Fall 2022 Capstone Project

Final Report

Automatic landcover change detection and classification from Satellite images

JPMC 3

Data Science Institute
COLUMBIA UNIVERSITY

JPMORGAN CHASE & CO.

Team members:

Binny Naik, bn2341
Gangadhara Reddy Velagala, gv2310
Nishi Amish Modi, nam2238
Sai Prashanth Pathi, sp3915
Vishwas Reddy Thuniki, vt2356

Industry Mentor:
Dr. Saba Rahimi

Faculty Mentor:
Prof. Adam S. Kelleher

Table of contents

1. Introduction

1.1 Project Scope and Goal

This Capstone project is a collaboration between Columbia University at the Data Science Institute and JPMorgan Chase & Co. The objective of this Capstone Project is to build a Deep Learning model to automate the land cover change detection and classification from multitemporal, multiresolution, and multispectral satellite imagery over the Kigali region of Africa.

The goal of our project is to establish a semantic segmentation model in a specific region of interest describing land cover change between two consecutive time frames (2016 and 2019) using NICFI (Norway's International Climate and Forest Initiative) images. Specifically, we were expected to detect the loss or gain of "tree canopy" land cover class. The scoring metric is the average intersection over union (IoU) between the predicted and ground truth labels. This problem is made difficult by the very different spectral characteristics of the two years which have been derived from the dynamic word labels as well as the lack of high resolution satellite images for training purposes.

The 5m resolution NICFI imagery dataset from 2016 and 2019 (input images) and the Dynamic World Labels (target images) are being used for model training. The scope of the project is to classify the land cover into Tree Canopy, Buildings/Impervious land and Water for each pixel by building a model and then using it to detect land cover change for each class from satellite images of 2016 and 2019. We rely on Dynamic World images for model training purposes, however, we also manually annotate a set of 14 images to ensure that the model performance does not deviate with the real world ground-truth labels.

We also focus on applying exploratory data analysis strategies to explore our data and find any interesting and meaningful patterns in the images. After researching from multiple resources we decided to train a simple neural network using pixel values as our baseline model for detecting land cover change. We trained a few models using Unet and FCN (Fully Convolutional Network) using patches as well as compressed images for detecting the land cover change. These models have been trained using various pre-trained architectures as their backbone to gain the desired results. The project will include the use of Google Earth Engine (GEE) for the manual extraction of images. We decided to use Google Earth Engine as it combines a multi-petabyte catalog of satellite imagery and geospatial datasets with planetary-scale analysis capabilities. The result of our project is a deep learning model that will enable us to automatically detect the regions affected by deforestation in Kigali in 2019 as compared to 2016.

2. Data

2.1 Data Collection

All the data has been manually extracted from Google Earth Engine using NICFI and Dynamic World V1 satellite images. For this project we have considered two different time periods of 2016 and 2019 to capture the effect of land cover change in the region of Kigali, Rwanda. We have specifically only selected the data for the month of June/July from both years. The reason for selecting the same season/time frame is for true comparison of land change ignoring the effect of seasonality. Example: Spring season cannot be compared to Winter season as the vegetation cover will be different because of natural reasons.
The Area of Interest covering 3875.32 km2 area has been split into 256 smaller images for the purpose of exploratory data analysis and modeling.

Data sources:
NICFI: This image collection provides access to 5m resolution satellite monitoring of the tropics for the primary purpose of reducing and reversing the loss of tropical forests, contributing to combating climate change, conserving biodiversity, contributing to forest regrowth, restoration and enhancement, and facilitating sustainable development.

Dynamic World: This is a 10m near-real-time (NRT) Land Use/Land Cover (LULC) dataset that includes class probabilities and label information for nine classes.

Ground Truth Labels: The images sourced from NICFI are manually labeled into 3 classes using Ground Works. For the annotation we manually mark regions which are trees, water and buildings/land/impervious using NICFI as reference. This is a tedious and time consuming process as each image has to be looked at in detail and labeled accordingly. These manually labeled images are used as ground truth labels. Each team member has labeled individual images hence we can expect some annotator bias in the ground truth labels.

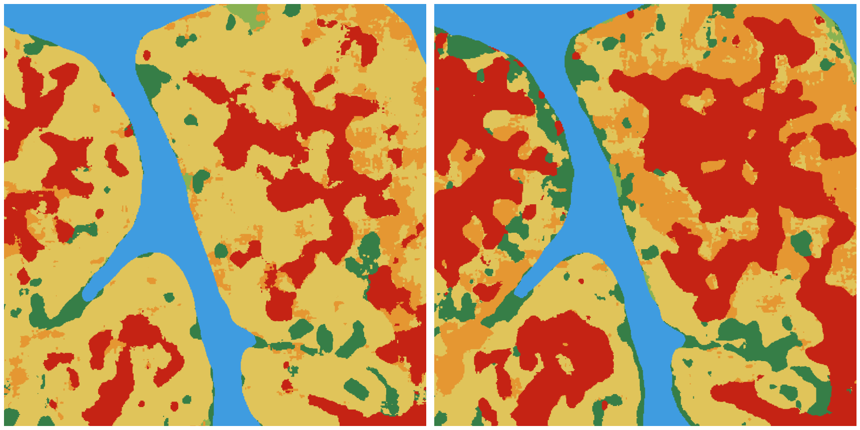


Figure 1: In the first row we have NICFI 2016 image on left and 2019 on right. In the bottom row we have labeled Dynamic world 2016 image on left and 2019 on right.

