Detecting and Segmenting Solar Farms in Satellite Imagery: A Study of Deep Neural Network Architectures

Erling Olweus* and Ole Jakob Mengshoel*

Abstract— **In line with global sustainability goals, such as the Paris Agreement, accurate mapping, monitoring, and management of solar farms are critical for achieving net zero emissions by 2050. However, many solar installations remain undocumented, posing a challenge. This paper studies semantic segmentation using deep neural networks, including networks constructed using network architecture search (NAS), for solar farm detection. Semantic segmentation has evolved through technologies like Fully Convolutional Networks and U-Net, which have shown strong performance on satellite imagery. For NAS, Differentiable Architecture Search and its variants like Auto-DeepLab have become efficient ways to automate the creation of neural network architectures. This work compares models generated using Auto-DeepLab to Solis-seg, a Deep Neural Network optimized for detecting solar farms in satellite imagery. Solis-seg achieves a mean Intersection over Union (IoU) of 96.26% on a European Sentinel-2 dataset, with Auto-DeepLab models lagging slightly behind. Our results for Solis-seg also challenge the prevailing method of using transfer learning from classification tasks for semantic segmentation. Thus, this work contributes to both the field of earth observation machine learning and the global transition to renewable energy by studying an efficient, scalable approach to tracking solar installations. We believe that this paper offers valuable insights into applying advanced machine learning techniques to solar farm detection and can be useful for further research in earth observation and sustainability.**

I. Introduction

Context. With the Paris Agreement of 2015, most nations globally have committed to reaching net zero emissions by 2050. Achieving this goal necessitates a large-scale shift from fossil fuels to renewable energy alternatives, such as solar and wind power. Currently, fossil fuels account for approximately 80% of global energy consumption and are responsible for the emission of large amounts of CO_2. The transition towards green energy sources——including wind, hydro, and solar——is crucial for fulfilling the climate objectives set by the Paris Agreement within the specified timeline. Clearly, the development and management of a solar energy infrastructure is a key component of this transition.

The satellite images we consider come from the European Space Agency's Sentinel project,[1] specifically the Sentinel-2 mission. Sentinel-2, launched in 2015, focuses on tracking changes on the Earth's surface. It uses a multispectral camera, which captures images across 13 spectral bands with a resolution of $10m^2$ per pixel. While this level of resolution could pose problems for certain tasks, it tends to be sufficient for large structures such as grid-connected photovoltaic (PV) plants. These plants are often larger than $10{,}000m^2$ [11], which makes them distinguishable even at these resolutions.

Challenges. Detecting solar panels from satellite images promises to partly address the issue of managing the solar energy infrastructure. One way to accomplish this detection task is to use Machine Learning (ML), including Deep Neural Networks (DNNs) [23], [11], [8], [4]. Despite the demonstrated prowess of Neural Architecture Search (NAS) in surpassing human-designed architectures in image classification [6], its application in the field of solar farm segmentation from satellite imagery remains uncharted territory. While PV plants often are large, Sentinel-2's resolution of $10m^2$ per pixel makes it a challenge to discern PV plants from similar-looking structures, such as rice paddies, greenhouses, parking lots, and lakes. Sentinel-2's multispectral camera partially mitigates this issue by utilizing the unique spectral profile of solar farms [11], [9].

Even though NAS has seen extensive use in well-established benchmarks, its practical application for novel datasets is still under-researched [29]. Thus we consider several research challenges and questions related to detecting and segmenting solar farms in satellite images in this paper: The questions relate to the performance of different DNN architectures, DNN transfer learning with fine-tuning for segmentation versus learning to segment from scratch, the computational cost of NAS for DNNs, and the comparison of NAS-generated DNNs versus foundation models.

