Review on Semantic Segmentation of Crop Fields Using Artificial Intelligence Models

Salvador Salgado Normandia Ariadne Álvarez Reyes Diego Corrales Pinedo

***Abstract*—This paper evaluates the effectiveness and performance of five models: DeepLabV3, FCN, CCNet, ENCNet, and PSPNet; in the semantic segmentation of crop field images obtained from UAVs. Each model was used pretrained, adjusted for a custom dataset and assessed for loss, accuracy and intersection over union. They were trained locally, using various open-source libraries such as OpenMMLab, PyTorch and TensorBoard to facilitate training and visualization. The dataset itself was preprocessed to optimize accuracy and training times. Results show that ENCNet comparatively outperforms in mIoU (58.33%) and aAcc (88.12%), considering the limitations of the imbalanced dataset. FCN proved to be the least accurate, with only 84.35% aAcc and 55.58% mIoU.**

***Index Terms*—Semantic Segmentation, Deep Learning, Convolutional Neural Networks (CNN), Pixel-wise Classification Mean Intersection over Union (mIoU).**

INTRODUCTION

Semantic segmentation is a fundamental task in the field of computer vision, involving the classification of each pixel in an image into predefined categories [11]. This process is critical for numerous applications, including autonomous driving, medical imaging, and precision agriculture [**?**, **?**]. Precision agriculture is the application of technologies to assist in performing and managing agricultural activities, improving their efficiency [19]. Specifically, the collection of aerial data with Unmanned Aerial Vehicles (UAVs) and its automated analysis with semantic segmentation has proven to be useful for various applications; such as crop cover and type analysis, forest tree species labeling, weed segmentation, predictive agriculture and pest and disease identification [**?**]. Overall, the application of this technology is time and cost-efficient, increasing yields and reducing the need for human labor [1].

Recent advancements in deep learning have led to the development of various models that significantly enhance the performance of semantic segmentation tasks [**?**]. In this paper, we explore and evaluate the effectiveness of five of these models: DeepLabV3 [**?**], Fully Convolutional Network (FCN) [11], Criss-Cross Network (CCNet) [**?**], Efficient Neural Network (ENCNet) [**?**], and Pyramid Scene Parsing Network (PSPNet) [**?**]; in the segmentation of crop field satellite imagery. Thus, this study aims to provide valuable insights into the comparative strengths and limitations of each model for the semantic segmentation of crop fields, guiding further research in selecting the most suitable approaches for this task.

Specifically, pretrained versions of each model will be adjusted for a custom dataset and compared on the accuracy of their prediction.

I. STATE OF ART

In recent years, the application of artificial intelligence (AI) in agriculture has gained significant momentum. Semantic segmentation, a computer vision task aimed at classifying each pixel in an image into predefined categories, holds promise for revolutionizing precision agriculture. In the realm of semantic segmentation, researchers have made significant strides in advancing the field through the development of novel deep learning architectures. This section delves into the commonalities shared among key research papers and highlights their contributions and applications.

Fully-Convolutional Networks (FCNs) were among the first architectures to achieve state-of-the-art performance in semantic segmentation tasks. The novel skip architecture, represented in a simplified form in Figure 1, combines deep semantic information with fine appearance details, resulting in more accurate segmentations. Long et al. (2015) [11] achieved a notable 30% relative improvement in mean Intersection over Union (mIoU) on the PASCAL VOC-2012 dataset, reaching 67.2% mIoU over previous architectures. The model is highly efficient, enabling real-time applications with fast inference times of around 175 ms (averaged over 20 trials for a 500 × 500 input on an NVIDIA Tesla K40c). The FCN approach demonstrated versatility by successfully adapting and fine-tuning contemporary classification networks such as AlexNet, VGG, and GoogleNet for segmentation tasks. Since then, this has been improved to FastFCN and applied to satellite imagery successfully by Onim et al. (2020) [14].