2.2 Data Cleaning

All the images had to be cleaned and standardized before processing. The cleaning and wrangling tasks included -

- Selecting the desired number of pixels and area covered per image
- Constant time frame of extracted images for each year
- Selecting the appropriate bands from each satellite
- Extracting image in multiple formats as TIF, PNG, etc
- Each satellite/source has separate preprocessing

2.3 Exploratory Data Analysis

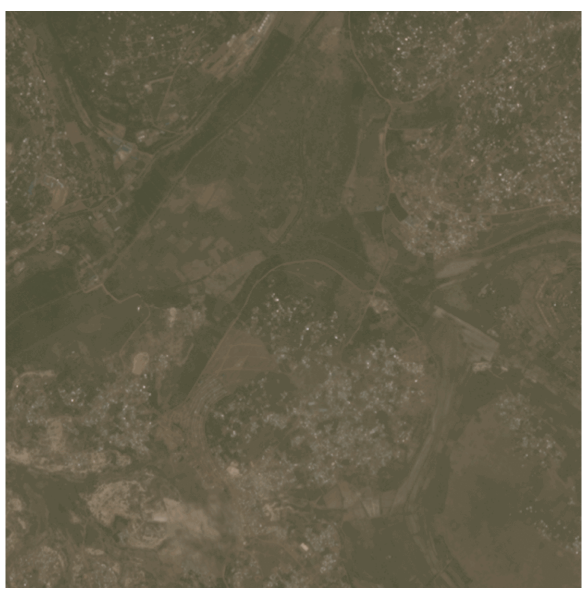


Figure 2: GIF showing the land cover of two different time periods using satellite imagery

In the reference paper [1] there are 4 classes for land cover each being water, tree cover, low vegetation and impervious. Let's look at the class distribution for our Area of Interest (AOI).

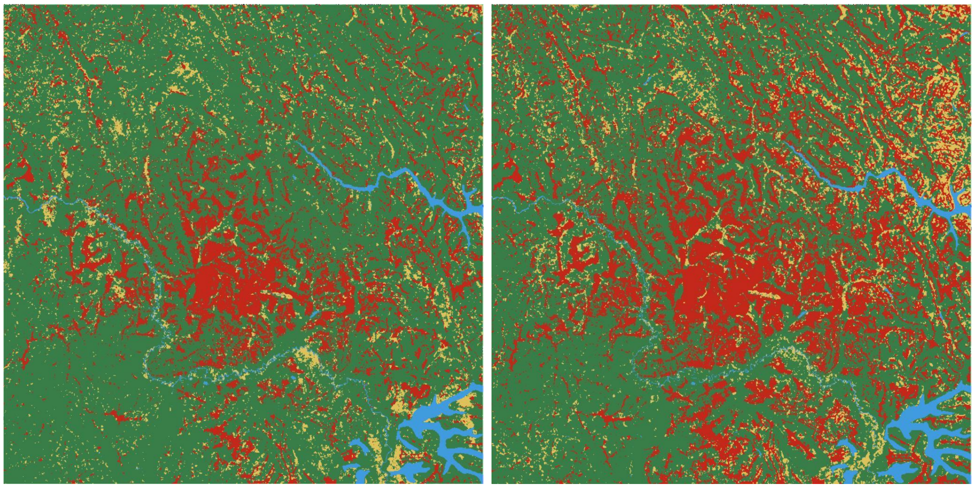


Figure 3: Figure on left is 2016 and on right 2019 Dynamic world images of AOI (Kigali) using 4 classes

If you look at the image you can see that the low vegetation cover/class is very low.
Coming to NICFI satellite image it is of 5m resolution, which is not a high resolution satellite image, which makes it difficult for ground truth labeling of land cover. Hence, we are going ahead with 3 classes in total for our analysis and project. The three classes are tree canopy, water and impervious/built area.

The whole AOI region covering 3890 km2 has been split into 256 images covering roughly 15.2 km2 square area each. We have analyzed the class percentage for each class for total AOI and across all images.

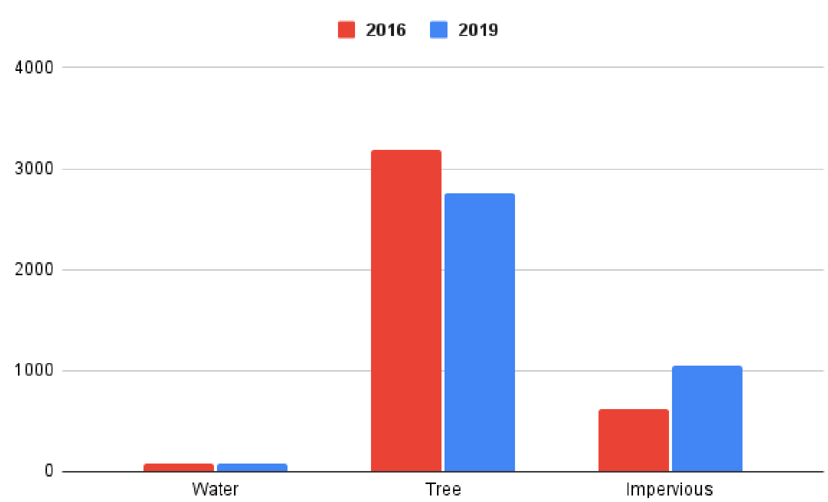


Figure 4: Land cover (km2) by each class for total AOI

Figure 4 shows us how the overall land cover has changed over time. We can see that tree cover of 3188.38 km2 in 2016 has reduced to 2758.16 km2 in 2019. Impervious land has increased from 619.62 km2 in 2016 to 1051.21 km2 in 2019. Water has shown slight decrease from 82.31 km2 to 80.94 km2.

