Semantic Segmentation of Satellite Imagery

Nancy Nigam
Computer Science
CIMS, NYU
New York City, US
nn2163@nyu.edu

Jorge Roldan
Computer Science
CIMS, NYU
New York City, US
jlr9718@nyu.edu

Aditya Upadhyaya
Computer Science
CIMS, NYU
New York City, US
au2056@nyu.edu

***Abstract*—The ability to automatically classify the class at the pixel level of satellite imagery has a wide range of applications including monitoring, managing, and detecting changes of land cover. In this work, we used different versions of the ResNet model for classification tasks, and U-Net for the segmentation tasks. We explored the effect of three different losses, namely, Focal, Dice, and Cross entropy together with the mentioned models to find out the ideal configuration.**

We trained and tested our models on three datasets: LandCoverNet, Crop Type In Ghana, and Landcover.ai. Due to challenges such as cloud cover and low-resolution images, we decided to mostly focus and tune our models using the Landcover.ai dataset. The mean IoU scores for different models on the Landcover.ai datasets we obtained are 0.884 for U-Net with Resnet34, 0.871 for U-Net with ResNet18, 0.887 for U-Net with ResNet101, and finally 0.886 for U-Net++ with ResNet34. We also observe that Focal loss performs significantly better than Dice loss when we have a skewed dataset on our hands.

Github: semantic-segmentation-for-satellite-imagery

I. Introduction

The evolution of remote sensing technologies over the last couple of decades has drastically increase the amount of satellite imagery datasets available and open to the public. The increase of data available plus the advances in the fields of Machine Learning and Computer Vision to study and gain insights from these datasets has open the door to opportunities to tackle new type of problems at scale.

One of these opportunities is the ability to manage and survey the properties of different land covers all around the world by automating the identification of terrains, natural or artificial structures, or any other object of interest using semantic segmentation models. These models receive as an input a source image and a mask where each pixel indicates the class of that pixel in the source image. Once the model is trained, it will ideally be able to predict a mask for a new, unseen image.

Automatic classification at the pixel level of satellite imagery can then play a key role at monitoring, managing, and detecting changes on land covers. This is directly related to the Life on Land objective from the United Nation's Sustainable development goals [9], which deals with protecting, restoring and promoting sustainable use of terrestrial ecosystems. Leveraging Machine Learning models for semantic segmentation tasks on these large satellite imagery datasets is a promising tool for successfully accomplishing this sustainability goal.

II. Related Work

The application of semantic segmentation models on satellite imagery have been shown to perform well in many cases [19][13][16][3][18][4][2][12][7]. Authors in [19] trained a ResNet model architecture for classification on the BigEarthNet dataset [17]. They also trained a modified U-Net architecture for the segmentation tasks using a customized dataset, which combined a CORINE Land Cover map as well as Sentinel-2 source images. Using these combinations of architectures and datasets, they obtained an overall F1 score of 0.749 for the classifier with 43 possible classes, as well as a high IoU score for the segmentation model.

Authors in other works such as [13] used a pretrained ResNet50 and transfer it into a U-Res-Net for classification tasks. They also used a customized version of U-Net for the segmentation tasks. A common technique in these works is to use data augmentation to overcome the challenge of small size of the satellite imagery datasets currently available.

Other interesting applications of semantic segmentation models is to detect changes in different land covers terrains over time as done in [16]. In this work, the authors use Deeplab v3+ [6] and build a complete training pipeline to create a land cover change detection system achievin a mean IoU of 0.756.

III. Datasets

A. LandCoverNet Dataset

1) Dataset Characteristics: The LandCoverNet dataset consists of 1980 image chips with a resolution of (256px x 256px) where each image chip correspond to a specific location across Africa. For each image chip there are several source images at different time stamps during 2018 obtained from the Sentinel-2 Surface reflectance product (L2A). Furthermore, each image has a corresponding mask with the labels for each pixel as shown in table I [1].