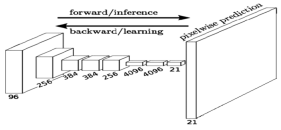

Figure 1: Example FCN Model Architecture

Pathak et al. (2015) [16] introduced Deep Convolutional Neural Networks (DCNNs), seen in Figure 2, for image segmentation using weakly annotated data. Their EM-Adapt method outperformed the previous MIL-FCN approach by 13.9% on the PASCAL VOC dataset. This is not only more accurate, but also demonstrates the potential of leveraging weak labels for training robust segmentation models. A deep learning approach for detecting cropland parcels applied by Wu et al. (2023) [21] demonstrated a high degree of precision, with an mIoU of 96.7% on a custom dataset. This precision is crucial for precision agriculture and ensures accurate field management practices.

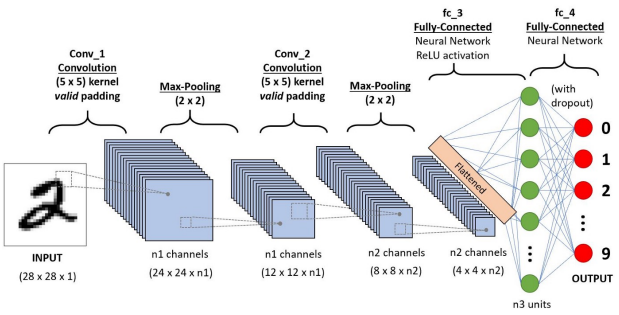

Figure 2: Example DCNN Model Architecture

He et al. (2016) [7] introduced residual learning, which was applicable to new and previous DCNNs, improving their training by addressing the vanishing gradient problem. The Residual Networks (ResNets) became a foundation for many segmentation models and were applied to SegNet, DeepLab and PSPNet as their backbone.

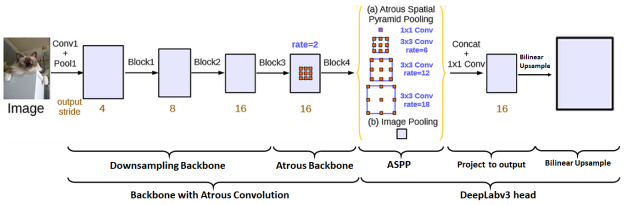

Figure 3: Example DeepLab Model Architecture

Chen et al. (2016) [4] introduced the DeepLab series, an evolution of a DCNN architecture specifically made for semantic segmentation. DeepLab utilizes atrous (dilated) convolutions to capture multi-scale context without increasing computational complexity, seen in Figure 3. Thus, it achieved a mIoU of 79.7% on the PASCAL VOC-2012 dataset. Since then, it has been improved up to DeepLabV3+ by Chen et al. (2018) [5], which integrated the Atrous Spatial Pyramid Pooling (ASPP) from DeepLabV2 (consisting of the aggregation of atrous convolutions with different sampling rates to the input feature map), and the encoder-decoder of DeepLabV3 with a new decoder module. DeepLabV3+ was applied by Wang et al. (2024) [20] for the semantic segmentation of corn plot satellite imagery, resulting in a mIoU of 87.39% with 512x512 images, up to 94.26% mIoU with 2048x2048 images.

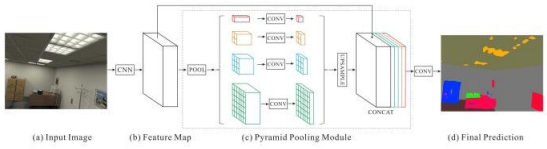

Figure 4: Example PSPNet Model Architecture

Hengshuang et al. (2016) [8] created another DCNN architecture specifically for semantic segmentation: Pyramid Scene Parsing (PSPNet). This applies pooling operations at multiple

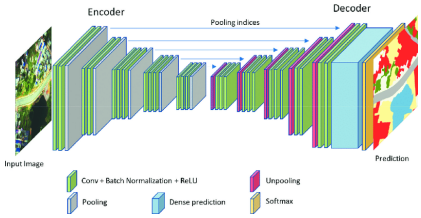

Figure 5: Example SegNet Model Architecture

levels -as represented in Figure 4- to achieve an mIoU of 85.4% on PASCAL VOC-2012. The DSCA-PSPNet architecture was created by Yuan et al. (2024) [22] specifically for the semantic segmentation of crop fields, combining PSPNet with Dynamic Squeeze-and-Excitation Context (D-scSE) blocks. It achieved an mIoU of 87.58% on a custom dataset.