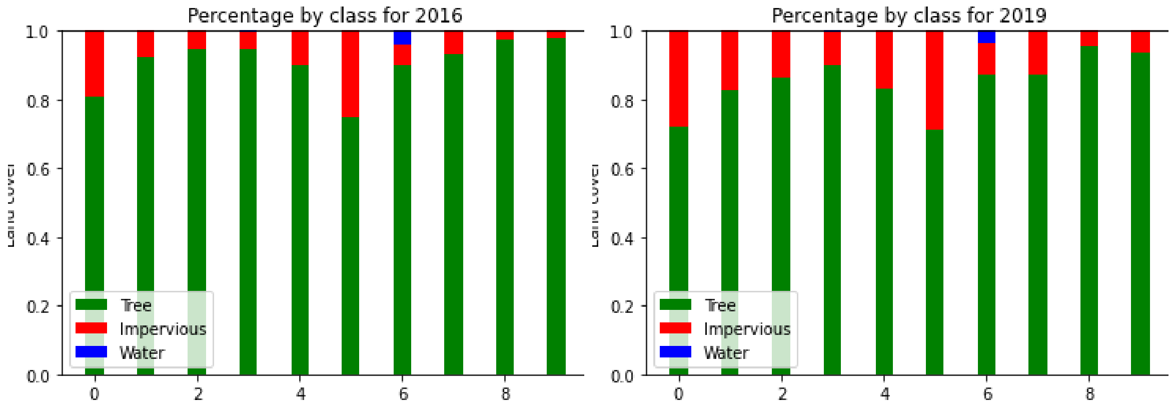


Figure 5: Class percentage per image

In figure 5 we see how the percentage of each class has changed in the first 10 images. Tree percentage has decreased and percentage of impervious has increased which is the expected result since forests are converted into buildings with increasing population

Above we have seen the percentage change for the total area of interest and for the first 10 images. For this project we want to perform a more detailed analysis for each image (smaller regions of interest with 16 km2) and show how the land cover has changed spatially.

Land Cover Loss / Gain:

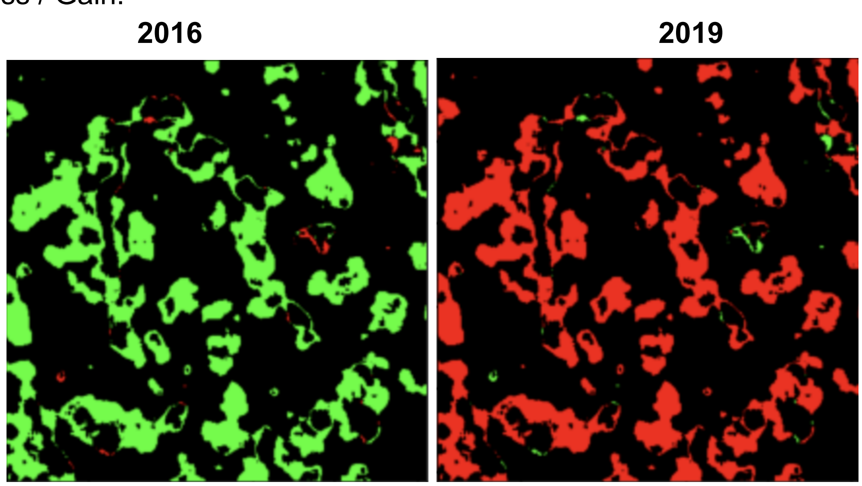


Figure 6: Example of Land Cover Loss/ Gain

Here, the green color in the bottom left image shows that Trees are lost in those pixels whereas Red in the bottom right image shows that corresponding tree loss led to gain in Impervious surface. Black indicates that there is no change in the land cover.

3. Methodology

3.1 Pre-processing

Since the images generated have dimensions of 4096x4096, it requires very high computational power to train even a simple Convolutional Neural Network architecture on this. Thus we tried two approaches to handle this:

3.1.1 Compressing Images: Original images of dimensions 4096x4096 are compressed to lower resolutions of either 128x128 or 256x256 for modeling. This reduces the data quality, however, the computational resources required are significantly lower. Compression is done using opencv library using bilinear interpolation approach

3.1.2 Creating Patches: Original images of dimensions 4096x4096 are divided into patches of 1024x1024 i.e. one image is divided into 16 patches. These patches are further compressed to 256x256 for model training. Smaller patches of size 256x256 and 512x512 miss out on the surrounding context, thus have been discarded

In addition to the above steps, model specific pre-processing steps are applied before training the model. Example: Inception model requires pixel values to be normalized with a mean 0.5 and standard deviation 0.2 approximately for all the channels.

3.2 Modeling

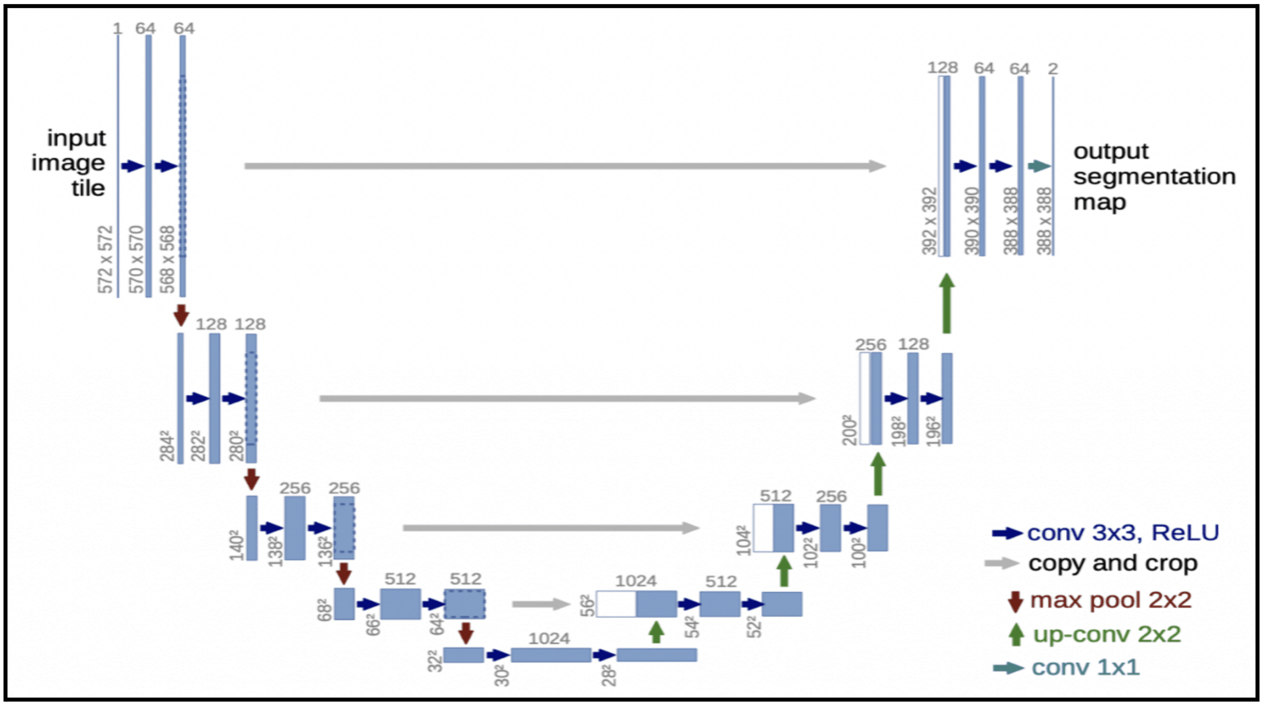
Of the generated 512 images (2016 and 2019 combined), we have used 400 images (200 unique images for both 2016 and 2019) for training the model and 100 images (50 unique images for both 2016 and 2019) for validation. The remaining 12 images were reserved for calculating results using manual validation.