2) Data collection, preprocessing, and challenges: The LandCoverNet dataset [1] was collected using the Radiant MLHub API, [11]. Since this API is in the early stages of development, we faced many issues while collecting the data. The primary issue was that we couldn't filter out source images with clouds on it, therefore a significant portion of the dataset had source images with clouds.

The format of the sources and mask images was TIF, therefore we had to convert all of these images into PNG

TABLE I
LANDCOVERNET DATASET

LandCoverNet Dataset Classes		
Class Name	**Pixel Value**	**RGB Value**
Unknown	0	(0,0,0)
Water	1	(0,0,255)
Artificial	2	(136,136,136)
Natural	3	(209,164,109)
Snow/ice	4	(245,245,255)
Woody	5	(214,76,43)
Cultivated	7	(24, 104, 24)
(Semi) Natural	7	(0, 255, 0)

by stacking the Red Blue, and Green channels. The masks also had initially the TIF format with one channel, where the value of each pixel represents the class from table I. Finally, we used the RGB mapping to convert each mask to an RGB PNG image. The scripts we created to do this are provided in the data_preprocessing folder in the source code.

B. Semantic Segmentation of Crop Type in Ghana Dataset

1) Dataset Characteristics: The Crop Type in Ghana dataset [15] consists of images from Sentinel-1 and sentinel-2 from Ghana and South Sudan taken over 2016 and 2017. For each image chip, there are several source images at different time stamps, and a corresponding mask. There are 25 labels for different type of crops as shown in table II.

TABLE II
GHANA DATASET

Crop Type in Ghana Dataset Classes		
Class Name	**Pixel Value**	**RGB Value**
Unknown	0	(0, 0, 0)
Ground nut	1	(80,0,165)
...	...	...
Nyenabe	23	(193,0,76)
Pepper	24	(204,255,153)

2) Data collection, preprocessing, and challenges: The Ghana dataset was collected using the Radiant MLHub API [11]. One major advantage of this dataset is that it provides a cloud mask for each source image, which was not the case for the LandCoverNet dataset. We used this cloud mask in order to filter out any images that had clouds on it. We also converted all the source and masks images from TIF to PNG format. We have included these scripts in the data_preprocessing folder in the source code. A major drawback of this dataset is that the resolution of the images is just (64px x 64px), which is rather low to successfully train models for semantic segmentation.

C. LandCover.ai Dataset

1) Dataset Characteristics: The LandCover.ai dataset consists of satellite imagery (Fig. 1) for semantic segmentation applications covering a region of Poland of a total area of 216.2 km^2 [5]. The original dataset consists of 41 orthophoto tiles of 5 km^2 where 33 images and 8 images have a resolution of (9000 px x 9500 px) and (4200 px x 4700 px), respectively.

Fig. 1. LandCover.ai : Sample Images

The authors of this dataset [5] provided a script which we used to split these 41 images and their respective masks into 10,674 source and 10,674 masks images of (512 px x 512 px) resolution. The land cover of these images consist of agricultural areas (60 %) and mixed forests (29.6 %).

2) Classes and RGB Mapping: The name of the classes, their respective pixel value, and RGB value are shown in table IV. The annotations were made manually by the authors using VGG Image Annotator (VIA) [8]. According to [5], there are 12280 buildings (1.85 km^2), 72.02 km^2 of woodland, 13.15 km^2 of water, 3.5 km^2 of roads, and 125.75 km^2 of background, that is, the unknown class.

TABLE III
LANDCOVER.AI DATASET

Landcover.ai Dataset Classes			
Class Name	**Pixel Value**	**RGB Value**	**Color**
Unknown	0	(0,0,0)	
Building	1	(80,0,165)	
Woodland	2	(255,204,0)	
Water	3	(0,244,244)	
Road	4	(105,105,105)	

3) Data collection, preprocessing, and challenges: The source images and masks are LandCover.ai were originally provided in PNG and JPG format, respectively. The masks consisted of one channel where each pixel had the value of the class (0 - 4) as shown in table IV. In order to be able to use them in our model, we mapped each pixel value to the corresponding RGB value and generated a new PNG mask that can directly be used when training our model. This python script can be found in the data_preprocessing folder in our source code.