Badrinarayanan et al. (2017) [3] also developed DCNNs into an architecture for semantic segmentation called SegNet. SegNet's encoder-decoder architecture, represented in Figure 5, achieved a remarkable pixel accuracy of 90.3% and a mIoU of 58.5% on the PASCAL VOC-2012 dataset, making it suitable for large-scale satellite image analysis. In fact, PSPNet and SegNet were applied for crop and weed image segmentation by Radhika et al. (2022) [17], with over 90% accuracy for both on their dataset.

II. Methodology

A. Dataset Preprocessing

The original dataset is divided into training (803 images, 803 masks), validation (171 images) and testing (172 images) directories. However, the validation directory did not contain masks for its images, therefore making it impossible to validate the accuracy of a model on them. Instead, the training directory was divided approximately 80-20 into new testing and validation directories, each with their image and mask subdirectories. Overall, we had 643 images and their masks for training and 160 images and their masks for validation.

Images and masks were reduced from 2448x2448 pixels to 256x256 pixels in order to reduce training times.

B. Mask Configuration

The classes and their corresponding color in the masks can be seen in Table 1.

Table 1: Classes and corresponding mask color

Class	Class Number	Palette (RGB)	Color
urban_land	0	(0, 255, 255)	Celeste
agriculture_land	1	(255, 255, 0)	Yellow
rangeland	2	(255, 0, 255)	Fuchsia
forest_land	3	(0, 255, 0)	Green
water	4	(0, 0, 255)	Blue
barren_land	5	(255, 255, 255)	White
unknown	6	(0, 0, 0)	Black

Because of the compression of the images, some pixels in the

masks did not correspond exactly to their RGB value. Consequently, we converted the masks by replacing pixels in an RGB value approximate to the intended palette with the class number value. The range of colors for each class and their corresponding conversion can be seen in Table 2. The result were grayscaled masks, where the color value of a pixel corresponded with the number of the class it represented.

Table 2: Classes and corresponding conversion depending on their color range

Class	Class Number	Range (RGB)	Conversion (RGB)
urban_land	0	(0, 255, 255) to (30, 225, 225)	(0, 0, 0)
agriculture_land	1	(255, 255, 0) to (225, 225, 30)	(1, 1, 1)
rangeland	2	(255, 0, 255) to (225, 30, 225)	(2, 2, 2)
forest_land	3	(0, 255, 0) to (30, 225, 30)	(3, 3, 3)
water	4	(0, 0, 255) to (30, 30, 225)	(4, 4, 4)
barren_land	5	(255, 255, 255) to (225, 225, 225)	(5, 5, 5)
unknown	6	(0, 0, 0) to (30, 30, 30)	(6, 6, 6)

C. Systems Used

The model was trained in a local environment running on one of the following hardware configurations:

- Configuration 1: Custom PC
  - CPU: Intel i3-13100F
  - GPU: Nvidia RTX 3060 Ti
  - RAM: 16 GB DDR4 @ 3200 Mhz
- Configuration 2: Custom PC
  - CPU: AMD Ryzen 9 5900HS
  - GPU: Nvidia RTX 3070
  - RAM: 16 GB DDR4 @ 3200 Mhz
- Configuration 3: Apple MacBook Pro Retina 14" (SKU: MRX73E/A)
  - CPU: Apple M3 Pro
  - GPU: 18-core
  - RAM: 18 GB

D. Environment

In order to train the models on standardized environments across the different systems, the following packages were installed using pip in an Anaconda environment running Python 3.10:

- PyTorch:
  - PyTorch 2.1.0
  - Torchvision 0.16.0
  - Tochaudio 2.1.0
  - PyTorch-CUDA 11.8 (Although Configuration 3 could not use CUDA for training)
- OpenMIM:
  - OpenMIM (Latest as of writing)
  - MMEngine (Latest as of writing)
  - MMCV 2.1.0
- MMSegmentation (Latest version of GitHub repository as of writing)
- Tensorboard:
  - TensorboardX (Latest as of writing)
  - Tensorboard (Latest as of writing)
  - Future (Latest as of writing)