The prediction task here is multi class segmentation of three labels (Trees, Land and Water).

Primarily, two architectures are experimented due to their results on similar studies:

3.2.1 UNET

It has two components, an encoder which takes the input image and creates a low dimensional feature map and a decoder which takes this feature map as input and generates output image of required dimension. The segmentation_models_pytorch library allows us to change the encoder blocks with different state of the art architectures like Inception, ResNet, MobileNet and also control the number of channels in the decoder block.

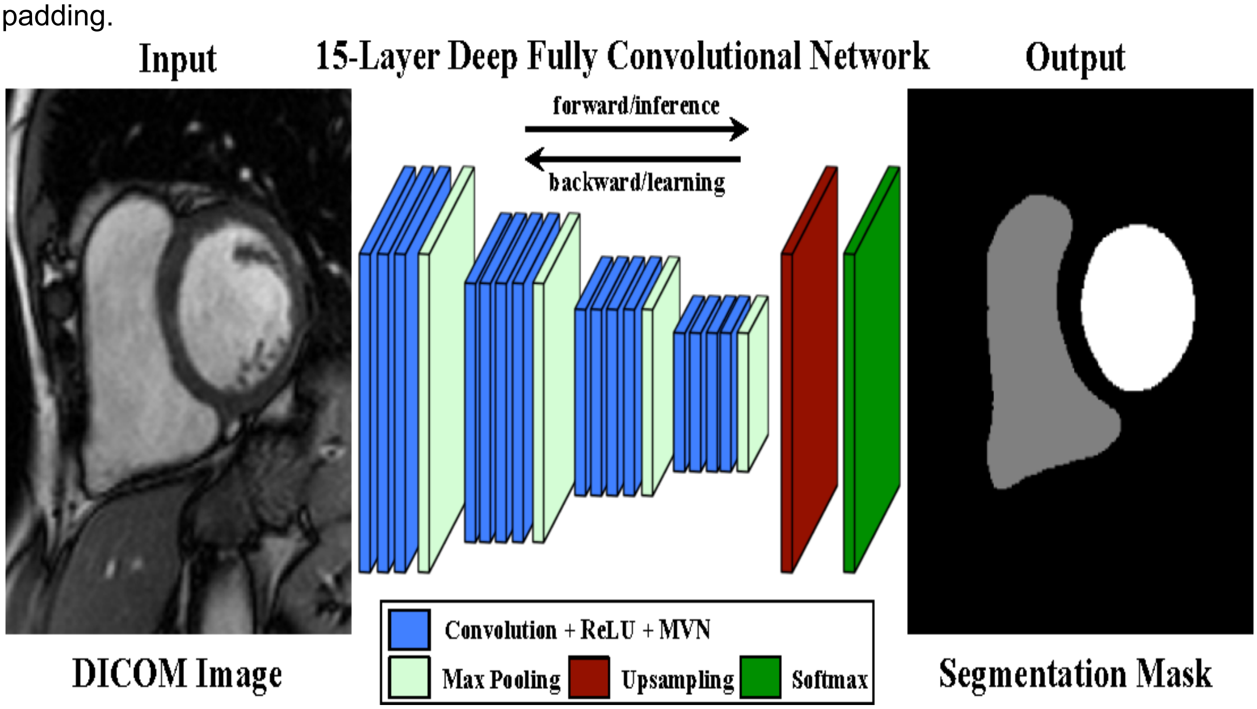


*The image displayed is for reference only. The actual model trained has a different encoder and decoder structure.
Source: https://arxiv.org/abs/1505.04597

Figure 7: UNET architecture

3.2.2 Fully Connected Networks

These are architectures where we use fundamental blocks such as convolutions, max pooling and others. The dimensionality of the output can be engineered to meet the requirements using padding.



*The image displayed is for reference only. The actual model trained has the same dimension maintained from the start using padding.
Source: https://medium.com/stanford-ai-for-healthcare/getting-to-the-heart-of-it-how-deep-learning-is-transforming-cardiac-imaging-22d34bf91a4e

Figure 8: FCN architecture

The above architectures are trained using both compressed images and patches. The best results are achieved when UNET architecture with Inception backbone is trained on patches. Evaluation metrics and comparison of models will be presented in the next section.

3.2.3. Baseline Neural Network

In addition to the above models, a neural network architecture has been trained using pixel values as features to understand the difficulty of the problem statement and evaluate performance of the model when compared to basic models.

Extracted 50,000 random pixels (Features are Red, Green and Blue pixel values) per each class from the 4000x4000 images to generate a training set and to ensure the training class samples are balanced. Built a neural network consisting of 3 Fully Connected Dense layers with 128 neurons with Relu Activation function, 1 Drop out Layer with 20% drop out and final Dense layer of 3 neurons with Softmax activation Function.

Images in the validation set are resized to 50x50 due to computational limitations and output segmentation mask is predicted using individual pixel values (2500 predictions per image).