Contributions. Recognizing the challenges mentioned above, this work[2] makes several contributions:

- Our Solis-seg DNN model clearly outperforms an incumbent model, Solis-transfer. Solis-seg attains the highest validation mIoU on a major solar farm dataset with continental scale coverage known to us, outperforming SolarNet [8] and Kruitwagen *et al.*'s model [11] on their respective datasets.
- Contrary to previous findings [8], our results suggest that transfer learning (from image classification to segmentation) may not work so well. Transfer learning can be time-efficient, but may inadvertently compromise segmentation performance when compared to training a model from scratch, as we did with Solis-seg.
- Our focus on solar farm segmentation in Sentinel-2 satellite imagery serves as a real-world study of NAS in semantic segmentation. Much NAS research focuses on classification, especially on the ImageNet or CIFAR

*Erling Olweus is with Atlas, Oslo, Norway; this work was done while he was at NTNU. erlingolweus@gmail.com

*Ole Jakob Mengshoel is with the Department of Computer Science, NTNU, Trondheim, Norway. ole.j.mengshoels@ntnu.no

[1]https://sentinels.copernicus.eu/web/sentinel/home

[2]This paper builds upon the MS Thesis of Erling Olweus [21].

datasets, with few studies on semantic segmentation.

This work was conducted in collaboration with Atlas.[3] One application of Atlas' cloud-native GIS technology is to evaluate locations for solar farm development. Our focus in this paper is on segmenting solar farms from satellite imagery.

II. Background and Research Questions

Identifying Solar Farms from Images. Several studies have explored the detection of solar panels in satellite imagery, utilizing both Artificial Neural Networks (ANNs) and other methods. For instance, a random forest model was employed by Plakman *et al.* [23] to detect solar panels, and this model was trained and evaluated using a publicly accessible dataset from the Netherlands. Hou *et al.* developed SolarNet, a system that integrates Expectation-Maximization Attention Networks and a U-Net architecture, to uncover new photovoltaic (PV) systems in China [8]. Meanwhile, in Brazil, a study used high-performing segmentation models with different pre-trained backbones [4]. Stanford researchers have identified and compiled US solar installations into the publicly accessible DeepSolar database [31].[4] Astraea Earth trained a Deep Convolutional Neural Network in the US and used it to identify new Chinese solar farms [12].

Kruitwagen *et al.* released a global dataset of solar energy facilities, expanding the existing asset-level data by an impressive 432% [11]. This work represents the most substantial single contribution to this field to date, measured by the number of previously unknown facilities discovered and added to public datasets. Focusing on PV platforms larger than 10,000m^2, they achieve a precision of 98.6%, a recall of 90%, and an Intersection over Union (IoU) of 90% for the segmentation task on their test set. They employ a U-Net-based Convolutional Neural Network (CNN) model and two sources of remote sensing imagery to achieve these results. Non-visible bands of Sentinel-2 are utilized, demonstrating their significant role in the model's solar panel recognition.

Research Question 1 (RQ1): How well do different DNN model architectures, including NAS models, perform semantic segmentation of solar farms in Sentinel-2 imagery?

Semantic Segmentation. Semantic segmentation is an area where CNNs have success, sparked by the victory of the Fully Convolutional Network (FCN) [17] in the COCO segmentation task in 2014. This achievement is credited to replacing the fully connected layers at the end of popular networks like AlexNet, VGG, and GoogLeNet with convolutional layers. This modification led to significant speed increases during both forward and backward passes in training [17]. The method employs upsampling techniques to restore the output feature map of the image to its original size for pixel-by-pixel predictions.

U-Net was improved in 2017 by incorporating the output before each subsampling stage as input during the upsampling phase. This enhancement aids in more accurately mapping recognized features back to the original image size [24]. In comparison to other approaches, U-Net is particularly effective for semantic segmentation on remote sensing imagery due to its strong performance even with little training data [27]. The U-Net architecture has been used for semantic segmentation of solar farms [8] [11].

Dilated convolutions, also referred to as "atrous" convolutions, are a variant of CNN layers that utilize dilated kernels to enlarge the receptive field of a layer [1]. Traditional CNNs determine the receptive field of a layer based on its filter size and stride. However, dilated convolutions employ filters with gaps or "dilations," the size of which is decided by the dilation rate, enabling the filters to cover a larger input area without augmenting the number of parameters or computational complexity. This benefits semantic segmentation, where maintaining spatial resolution while increasing receptive field to capture long-range dependencies in data is crucial [7].