E. Dataset Configuration

The modified dataset was placed in the MMSegmentation directory hierarchy as per Figure 6. The dataset's class was also added in mmseg>datasets and registered in mmseg>datasets>__init__.py and mmseg>utils>class_names.py.

```
mmsegmentation
├── mmseg
├── tools
├── configs
├── data
│   ├── deepglobe
│   │   ├── ann_dir
│   │   │   ├── train
│   │   │   ├── val
│   │   ├── img_dir
│   │   │   ├── train
│   │   │   ├── val
```

Figure 6: Directory hierarchy of dataset in MMSegmentation

The dataset's configuration was placed in configs>_base_>datasets, where the training pipeline and the validation and training dataloaders were configured with the following parameters:

- Training:
  - Batches: 8
  - Workers: 4
  - Preprocessing steps:
    - Random resize: Resize image to between 50% and 200% its original resolution
    - Random flip: 50% Chance to flip the image.
    - Photometric distortion: Change contrast, saturation, hue, brightness, etc. of image
- Validation:
  - Batches: 1
  - Workers: 4

A schedule was also configured in configs>_base_>schedules with the following parameters:

- Learning Rate: 0.001
  - Learning rate scheduler: Polynomial. Learning rate will decrease following a polynomial function
  - Minimum learning rate: 0.0001
  - Power: 0.9. Power of the polynomial function
  - Starting iteration: 0
  - Ending iteration: 20,000
  - Apply scheduler by iteration, rather than by epoch
- Iterations: 20,000
- Checkpoint: Every 2,000 iterations
- Save best iteration based on mIoU and aAcc

All of these parameters were chosen by trial and error to achieve a balance of training speed and accuracy of results in every hardware configuration used.

F. Model Configuration

As MMSegmentation provides pretrained and preconfigured versions of each model, the only changes made to the configuration files of each model in configs>_base_>models were the following:

- Mean: Adjusted the mean accordingly to that of our dataset: [0.4089, 0.38, 0.2826].
- Standard Deviation: Adjusted the standard deviation accordingly to that of our dataset: [0.1112, 0.089, 0.08].
- Number of classes: Adjusted accordingly to that of our dataset: 7.

III. Results

A. Metrics

For evaluation, we employed three commonly used metrics in semantic segmentation:

- Loss: The difference between predicted values and target values. In this case, we are using Cross-Entropy Loss, or: the difference between the predicted probability distributions and the target probability distributions.
- Average Accuracy: The average per-pixel accuracy of the prediction, regardless of class.
- Mean Intersection over Union (mIoU): The mean intersection over union across all classes.

B. Perfomance

The performance of each model -based on their best iteration- on our dataset in each of our metrics can be seen in Table 3.

More detailed results showing the evolution fo these metrics throughout execution for each model can be seen in Figures 7, 8, 10, 11 and 9.