3.2.4. Handling Class Imbalance

As we saw in the EDA section, there is significant class imbalance in the data, specifically water and land are less represented classes. Thus, training the model becomes very difficult since complex architectures are easily prone to overfitting. To account for the class imbalance, we experimented with adjusting the class weights when defining the loss function which is Categorical Cross Entropy loss. A combination of 0.35 for water, 0.15 for trees and 0.50 for land gave the best results. Also, we experimented by training a separate binary classifier for water and combining the results with the existing model. However, the results are better with the current model. Thus, we did not proceed further with this approach.

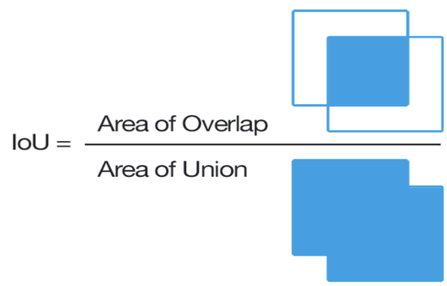
4. Results

4.1 Evaluation Metrics

We have used IoU (Intersection over Union) score to evaluate and compare different models.

IoU score: It is the most widely used evaluation metric for the tasks of Object detection and Image segmentation. The score ranges between 0 and 1, where 1 indicates that the segmentation mask created exactly matches the ground truth and 0 indicates that the predicted mask has no match with the ground truth.

IoU score can be calculated for each class i.e. water, trees and land separately. This allows us to evaluate the model performance on individual classes as opposed to overall score.



Source: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Figure 9: IOU

Here, the original images have a dimension of 4096x4096, however since there is some level of compression used in both the approaches, the images used for evaluation won't be of the same dimension. For example, when we created 1024 patches, we compressed them to 256x256 for model training, thus the output of the model is also 256x256. When all these outputs are joined, the resulting image would be 1024x1024 as opposed to 4096x4096. However, since IoU metric uses area, the results achieved are a good approximation.

To compare the models, IoU score has been calculated on predicted images of validation data. In addition to that, IoU score has also been calculated on Land cover loss and land cover gain as well. The best performing model has been used for manual validation. Also, precision, recall and accuracy are calculated for each pixel and averaged at image level.

4.2 Model Comparison

Following are the results of various models we experimented with. The results are calculated on the validation data i.e. 100 images (2016 and 2019 combined) for segmentation task and 50 images for Land Cover change detection task. For the segmentation task, the average of individual class percentage for validation images (100 in total) is 77.26% for Trees, 21.49% for Land and 2.51% for water.

4.2.1. Neural Network trained using pixel values (Baseline):

Segmentation:

Class	Average Precision	Average Recall	Average IoU
Trees	78.00%	91.00%	72.47%
Impervious	48.0%	11.00%	8.16%
Water	7.00%	28.00%	2.32%
Average Accuracy	73.00%		
IoU Overall	28.23%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	1.00%	6.00%	1.00%	0.00%	20.00%	1.00%
Avg Recall	11.00%	3.00%	28.00%	12.00%	2.00%	7.00%
Avg IoU	0.80%	1.86%	0.43%	0.26%	1.48%	0.62%

4.2.2. UNET with INCEPTIONv2 using 1024 patches (Best Model):

Segmentation:

Class	Average Precision	Average Recall	Average IoU
Trees	95.23%	93.34%	89.23%
Impervious	75.34%	78.34%	62.97%
Water	35.33%	33.67%	29.41%
Average Accuracy	92.09%		
IoU Overall	65.76%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	9.53%	50.74%	7.89%	9.07%	51.14%	8.19%
Avg Recall	12.86%	48.17%	10.44%	12.12%	48.14%	11.54%
Avg IoU	5.29%	32.38%	5.18%	5.88%	32.50%	4.49%

4.2.3. UNET with RESNET34 using 1024 patches:

Segmentation:

Class	Average Precision	Average Recall	Average IoU
Trees	94.41%	86.44%	82.07%
Impervious	62.87%	80.79%	54.48%
Water	15.42%	16.46%	10.36%
Average Accuracy	86.30%		
IoU Overall	51.82%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	2.62%	40.52%	2.07%	1.32%	41.80%	3.48%
Avg Recall	18.01%	33.11%	3.35%	8.91%	33.07%	18.33%
Avg IoU	2.15%	20.78%	1.47%	0.99%	21.22%	2.50%

4.2.4. Fully Connected Network with 1024 patches:

Segmentation:

Class	Average Precision	Average Recall	Average IoU
Trees	86.28%	79.79%	69.94%
Impervious	45.93%	57.62%	31.37%
Water	8.84%	9.78%	5.62%
Average Accuracy	74.57%		
IoU Overall	35.64%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	0.96%	21.56%	1.02%	0.48%	23.90%	0.99%
Avg Recall	24.34%	15.97%	2.12%	5.86%	15.86%	24.57%
Avg IoU	0.92%	9.58%	0.72%	0.33%	9.99%	0.94%

4.2.5. UNET with MOBILENETv2 using Compressed Images (224x224):

Segmentation:

Class	Average Precision	Average Recall	Average IoU
Trees	90.20%	83.21%	76.50%
Impervious	47.78%	67.23%	38.50%
Water	8.07%	0.02%	0.02%
Average Accuracy	81.69%		
IoU Overall	48.35%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	1.85%	25.52%	0.00%	0.00%	25.48%	1.80%
Avg Recall	14.23%	27.00%	0.00%	0.00%	27.18%	16.00%
Avg IoU	1.48%	14.16%	0.00%	0.00%	14.20%	1.46%

4.2.6. UNET with MOBILENETv2 using Compressed Images (128x128):

Segmentation:

Class	Average Precision	Average Recall	Average IoU
Trees	83.90%	95.60%	80.92%
Impervious	64.93%	28.03%	23.42%
Water	17.00%	2.08%	2.04%
Average Accuracy	83.55%		
IoU Overall	44.45%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	1.73%	30.55%	6.11%	0.21%	32.10%	1.80%
Avg Recall	8.69%	11.07%	0.71%	0.23%	11.22%	10.87%
Avg IoU	0.92%	8.32%	0.65%	0.07%	8.47%	0.98%