Research Question 2 (RQ2): How does transfer learning [8] (from image classification to segmentation) compare to training a DNN model from scratch when it comes to segmenting solar farms in Sentinel-2 images?

Neural Architecture Search. The roots of Neural Architecture Search (NAS) can be traced back to 1989, when an evolutionary algorithm was first applied to optimize ANN architectures [18]. Since that seminal work, an array of diverse algorithms has been introduced to enhance the efficiency and robustness of neural architecture generation. NAS algorithms fall into two main categories: one-shot methods and black-box methods. A NAS method may not fall squarely into either category and may straddle both [29]. Different NAS techniques, including Bayesian optimization, evolutionary algorithms, and reinforcement learning, have been widely adopted [16]. One downside of these techniques is their significant computational cost. Some studies report using thousands of GPU days for experiments [6], [29]. In contrast, one-shot methods have gained traction due to their considerable efficiency. These methods manage to generate promising results within a far shorter time span [30].

NAS algorithms are designed to refine architectures within a specific search space, with cell-based search spaces being notable [29]. In these spaces, DNN architectures are conceptualized through a sequence of "cells." A cell is a modular component that, when combined with other cells, creates larger neural networks [5]. Each cell represents a unique arrangement of layers and connections and is typically repeated in a set macro-architectural pattern, facilitating the creation of a wide array of network architectures [15], [20].

Differentiable Architecture Search (DARTS) [15] presents a novel approach to network architecture search. DARTS combines a cell-based search space and a gradient-based one-shot model, facilitating efficient exploration and evaluation of architectures. The search space is structured as a Directed Acyclic Graph (DAG) where each edge performs one of eight potential operations.

Auto-DeepLab (ADL) is a specialized DARTS variant developed to create effective architectures specifically for

[3] https://atlas.co

[4] https://deepsolar.web.app/

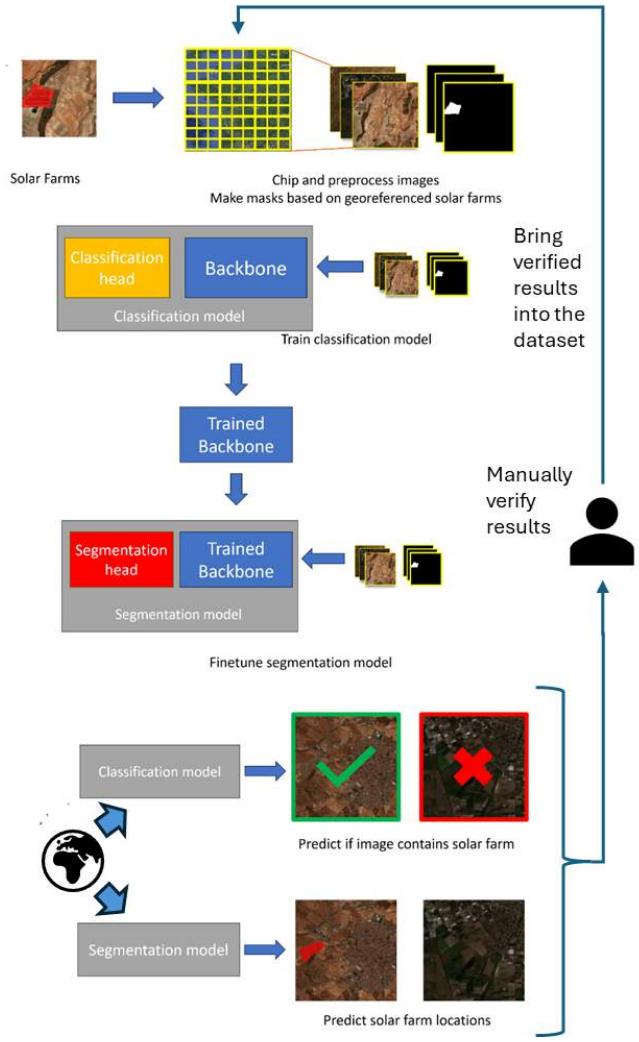

Fig. 1: A typical ML pipeline for discovering new solar farms, also showing how Solis-transfer is trained. Solis-seg is trained in a similar but simpler way since there is no training and transfer of a classification model backbone.

semantic segmentation within the DeepLab framework [14]. Originating from the work of Liu et al., ADL enhances the DARTS-based, cell-centric search space [15] by incorporating a hierarchical component to manage spatial resolution during the architecture search [19].