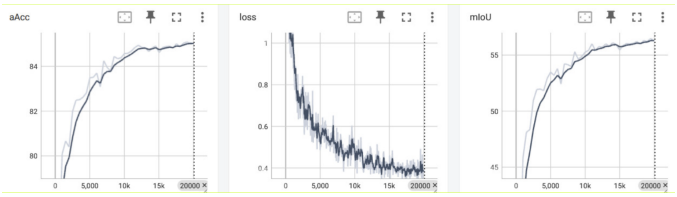

Figure 7: DeepLabv3 Prediction from Tensorboard

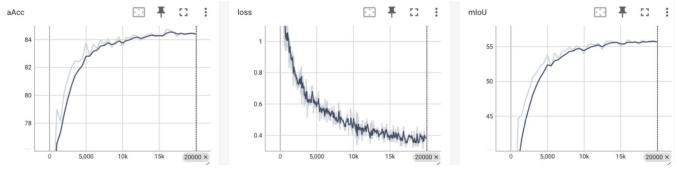

Figure 8: FCN Prediction from Tensorboard

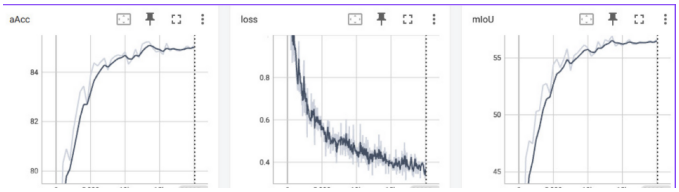

Figure 9: PSPNET Prediction from Tensorboard

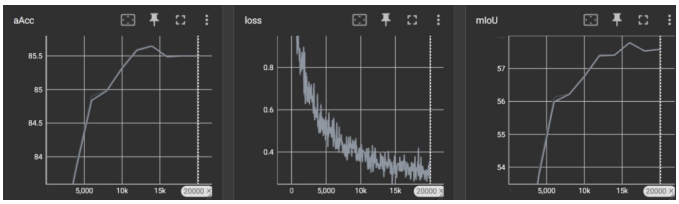

Figure 10: CCNET Prediction from Tensorboard

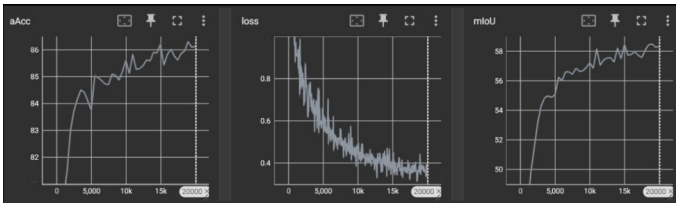

Figure 11: ENCNET Prediction from Tensorboard

Table 3: Performance comparison between 5 models with our dataset

Model	loss	aAcc	mIoU
DeepLabV3	0.3735	85.02	56.28
FCN	0.3787	84.35	55.58
CCNet	0.2613	85.5	57.59
ENCNet	0.3318	88.12	58.33
PSPNet	0.415	85.07	56.64

Our experimental results show that each model had a relatively high per-pixel accuracy of at least 84.35% with FCN. However, no model surpassed the 58.33% mIoU of ENCNet. This could be attributed to an unbalanced dataset.

Figure 16 shows the frequency of classes across the dataset. As can be seen, there is a big imbalance in the representation of classes, with Agriculture Land composing 57.4% of pixels in the dataset and Unknown just 0.05%. In fact, there were around 10 images that contained Unknown (mainly used to classify clouds) in the entire dataset. Because of this imbalance, no execution of any model surpassed 0.0% aAcc or mIoU for the Unknown class. Since mIoU is a measure of average accuracy across classes, these low results brought down the overall mIoU for all models.

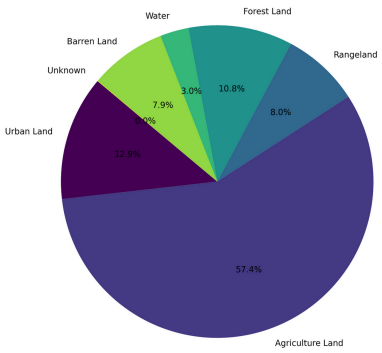

Figure 12: Frequency of classes across the dataset

Despite this, the best-performing model was ENCNet with an aAcc of 88.12% and a mIoU of 58.33%. This result can be attributed to how this model obtains global contextual information, which allows for a better understanding of context within an image and identifies the relationship and interactions between different objects and regions of the image. This contextual information is obtained by summarizing the entire scene's context into a set of encoded features. This is done using an Encoding Layer, which aggregates feature statistics from the entire image. The encoded context is then used to generate scaling factors that highlight class-specific features, allowing the model to emphasize relevant features for each class [**?**]. The EncNet includes the Context Encoding Module, which captures global contextual information and the Semantic Encoding Loss (SE-loss), enhancing the model's performance by predicting the presence of object categories and improving its understanding of global context. This dual approach significantly enhances segmentation accuracy

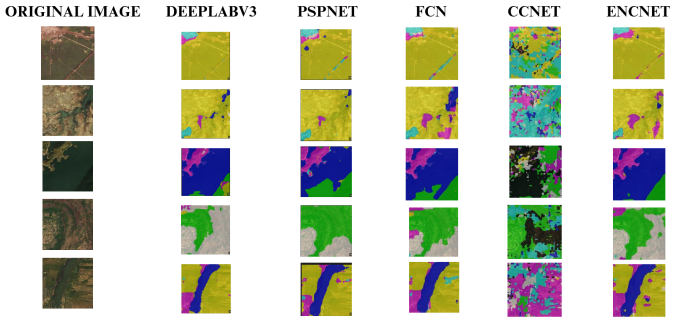

Figure 13: Predictions made by each model on test images

with minimal computational overhead.