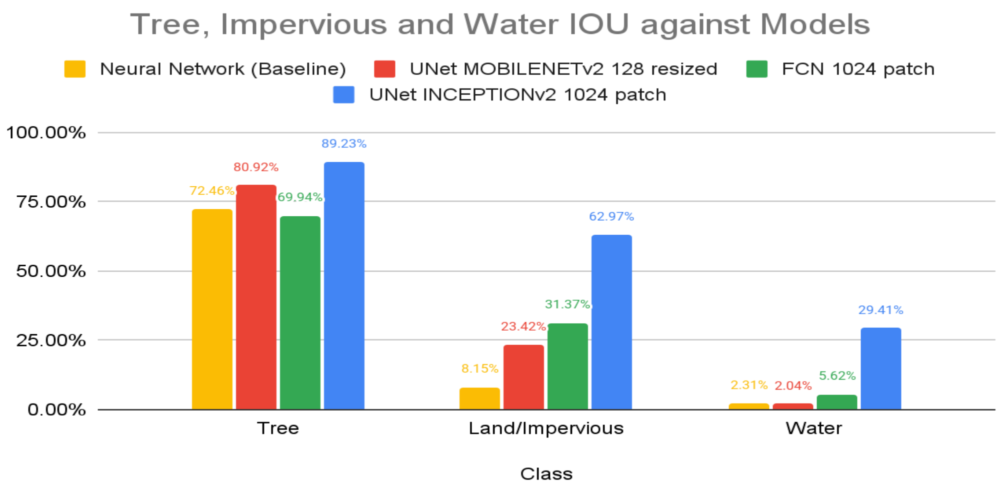


Figure 10: IoU scores of models experimented with different approaches

4.3 Visualizing Predictions

Sample Predictions of Segmentation task by the best performing model:

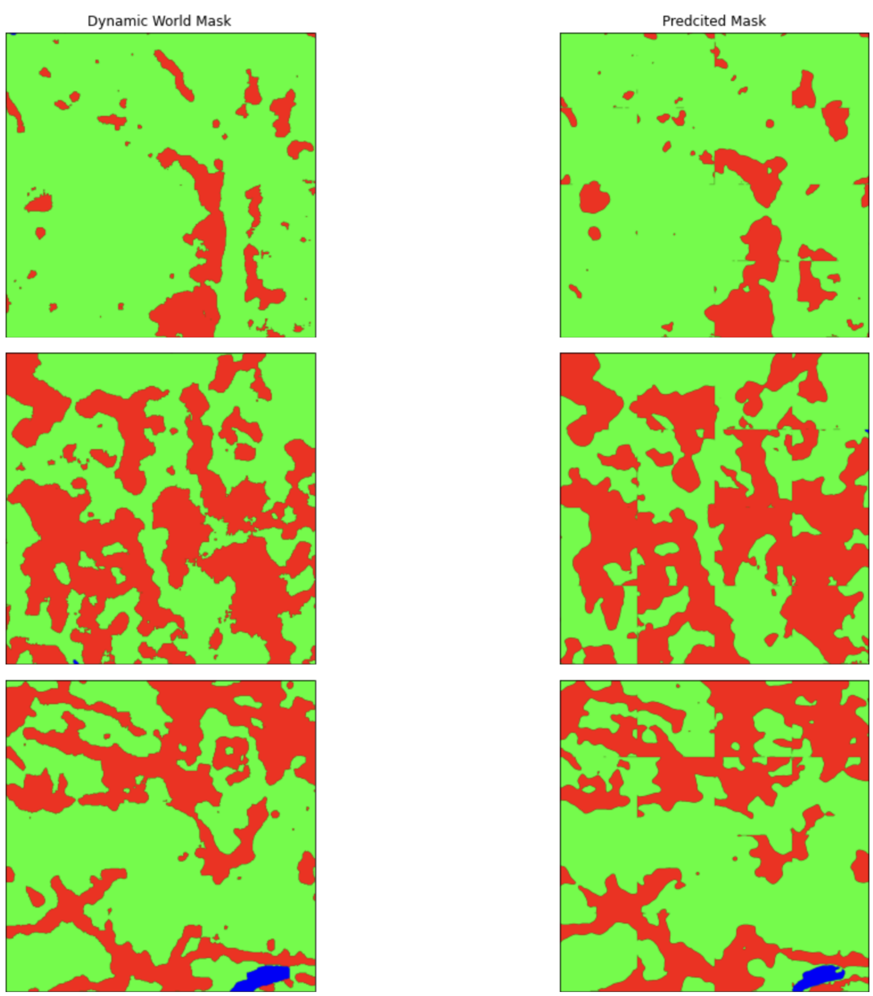
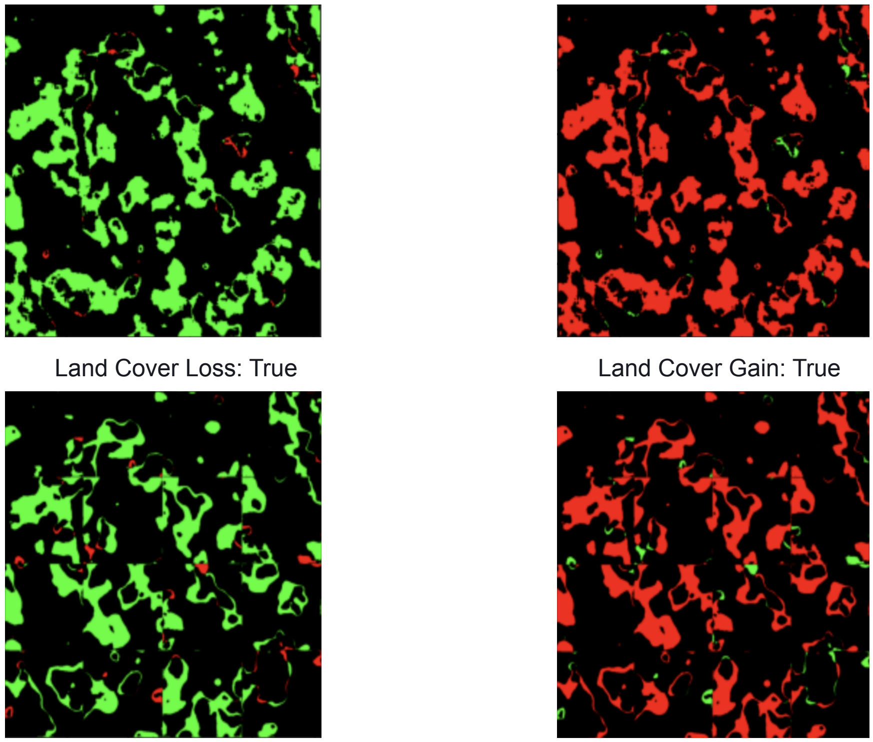