NAS is computationally demanding, introducing substantial overhead to an ML pipeline. This raises the following two research questions.

Research Question 3 (RQ3): When is the extra cost of performing NAS worthwhile for the purpose of detecting and segmenting solar cells in Sentinel-2 satellite imagery?

Research Question 4 (RQ4): How do highly specialized models discovered through NAS stack up to generalized foundation models, like GPT-4 [22] and SAM [10], that excel across a multitude of tasks within a domain?

III. Methods and Models

A. Detecting Solar Farms

To discover solar farms in remote sensing imagery, certain processes are smilar across various ML studies [12], [11], [31], [8]. These processes form a complex, sophisticated pipeline for both training an ML model and deploying it in real-world scenarios. Although there are slight variations, the core processes and their ordering remain largely consistent as refleced in the pipeline of Figure 1.

The pipeline commences (see top of Figure 1) with the identification and labeling of known solar farms on satellite images as georeferenced polygons, often using a GIS tool such as QGIS.[5] These images then go through a series of preprocessing operations, including cloud removal, image standardization, and chipping or subdividing the images into smaller segments that can be efficiently processed by the network. These chips[6] become our dataset.

The pipeline's next phase involves training a classification model using a dataset of chips, with and without solar farms. Once trained, a segmentation head (see middle of Figure 1) is attached to the model and this amalgamated DNN is further fine-tuned for segmentation tasks. Approaches differ in whether they completely freeze the weights of the backbone, or allow the weights to be modified in the training of the segmentation model. Slight modifications are usually introduced to the backbone to preserve spatial information during its application for segmentation tasks [7].

Hou *et al.* largely attribute the success of this approach to the activation mapping for the classification model, which resembles a dense prediction or segmentation architecture [8]. This claim is intuitively plausible, as the model needs to learn the unique features of solar farms to correctly predict their presence in an image. An advantage of this transfer learning strategy is the time efficiency it offers compared to training an entirely new network from scratch.

In the pipeline's last phase, the trained models are deployed over extensive areas as represented by the globe at the bottom of Figure 1. The images of these areas undergo the same preprocessing steps, without prior manual identification and labeling of solar farms. Following this process, the models' findings are manually inspected and confirmed solar farms are added to the dataset. The cycle can repeat, as depicted in Figure 1, with the augmented dataset.

We contrast a model pre-trained on solar farm classification with one exclusively trained for segmentation tasks. We refer to these DNN models as Solis-transfer and Solis-seg respectively. They are both ResNet-50 models with dilated convolutions instead of strided convolutions and with a DeepLabV3 segmentation head (inspired by previous research [1]). The code to train the model is publicly available.[7]

[5] https://qgis.org/en/site/

[6] The chipped images derive their ground truth from the labeled polygons. If any segment of the image overlaps with a part of the polygon, it is labeled as a "solar farm". For classification purposes, any chip encompassing a portion of a solar farm is labeled as "solar farm".

[7] https://github.com/TheAtlasRepository/solis.

TABLE I: Comparison of one-shot NAS methods specializing in segmentation on the Cityscapes test set

Architecture	GPU Days (search)	mIoU
Auto-DeepLab [14]	**3**	**82.1**
DCNAS [32]	**5.6**	**84.3**
GAS [13]	6.7	73.5
SqueezeNAS [25]	14.6	75.54
FasterSeg [2]	2	71.5

B. Network Arcitecture Search (NAS)

Determining the appropriate NAS methodology hinges on several factors. For us, several criteria emerged as critical: the computational expense associated with the search, the task specificity, the documented performance of the algorithm, and the availability of source code or libraries for implementing the chosen method.