Interestingly, CCNet had the lowest loss at 0.2613, outperforming all other models by at least 0.0705. This may be due that this model takes advantage of criss-cross attention module for each pixel. This traits allows each pixel to better obtain contextual information on all other pixels that are in it criss-cross path. This enhances feature representation and helps the model better distinguish between classes, leading to more accurate segmentation and faster convergence, thereby reducing loss [**?**].

The worst performing model was FCN with an aAcc of 84.35% and a mIoU of 55.58%. This might be because, though pioneering in adapting convolutional neural networks for pixel-wise image segmentation, are less effective compared to more recent models like DeepLabV3, CCNet, ENCNet, and PSPNet due to several limitations. FCNs capture limited contextual information and suffer from resolution loss during upsampling, which impacts their ability to segment fine details and small structures accurately. Their simple architecture lacks advanced components such as attention mechanisms and context encoding modules that enhance feature representation and segmentation accuracy. Additionally, FCNs do not explicitly model long-range dependencies, which are crucial for accurate segmentation in complex scenes. Empirical evidence from the study shows that FCN has the lowest average accuracy and mean Intersection over Union (mIoU), indicating its inferiority in handling the dataset's specific challenges. Thus, while FCNs laid the groundwork for semantic segmentation, newer models with sophisticated techniques offer significant improvements in accuracy and robustness, making them better suited for applications requiring high precision.

We can also observe the predictions made by each model against the target mask in Figure 14 and their predictions for test images in Figure 13.

Comparative analysis reveals that, despite having similar numerical results, each model can have notably different predictions. In the case of DeepLabV3's prediction of the first mask, it seems to struggle with recognizing more complex, detailed compositions of features. This may be due to its ASPP technique, which captures features at different context levels using atrous convulsions at different sampling rates. This could make it more dependant on the overall composition of the features themselves, which did not have a large variety in training due to the size of

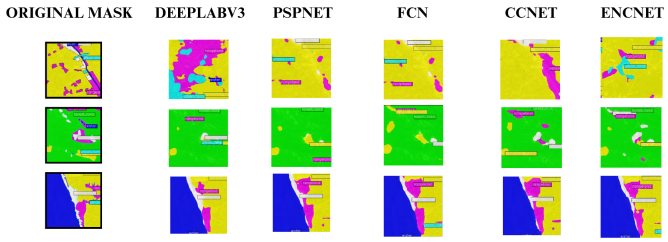

Figure 14: Predictions made by each model compared to target mask

the dataset and the limited augmentation applied to it. Furthermore, that mask is mainly composed of low frequency classes: Water, Rangeland and Urban Land. This would catalyze the low variety in feature compositions when making the prediction, as the model saw even less of these classes.

Also notable are the predictions made by CCNet on the test images. As can be seen in Figure 15 and Figure 13, they are almost completely inaccurate.

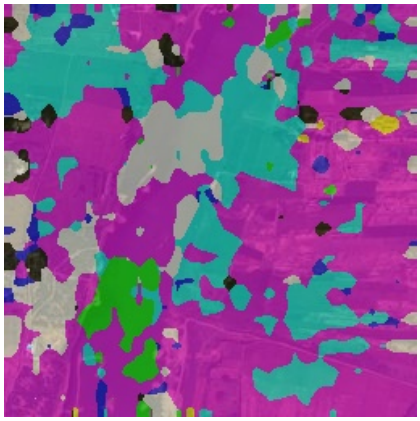

Figure 15: Initial CCNET results for image 793841

As seen in Figure 14, the results with images used for training (with a given mask) were approaching an accurate prediction, thus the test results were inconsistent. With this said, after further investigating, the different classes where not obtaining the aAcc and mIoU values visualized during training, having all of their values at 0.0%, with the only exception being urban_land. Once the model was retrained we were able to obtain the correct mIoU and aAcc for each class:

Class	IoU	Acc
urban_land	68.76	82.21
agriculture_land	84.87	92.8
rangeland	34.83	45.52
forest_land	75.57	93.33
water	76.27	81.57
barren_land	66.16	75.86
unknown	0.0	0.0

Figure 16: mIoU and aAcc results for CCNet after new training

After applying this new trained CCNet model to image 793841, the result image better approach the rest of results obtained by the rest of the models, as seen in Figure 17.

Therefore, the numerical results obtained for CCNet seen in Table 3 were accurate, but an unforeseen issue when testing made the predictions look otherwise. Correcting this issue, we can see that CCNet had comparatively accurate predictions on images it had not seen before.

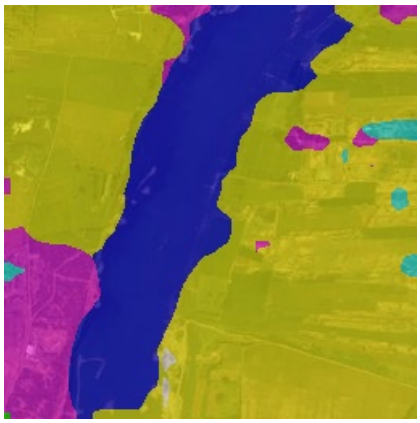

Figure 17: Image test output with fixed CCNet model

IV. Improvements

Since the mIoU of every model was affected by the imbalanced dataset, using a larger dataset with more representation of every class would yield better results.

Also, the only data augmentation applied during training was resizing, flipping and photometric distortion. Other techniques; such as cropping, rotating or random erasing, could further enrich the training data by getting the model to see a wider variety of features. This might help mitigate inaccurate predictions, such as that made by DeepLabV3 on the first mask in Figure 14.

Finally, more complex models could have been used. In the case of DeepLab and FCN, DeepLabV3+ and FastFCN are more recent versions of these models. For others, combined models, such as PSPNet-UNet (which combines PSPNet with UNet, a model recognized for its ability to discern finer details [**?**]), could have yielded better results.

V. Conclusion

In this study, we compared five semantic segmentation models: DeepLabV3, Fully Convolutional Network (FCN), Criss-Cross Network (CCNet), Efficient Neural Network (ENCNet), and Pyramid Scene Parsing Network (PSPNet) for segmenting satellite images of crop fields. Each model was evaluated for its loss, accuracy, and intersection over union.

ENCNet was the best performing, achieving an aAcc of 88.12% and a mIoU of 58.33%. However, all results were limited by the imbalanced dataset, which greatly reduced all mIoU results. This means ENCNet proved to be the most appropriate of all models in the semantic segmentation of crop field images due to its effective class-level feature representation.

On the other side of the spectrum, the now-outdated architec-

ture of FCN resulted in it being the worst performing model at just 84.35% aAcc and 55.58% mIoU.

In conclusion, ENCNet showed superior aAcc and mIoU performance. However, the limitations imposed by the dataset may not make the results representative for similar application with larger, more balanced datasets. Implementing the improvements mentioned would surely improve the results of each model and, thus, improve the accuracy of this conclusion.

REFERENCES

[1] Abdullahi, H. S., Mahieddine, F., & Sheriff, R. E. (2015). Technology impact on agricultural productivity: A review of precision agriculture using unmanned aerial vehicles. In P. Pillai, Y. Hu, I. Otung, & G. Giambene (Eds.), Wireless and Satellite Systems. WiSATS 2015 *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 154*, 388-400. Springer. doi: 10.1007/978-3-319-25479-1_29.

[2] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[3] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(12), 2481-2495. doi: 10.1109/TPAMI.2016.2644615.

[4] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(4), 834-848. doi: 10.1109/TPAMI.2017.2699184.

[5] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *European Conference on Computer Vision (ECCV)*, 801-818. doi: 10.1007/978-3-030-01234-2_49.