Figure 11: Predicted semantic segmentation images

Sample Predictions of Land Cover change by the best performing model:



Land Cover Loss: True   Land Cover Gain: True

Land Cover Loss: Predicted   Land Cover Gain: Predicted

Figure 12: Predicted Land Cover change images

4.4 Check for Overfitting

In order to ensure that the best model i.e. Unet with Inception backbone trained on 1024 patches is not overfitting, we have calculated the results on training data i.e. 400 images (2016 and 2019 combined) for segmentation task. Following are results:

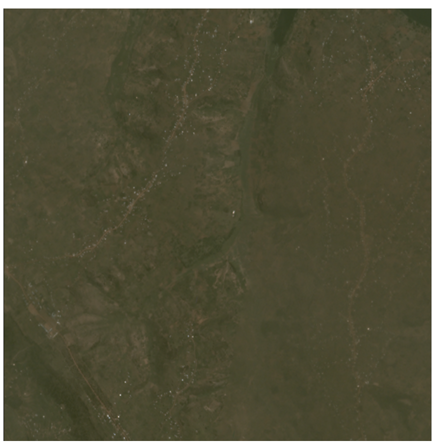
Class	**Average Precision**		**Average Recall**		**Average IoU**	
	Train	**Val**	**Train**	**Val**	**Train**	**Val**
Trees	96.34%	95.23%	93.91%	93.34%	90.74%	89.23%
Impervious	77.66%	75.34%	83.46%	78.34%	68.39%	62.97%
Water	42.87%	35.33%	41.28%	33.67%	36.34%	29.41%
	Train			**Val**		
Average Accuracy	93.41%			92.09%		
IoU Overall	70.52%			65.76%		

As we can see, the difference between train metrics and Val metrics is not very significant. Thus, we can be sure that the model is not overfitting.

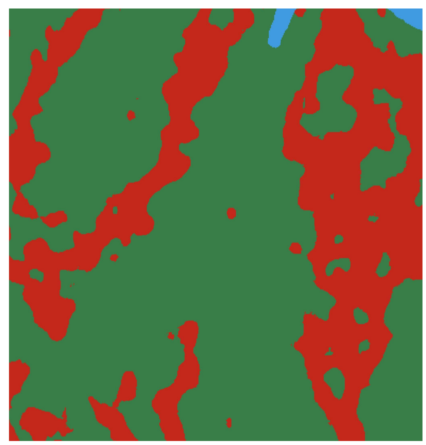
4.5 Manual Validation

In addition to the results in the above section, predictions from this model have been used for manual validation as well.

Manual Validation: Inorder to make sure that predicted images are inline with the ground truth, we manually annotated 14 images (7 unique images of 2016 and 2019) using a software called Groundwork. The images are then exported to create segmentation masks for evaluation. One major challenge we faced was with the quality of the images. Since NICFI images are 5m resolution, we could not accurately distinguish between buildings and barren lands with shrubs. That led to a significant difference between the classes in Dynamic world masks (training masks) and Ground truth masks. This led to the poor results when model predictions are compared with Ground truth masks as expected.



Ground Truth Mask



Training Mask

Following are the results when **Dynamic world labels** are compared against **Manual labels:**

Segmentation:

Class	Average Precision	Average Recall	IoU
Trees	82.65%	89.99%	75.97%
Impervious	32.17%	20.62%	14.61%
Water	15.38%	13.35%	8.56%
Average Accuracy	77.24%		
IoU Overall	40.74%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	0.96%	16.88%	1.71%	8.40%	36.14%	2.15%
Avg Recall	5.36%	10.92%	4.33%	5.75%	11.40%	6.08%
Avg IoU	0.72%	6.82%	1.27%	3.80%	8.98%	0.94%

Following are the results when **Model predictions** are compared against **Manual labels**:

Segmentation:

Class	Average Precision	Average Recall	IoU
Trees	81.37%	90.22%	75.07%
Impervious	35.03%	20.44%	15.03%
Water	23.61%	16.45%	11.96%
Average Accuracy	76.51%		
IoU Overall	41.45%		

Land Cover Change:

Metric	+Tree	-Tree	+Water	-Water	+Land	-Land
Avg Precision	1.32%	24.84%	4.90%	58.13%	52.36%	3.84%
Avg Recall	12.09%	14.01%	6.83%	10.17%	14.12%	12.68%
Avg IoU	1.17%	9.53%	2.01%	5.57%	11.89%	2.54%

5. Conclusion

The developed model has the ability to segment and detect land cover change better than random guessing and baseline model. As we can see from the above section, UNET with Inception v2 trained on 1024 patches is the best performing model with an overall IOU score of 66% which is 37% better than the baseline Neural Network model. The model can be leveraged to get an estimate of important metrics like deforestation rate over time.The model given the 5m resolution image is able to perform segmentation of the images with good accuracy, coming to the land cover change the model needs more detailed/high quality images to capture even the small changes, which comes under the future scope.