Two major advantages of the Landcover.ai dataset, compared the LandCoverNet and Crop Type in Ghana dataset is that the source images in Landcover.ai did not contain clouds since the authors [5] filtered out any of these images. Fur-

thermore, the high-resolution and quality of the images makes this dataset the ideal one to train our semantic segmentation models.

IV. Methodology

A. Architectures

For the purpose of our experiments, we leverage a U-Net architecture [14]. U-Net is a fully convolution neural network used extensively for image segmentation. It involves encoder and decoder components connected with skip connections. For the classification model that serves as our encoder and extracts features of different spatial resolution, we use a ResNet architecture [10]. As part of our ablation study, we also review U-Net++ [21] which is essentially a nested architecture consists of convolution layers on skip pathways.

B. Evaluation Metrics

The lead evaluation metric for us is the Jaccard Index, also known as Intersection over Union (IoU).

$$IoU = \frac{target \cap prediction}{target \cup prediction}$$

In addition to IoU, we also capture Pixel Accuracy, Precision, and Recall. High pixel accuracy (the number of pixels classified correctly) doesn't always imply good segmentation results, especially in cases of class imbalance.

V. Setup and Experiments

A. Setup

Our entire data augmentation, training, and prediction pipeline is implemented on the PyTorch framework and we use the NVIDIA V100 GPU to speed up our computations. The SMP library [20] provides us segmentation models with pre-trained backbones.

Typically, neural network initialized with weights from a pre-trained network shows better performance and for our purposes, we initialize our ResNet encoder with ImageNet weights. The encoder depth is kept at a constant value of 5 throughout the scope of our experiments.

B. Hyperparameters

Our seed dataset is divided into train-validation-test with a ratio of 80:10:10. A batch size of 32 works best for our use-case and we keep this value constant throughout. We start off with SGD, but the Adam optimizer soon becomes our choice for all our experiments as it helps us converge faster and we apply a softmax activation after the final convolution layer. A bit of fine-tuning helps us arrive at a starting learning rate of 1e-4 for our StepLR with a multiplicative factor of 0.1 every 15 epochs. The step size varies when it comes to training with different encoders. Finally, the choice for our loss function is the Focal Loss as it helps us address class imbalance. We do consider other loss functions (Cross Entropy Loss and Dice Loss) and we discuss this in detail in the next section.

C. Experiments

We conduct a number of experiments across multiple configurations of our pipeline and we discuss them in detail.

1) Dataset Challenges: Datasets in their raw form pose a serious challenge due to a myriad of factors discussed in the previous section. Training results on the LandCoverNet and Ghana Dataset achieves a Pixel Accuracy in the lower range of 80s, but that is largely due to all pixels being classified as 'unlabeled'/'unknown' which happens to be the majority class. The mean IoU values fair poorly and tend to oscillate in the range of 0.3-0.4. Owing to these factors, the following set of experiments focus solely on the LandCover.ai dataset.

2) Loss Functions: The choice of loss function plays a major role for any architecture as it dictates our learning process. In an effort to gain a deeper understanding of the impact of our choices, we do a comparative analysis of three popular loss functions - Focal, Dice, and Cross Entropy. To keep all other parameters constant, We plug a ResNet18 encoder (with pre-trained ImageNet weights) in our U-Net. Training is done for a total of 40 epochs. We keep our initial dataset as the baseline (Fig 6) and capture the results when we intentionally skew the dataset heavily towards one of the classes (woodland) (Fig 7). Fig 1 depicts the progression of IoU over the iterations when the dataset isn't imbalanced.