[6] Han, Z., Dana, K. J., Shi, J., Zhang, Z., Wang, X., Tyagi, A., & Agrawal, A. (2018). Context Encoding for Semantic Segmentation. arXiv: Computer Vision and Pattern Recognition.

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. doi: 10.1109/CVPR.2016.90.

[8] Hengshuang, Z., et al. (2016). Pyramid Scene Parsing Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6230-6239. doi: 10.1109/CVPR.2017.660.

[9] Huang, Z., Wang, X., Wei, Y., Shi, H., Liu, W., & Huang, T. (2019). CCNet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603-612.

[10] Huang, Z., Wang, X., Wei, Y., Huang, L., Shi, H., Liu, W., & Huang, S. (2023). CCNet: Criss-Cross Attention for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(6), 6896-6908. doi: 10.1109/tpami.2020.3007032.

[11] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431-3440. doi: 10.1109/CVPR.2015.7298965.

[12] Luo, Z., Yang, W., Yuan, Y., Gou, R., & Li, X. (2015). Fully Convolutional Networks for Semantic Segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431-3440. doi: 10.1109/CVPR.2015.7298965.

[13] MMsegmentation. (n.d.). https://mmsegmentation.readthedocs.io/en/main/

[14] Onim, M. S. H., et al. (2020). LULC classification by semantic segmentation of satellite images using FastFCN. *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, 471-475. doi: 10.1109/ICAICT51780.2020.9333522.

[15] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, 8024-8035. Curran Associates, Inc.

[16] Pathak, D., Krahenbuhl, P., & Darrell, T. (2015). Constrained Convolutional Neural Networks for Weakly Supervised Segmentation. *IEEE International Conference on Computer Vision (ICCV)*, 1796-1804. doi: 10.1109/ICCV.2015.209.

[17] Radhika, K., et al. (2022). Classification of paddy crop and weeds using semantic segmentation. *Cogent Engineering, 9*(1). doi: 10.1080/23311916.2021.2018791.

[18] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image*

Computing and Computer-Assisted Intervention, 234-241. Springer.

[19] Su, K., Zhu, X., Li, S., & Chen, W. H. (2023). AI meets UAVs: A survey on AI empowered UAV perception systems for precision agriculture. *Neurocomputing, 518*, 242-270. doi: 10.1016/j.neucom.2022.11.020.

[20] Wang, W., et al. (2024). Sh-DeepLabv3+: An Improved Semantic Segmentation Lightweight Network for Corn Straw Cover Form Plot Classification. *Agriculture, 14*(628). doi: 10.3390/agriculture14040628.

[21] Wu, S., et al. (2023). Extraction and Mapping of Cropland Parcels in Typical Regions of Southern China Using Unmanned Aerial Vehicle Multispectral Images and Deep Learning. *Drones, 7*(5), 285. doi: 10.3390/drones7050285.

[22] Yuan, Y., et al. (2024). DSCA-PSPNet: Dynamic Spatial-Channel Attention Pyramid Scene Parsing Network for Semantic Segmentation in Precision Agriculture. *Computers and Electronics in Agriculture, 198*, 107034. doi: 10.1016/j.compag.2022.107034.

[23] Zhang, H., et al. (2018). Context Encoding for Semantic Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7151-7160. doi: 10.1109/CVPR.2018.00747.

[24] Zhang, H., et al. (2020). ResNeSt: Split-Attention Networks. arXiv preprint arXiv:2004.08955.

[25] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid Scene Parsing Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6230-6239. doi: 10.1109/CVPR.2017.660.

[26] Zhu, Z., Wang, S., Bai, X., Yao, T., Urtasun, R., & Dai, J. (2019). Learning Feature Aggregation for Deep Scene Parsing. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 7073-7083. doi: 10.1109/ICCV.2019.00723.

[27] Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8697-8710. doi: 10.1109/CVPR.2018.00907.

[28] Zou, X., & Shi, Y. (2022). Research on Semantic Segmentation of UAV Images Based on Deep Learning. *Remote Sensing, 14*(15), 3556. doi: 10.3390/rs14153556.