiple labels. The reason for this error is unclear, but a possible explanation is that distinguishing between land used for commercial or recreational purposes can be challenging, particularly when both areas are relatively open and do not change significantly throughout the year. Moreover, in the ground truth image, the commercial area is not easily identifiable either.

Had this project followed the plan of using the original dataset, the UNet-based models would have been easier to train and adapt to multiple classes, and with sufficient samples, they would not have been as susceptible to overfitting. It would also have been simpler to implement class weights and learning rate schedulers to address class imbalances. While the decision tree shows good accuracy, its prediction quality is less consistent compared to the UNet-based models. Furthermore, scaling the UNet models to large datasets is more feasible than scaling the decision tree for such tasks.

6 Conclusion and Future Work

This project explored the effectiveness of different machine learning models for land cover classification using the OpenSentinelMap dataset, comparing a baseline Random Forest classifier, a standard UNet model, and a modified UNet model with an additional LSTM layer that incorporated temporal weather data. The results demonstrated that the UNet w/ LSTM outperformed the UNet in all metrics, but could not beat the decision tree. However, the predictions of the UNet w/ LSTM are more coherent than the decision tree.

This suggests that the UNet with LSTM can be more generalized and improved on by using weather for agricultural labels. However, the hybrid model underperformed when compared to the standard UNet for other labels. This suggests that the inclusion of weather data overcomplicated the model for this specific problem. Land cover classification for a single sentinel tile, as defined by the dataset and task, appeared to be sufficiently simple for a decision tree to perform well without the need for additional temporal features. The added complexity of integrating weather data introduced more noise for non-agricultural labels, particularly given the small dataset size, and straightforward nature of the classification task.

These findings highlight the importance of evaluating the complexity of the problem before introducing additional data modalities or advanced architectures. While the hybrid model demonstrated the possibility to combine spatial and temporal data, its limitations show that either a more complex problem or larger dataset is needed to justify such a model.

This presents lots of opportunities for future work for this specific problem. As stated before, these models were only tested on a very small subset of the entire OpenSentinelMap, with only one tile out of hundreds being tested for only one year. The obvious next step would be to adapt this project to account for more tiles from different years of data, and test the generalizability of the modified hybrid UNet with LSTM model. Future work could also explore applying the hybrid model to datasets with greater temporal variability and information, or more challenging classification tasks, where the inclusion of temporal weather data or other external data might yield more significant improvements.

Overall, this project highlights the potential of hybrid spatial-temporal models in land cover classification and sets the stage for more expansive studies that can better use the wealth of available satellite and environmental data.

References

Eman A Alshari, Mohammed B Abdulkareem, and Bharti W Gawali. 2023. Classification of land use/land cover using artificial intelligence (ann-rf). *Frontiers in Artificial Intelligence*, 5:964279.

Wuttichai et al. Boonpook. 2023. Deep learning semantic segmentation for land use and land cover types using landsat 8 imagery. *ISPRS International Journal of Geo-Information*, 12(1):14.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Giles M. Foody. 2002. Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, 80(1):185–201.

Samuel Goward, Terry Arvidson, Darrel Williams, John Faundeen, James Irons, and Shannon Franks. 2006. Historical record of landsat global coverage. *Photogrammetric Engineering & Remote Sensing*, 72(10):1155–1169.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Dino Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel. 2017. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1685–1689.

Noah Johnson, Wayne Treible, and Daniel Crispell. 2022. Opensentinelmap: A large-scale land use dataset using openstreetmap and sentinel-2 imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1333–1341.

Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. 2017. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.

J. Muñoz Sabater, E. Dutra, A. Agustí-Panareda, C. Albergel, G. Arduini, G. Balsamo, S. Boussetta, M. Choulga, S. Harrigan, H. Hersbach, B. Martens, D. G. Miralles, M. Piles, N. J. Rodríguez-Fernández, E. Zsoter, C. Buontempo, and J.-N. Thépaut. 2021. Era5-land: a state-of-the-art global reanalysis dataset for land applications. *Earth System Science Data*, 13(9):4349–4383.

Praveen Ravirathinam, Rahul Ghosh, Ankush Khandelwal, Xiaowei Jia, David Mulla, and Vipin Kumar. 2024. Combining satellite and weather data for crop type mapping: An inverse modelling approach.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation.

Muhammad Talha, Farrukh A. Bhatti, Sajid Ghuffar, and Hamza Zafar. 2023. Adu-net: Semantic segmentation of satellite imagery for land cover classification. *Advances in Space Research*, 72(5):1780–1788.

Ava Vali, Sara Comai, and Matteo Matteucci. 2020. Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review. *Remote Sensing*, 12(15):2495.

Junshi Xia, Naoto Yokoya, Bruno Adriano, and Clifford Broni-Bediako. 2022. Openearthmap: A benchmark dataset for global high-resolution land cover mapping.

Weimo Zhou, Yujie Liu, Syed Tahir Ata-Ul-Karim, Quansheng Ge, Xing Li, and Jingfeng Xiao. 2022. Integrating climate and satellite remote sensing data for predicting county-level wheat yield in china using machine learning methods. *International Journal of Applied Earth Observation and Geoinformation*, 111:102861.