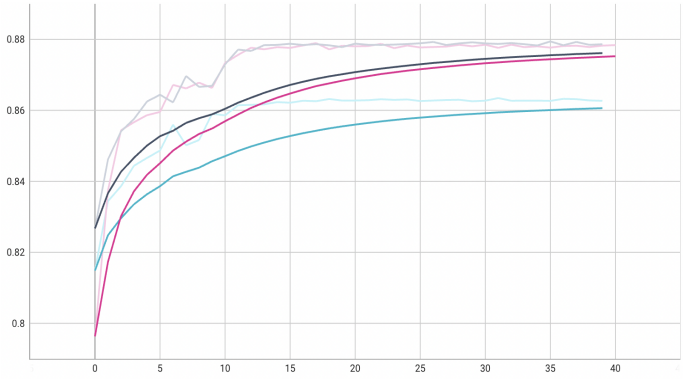

Fig. 2. LandCover.ai : Progression of IoU Scores (y-axis) over 40 epochs (x-axis) for Focal Loss (Magenta), Dice Loss (Teal), and Cross Entropy Loss (Gray) on a U-Net with a ResNet18 encoder pre-trained with ImageNet weights. The dataset doesn't have a significant imbalance.

3) Network and Encoders: With an intent to observe the effect of the depth of network, we try multiple encoders (ResNet18, ResNet34, and ResNet101) for our U-Net. Additionally, we evaluate a U-Net++ network based on a ResNet34 encoder. All encoders are initialised with ImageNet weights. We train for a total of 40 epochs and employ Focal Loss as our loss function. The visualization results are shown in Fig 8. Fig 3,4, and 5 depict the progression of IoU scores, Focal Loss values, and F Scores respectively.

VI. Results

1) The presence of cloud cover and/or low-resolution images results in poor results. Additional pre-processing steps are required before one can expect satisfactory

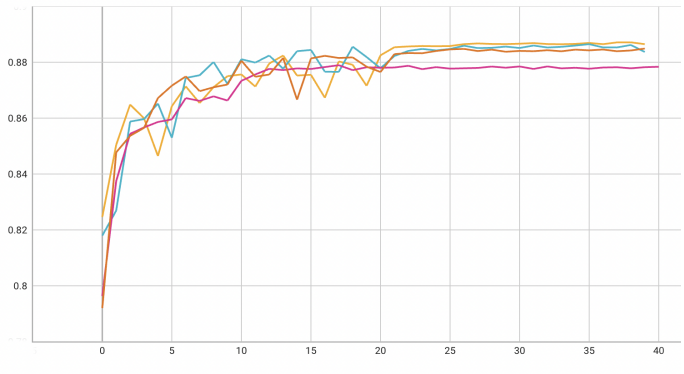

Fig. 3. LandCover.ai : Progression of IoU Scores (y-axis) over 40 epochs (x-axis) for U-Net ResNet18 (Pink), U-Net ResNet34 (Orange), U-Net ResNet101 (Blue), and U-Net++ ResNet34 (Yellow). Focal Loss is being employed and all encoders are initialized with pre-trained Image-Net weights.

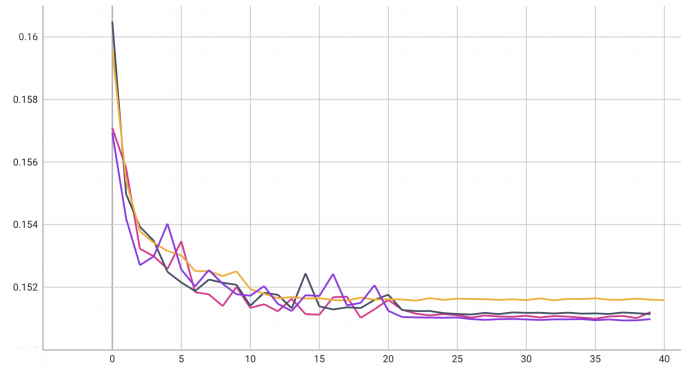

Fig. 4. LandCover.ai : Progression of Focal Loss values (y-axis) over 40 epochs (x-axis) for U-Net ResNet18 (Yellow), U-Net ResNet34 (Dark Gray), U-Net ResNet101 (Pink), and U-Net++ ResNet34 (Purple).