5.1 Qualitative Result

For each of the models we can clearly see that the drop seen in tree cover/class from 2016 to 2019 can be attributed to the growth in built/impervious cover. Through this we say that the model clearly depicts the initial hypothesis of a potential deforestation.

5.2 Future Scope

This project works as a starting point for working with segmentation models as an indicator for deforestation. However, the model's performance can be improved with the help of high resolution images. The use of more complex architectural models can be employed to implement image segmentation more precisely. The model has potential to be expanded further to incorporate more than 3 class labels provided we have high resolution data.

5.3 Challenges

The development of this project involved numerous challenges right from data collection to ML model development.

- Creating the required dataset of satellite images and dynamic world labels was highly time-consuming. We developed a JavaScript code to extract the required dataset and deciding on optimal coordinates and boundaries for image creation was demanding.
- The search for the right tool for implementing manual labeling was time-intensive. We explored different tools for manual labeling like QGIS, Labelbox, Label Studio, and Microsoft Land Cover, but each had its drawbacks.
- A significant challenge lies in the fact that NICFI images are 5m in resolution. So, we couldn't label images precisely which led to poor results in manual validation
- The computation time and power was the biggest challenge in the execution of this project. We had to make patches or compressed images since the original dataset was very highly dimensional (4096 x 4096).

Ethical considerations: There are no ethical considerations. All the data is openly available and is being used for academic and research purposes. The resources used have been cited accordingly.

6. References

[1] High-resolution land cover change from low-resolution labels: Simple baselines for the 2021 IEEE GRSS Data Fusion Contest https://arxiv.org/pdf/2101.01154.pdf

[2] Planet Team (2017). Planet Application Program Interface: In Space for Life on Earth. San Francisco, CA. https://api.planet.com

[3] Brown, C.F., Brumby, S.P., Guzder-Williams, B. et al. Dynamic World, Near real-time global 10 m land use land cover mapping. Sci Data 9, 251 (2022). doi:10.1038/s41597-022-01307-4

[4] 2020 IEEE GRSS Data Fusion Contest, 2020. http://www.grss-ieee.org/community/technical-committees/data-fusion/2020-ieee-grss-data-fusion-contest/.

[5] 2021 IEEE GRSS Data Fusion Contest Track MSD, 2021. http://www.grss-ieee.org/community/technical-committees/data-fusion/2021-ieee-grss-data-fusion-contest-track-msd/.

[6] Jonathan Long, Evan Shelhamer and Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation", Proc. Of IEEE conference on computer vision and pattern recognition, 2015.

[7] Tensorflow Image segmentation tutorial https://www.tensorflow.org/tutorials/images/segmentation

[8] Image-to-Image Translation with Conditional Adversarial Networks https://arxiv.org/abs/1611.07004

[9] Microsoft Land cover mapping project https://github.com/microsoft/landcover

[10] Google Earth Engine https://earthengine.google.com/

[11] Groundwork for manual labeling https://groundwork.azavea.com

[12] Segmentation Models library Pytorch https://segmentation-modelspytorch.readthedocs.io/en/latest/

[13] Semantic Segmentation using smp library tutorial https://github.com/qubvel/segmentation_models.pytorch/blob/master/examples/cars%20segmentation%20(camvid).ipynb

[14] U-Net: Convolutional Networks for Biomedical Image Segmentation https://arxiv.org/abs/1505.04597

[15] Project Repo: https://github.com/nishimodi2799/capstone_project

7. Individual Contributions

7.1 Vishwas Reddy Thuniki
Collaborated with other teammates to extract satellite images from GEE using Javascript. Trained UNET model using compressed images presented in the first Progress report. Created a Pytorch based code to create patches and load images in batches to train multiple architectures of both FCN and UNET with different backbones, of which Inception backbone gave the best results. Calculated the reported metrics for both segmentation task and Land cover change tasks for all models and participated in preparation of required submissions.

7.2 Binny Manojkumar Naik
Worked on extracting 512 NICFI satellite images and 512 Dynamic World Labels from GEE using JavaScript. Labeled satellite images and developed a python code to convert the Geojson files generated by all team members into PNG/Numpy arrays. Explored and implemented FCN model with vgg16, vgg11, and resnet backbones by training model on patches and full images. Calculated the results for the same. However, the models mentioned in this report outperformed. Collaborated with everyone in all the required submissions.

7.3 Nishi Amish Modi
Worked on conversion of JavaScript code to Python code for extraction of NICFI images from Google Earth Engine. Explored various image labeling tools for manual labeling of data. Tried to implement HRNet and Segnet model for generating predictions for land cover change detection. Collaborated with Prashanth to understand and implement the GitHub repository suggested by our mentor. Worked on the implementation of FCN model with vgg16, vgg11, resent and Mobilenet as their backbone. Created a few codes for preprocessing data and calculating the statistics required for the implementation of these models.

7.4 Gangadhara Reddy Velagala
Initially worked on exploring various labeling tools such as Microsoft Land Cover and QGIS.In QGIS, Model resulted in the errors during the model training. Hence moved with Groundwork for manual labeling of images. Ideated the use of neural network architecture for the task of image segmentation. Engineered a way to extract and use pixel values as features to train the neural network model. Also, used compression technique as an approximation to validate the images. Manually labeled a few images to test the model for manual validation. Calculated the required metrics for the model and participated in preparation of all required submissions.

7.5 Sai Prashanth Pathi
Worked on fast retrieval of data from GEE using python script but had a bottleneck with image quality. Explored Dynamic World and Skysat satellite images. Identified the land classes need to be restricted to 3 i.e tree, impervious and water, and performed extensive exploratory data analysis. Manually labeled 2 images using groundworks. Modified and implemented the repository present in [1] for this project. The new repo has UNET and FCN models along with the capabilities of processing images using patches of provided size and generates semantic segmented images for NICFI images using Dynamic world labels. Repo: https://github.com/sp3915/dfc2021-msd-baseline