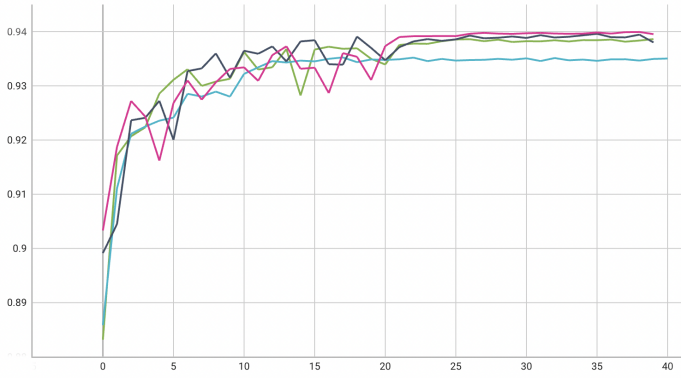

Fig. 5. LandCover.ai : Progression of F Scores (y-axis) over 40 epochs (x-axis) for U-Net ResNet18 (Blue), U-Net ResNet34 (Green), U-Net ResNet101 (Gray), and U-Net++ ResNet34 (Pink).

results. As a consequence, all our results are based off the LandCover.ai dataset.

2) Roads and Buildings pose a challenge owing to their minority representation and narrowness (in case of roads). A lot of samples also present with scenarios where the road network is hidden under a canopy of trees.
3) Dice Loss consistently misses out on the minority classes in cases of severe imbalance. This is shown best in Fig 7 where it fails to detect Roads.
4) Focal Loss performs marginally better (mean IoU = 0.875 on validation set) than both Dice and Cross Entropy when the dataset is slightly balanced. However, the difference is stark in cases of severe imbalance.
5) When it comes to the number of layers in our encoder, we observe that depth matters and ResNet18 consistently achieves a 1 point lower mean IoU score (0.871) on the validation set when compared to the other encoders over the course of 40 epochs.
6) ResNet101 on a U-Net network achieves a mean IoU score of 0.8871 on the validation set and a visual inspection of the results hint towards it being able to detect shapes concretely. The combination of U-Net++ and ResNet34 performs only slightly better (mean IoU = 0.886) than that U-Net and ResNet34 (mean IoU = 0.8843). The final numbers are shown in Table IV.

TABLE IV
MEAN IOU SCORES FOR DIFFERENT MODELS

U-Net with ResNet34 (Baseline)	0.884
U-Net with ResNet18	0.871
U-Net with ResNet101	0.887
U-Net++ with ResNet34	0.886

VII. CONCLUSION

In this study, we present a data processing and training pipeline that can be used to automate Satellite Imagery segmentation. Semantic Segmentation has a myriad of applications ranging from disaster resilience to tracking regional urbanization, environmental monitoring, and the Sentinel-2 mission has made this data readily available. We discuss the challenges that one can face while working on a new dataset, ways to overcome them, and create reliable images and masks for training. We then conduct a series of experiments to gauge the importance of choosing a loss function. Loss functions are a critical component of any learning-based approach and one of the factors that should govern this choice is the extent of class imbalance. We also observe the effects of the depth of an encoder by exploring different flavors of U-Net. Finally, the best performing model is the U-Net with a backbone of ResNet101 achieving an mIoU score of 0.886. In the future, we can extend this study to track changes in a particular landscape over a period of time.

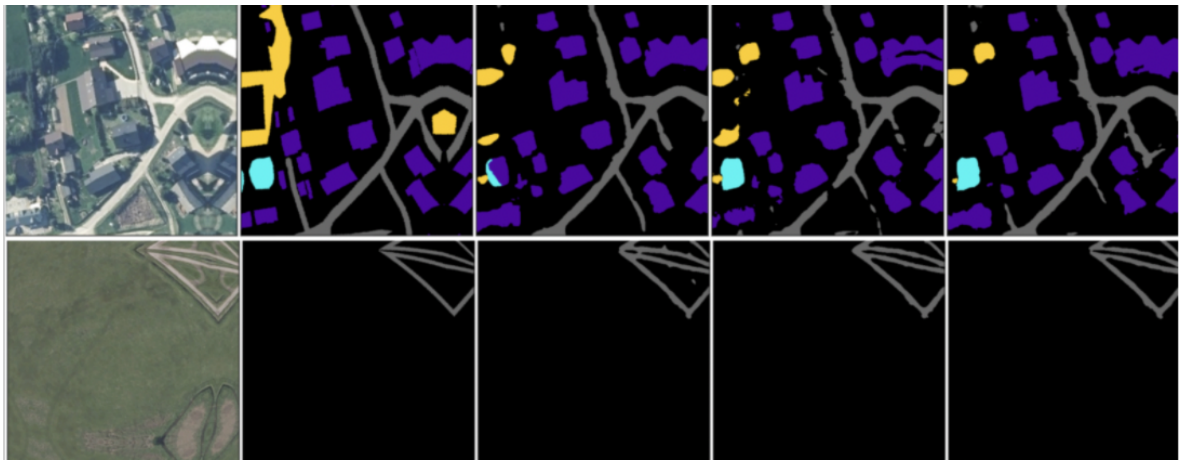

Fig. 6. LandCover.ai : Prediction on a slightly balanced dataset with a U-Net on a ResNet18 encoder. This serves as our baseline. The first column contains the actual images, the second column represents the target mask, the third, fourth, and fifth columns represent the Predicted Mask with Dice, Focal, and Cross Entropy Loss respectively.

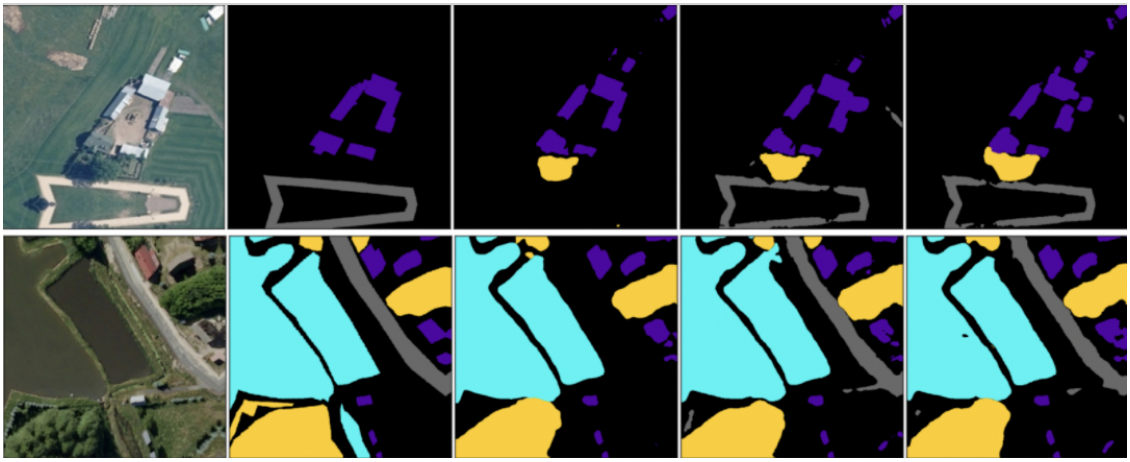

Fig. 7. LandCover.ai : Prediction on a highly-imbalanced dataset with a U-Net on a ResNet18 encoder. The first column contains the actual images, the second column represents the target mask, the third, fourth, and fifth columns represent the predicted mask with Dice, Focal, and Cross Entropy Loss respectively. Dice Loss(third columns) consistently fails to detect the minority class (roads).

Fig. 8. LandCover.ai : Comparing the effect of using different encoders and/or network architecture. From left to right : Actual Image (first column), Target Mask (second column), Predicted Mask by U-Net ResNet18 (third column), Predicted Mask by U-Net ResNet34 (fourth column), Predicted Mask by U-Net ResNet101 (fifth column), Predicted Mask by U-Net++ ResNet34 (last column). For the second row, the actual image poses a challenge for the model with a hidden road network and ResNet101 fares better than the others.

References

[1] Hamed Alemohammad and Kevin Booth. "LandCoverNet: A global benchmark land cover classification training dataset". In: *CoRR* abs/2012.03111 (2020). arXiv: 2012.03111. URL: https://arxiv.org/abs/2012.03111.

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Beyond RGB: Very High Resolution Urban Remote Sensing With Multimodal Deep Networks". In: *CoRR* abs/1711.08681 (2017). arXiv: 1711.08681. URL: http://arxiv.org/abs/1711.08681.

[3] R. Avenash and Prashanth Viswanath. "Semantic Segmentation of Satellite Images using a Modified CNN with Hard-Swish Activation Function". In: *VISIGRAPP*. 2019.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *CoRR* abs/1511.00561 (2015). arXiv: 1511.00561. URL: http://arxiv.org/abs/1511.00561.

[5] Adrian Boguszewski et al. "LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands and Water from Aerial Imagery". In: *CoRR* abs/2005.02264 (2020). arXiv: 2005.02264. URL: https://arxiv.org/abs/2005.02264.

[6] Liang-Chieh Chen et al. "Rethinking Atrous Convolution for Semantic Image Segmentation". In: *CoRR* abs/1706.05587 (2017). arXiv: 1706.05587. URL: http://arxiv.org/abs/1706.05587.

[7] Edward Collier et al. "Progressively Growing Generative Adversarial Networks for High Resolution Semantic Segmentation of Satellite Images". In: *CoRR* abs/1902.04604 (2019). arXiv: 1902.04604. URL: http://arxiv.org/abs/1902.04604.

[8] Abhishek Dutta and Andrew Zisserman. "The VIA Annotation Software for Images, Audio and Video". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: ACM, 2019. ISBN: 978-1-4503-6889-6/19/10. DOI: 10.1145/3343031.3350535. URL: https://doi.org/10.1145/3343031.3350535.

[9] *Goal 15 — Department of Economic and Social Affairs*. URL: https://sdgs.un.org/goals/goal15.

[10] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[11] *Landcovernet*. URL: https://mlhub.earth/data/landcovernet_v1.

[12] Adrien Nivaggioli and Hicham Randrianarivo. "Weakly Supervised Semantic Segmentation of Satellite Images". In: *CoRR* abs/1904.03983 (2019). arXiv: 1904.03983. URL: http://arxiv.org/abs/1904.03983.

[13] Vasilis Pollatos, Loukas Kouvaras, and Eleni Charou. "Land Cover Semantic Segmentation Using ResUNet". In: *CoRR* abs/2010.06285 (2020). arXiv: 2010.06285. URL: https://arxiv.org/abs/2010.06285.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[15] Rose Rustowicz et al. "Semantic Segmentation of Crop Type in Africa: A Novel Dataset and Analysis of Deep Learning Methods". In: *CVPR Workshops*. 2019.

[16] Renee Su and Rong Chen. "Land Cover Change Detection via Semantic Segmentation". In: *CoRR* abs/1911.12903 (2019). arXiv: 1911.12903. URL: http://arxiv.org/abs/1911.12903.

[17] Gencer Sumbul et al. "BigEarthNet: A Large-Scale Benchmark Archive For Remote Sensing Image Understanding". In: *CoRR* abs/1902.06148 (2019). arXiv: 1902.06148. URL: http://arxiv.org/abs/1902.06148.

[18] Onur Tasar et al. "DAugNet: Unsupervised, Multisource, Multitarget, and Life-Long Domain Adaptation for Semantic Segmentation of Satellite Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.2 (2021), pp. 1067–1081. DOI: 10.1109/TGRS.2020.3006161.

[19] Priit Ulmas and Innar Liiv. "Segmentation of Satellite Imagery using U-Net Models for Land Cover Classification". In: *CoRR* abs/2003.02899 (2020). arXiv: 2003.02899. URL: https://arxiv.org/abs/2003.02899.

[20] Pavel Yakubovskiy. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2020.

[21] Zongwei Zhou et al. "UNet++: A Nested U-Net Architecture for Medical Image Segmentation". In: *CoRR* abs/1807.10165 (2018). arXiv: 1807.10165. URL: http://arxiv.org/abs/1807.10165.