Our analysis, detailed in Section III-C, led us towards one-shot models, primarily due to their computational efficiency [29]. Among one-shot methodologies outlined in the NAS surveys by White *et al.* [29] and Elsken *et al.* [19], Auto-DeepLab (ADL) appeared as the best choice. Its focus on semantic segmentation, coupled with our prior experiences with DeepLab, contributed to our choice.

Since Auto-DeepLab's introduction in 2019, various works have built upon it, with changes to the search space or specific tailoring for tasks such as real-time video segmentation [13], [2], [25]. Among these works, DCNAS by Zhang *et al.* [32] is the one that directly enhances the performance of ADL on inference (as measured in mIoU as shown in Table I). Regrettably, the lack of public access to the DCNAS code limits its experimental usage by others. DCNAS also has almost double the search time of Auto-DeepLab, which would make DCNAS challenging to use with our dataset. Given these considerations, we opted for the original Auto-DeepLab. The availability of Auto-DeepLab's source code simplifies its integration into our experimentation process.

C. Details of Selection Criteria

To study the effectiveness NAS on our task, we chose Auto-DeepLab (ADL) as our NAS model. The selection was based on multiple criteria:

- **Computational Efficiency**: One-shot models like ADL significantly reduce the computational burden, making experimentation quicker.
- **Task Specificity**: ADL specializes in semantic segmentation, directly aligning with our research focus.
- **Documented Performance**: Previous works have validated ADL's effectiveness, providing a reliable starting point for our own evaluations [29].

The method for architecture search mirrors previous work [14], with the main difference being that we run the search on subsets of the dataset as discussed in Section IV. After searching for 40 epochs we decode the best model found and train it from scratch for 100 epochs on the entire dataset with an 80/20 train test split.

IV. Experimental Results

A. Experimental Settings

1) Hardware: Most of the experiments were conducted using hardware from the NTNU IDUN High-Performance Computing Cluster [26]. This included either an NVIDIA A100 GPU equipped with 40/80GB memory or an NVIDIA V100 GPU with 32GB memory. An NVIDIA RTX 3090 GPU[8] was also used for some tests.

2) Dataset: We use a proprietary dataset of Atlas, encompassing solar farms situated across Europe. This expansive dataset, consisting of 224x224 pixel chips from 12-band Sentinel-2 level-2A (l2a) images,[9] contains more than 200,000 images with about a 50/50 split between positives (containing solar farms) and negatives. All the positives additionally have masks. A couple of thousand are manually drawn, and the rest are sourced from previous Solis deployments, OpenStreetMap,[10] or other sources with free available masks for solar farms. While Sentinel-2 captures 13 bands, band B10 is excluded from l2a as it is used to monitor the atmosphere rather than the ground.

Given the resource-intensive nature of NAS and concerns about time spent, representative subsets of this dataset are employed during the architecture search process. While some of data are proprietary, the framework presented is dataset-agnostic and could potentially be employed with similar datasets, such as that of Kruitwagen *et al.*[11]

As highlighted by Elsken *et al.* [6], the scale of disparity between the sampled and full dataset size can influence the relative ranking of architectures. This is a potential concern, given that our final objective is optimizing the validation score on the larger dataset, not the subset. Nonetheless, the two tasks are closely related, and we believe that a random selection of images from a wide geographic coverage incorporating diverse geographical features will mitigate potential biases. Furthermore, the success achieved on relatively smaller datasets (around 1000–2000 images) as reported by Hou *et al.* [8] and Plakman *et al.* [23] is noteworthy. Considering China's diverse landscape, this observation is particularly pertinent for Hou *et al.*'s SolarNet [8].

To further diversify the training process, both during the search and retraining phases, we implement data augmentation. Specifically, images are subjected to horizontal and vertical flips with a 50% probability each before being fed into the model within the training loop. This data augmentation strategy makes for a robust and varied training dataset, enhancing the model's generalization capabilities even with smaller dataset sizes.

[8] https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3090-3090ti/

[9] https://sentinels.copernicus.eu/web/sentinel/sentinel-data-access/sentinel-products/sentinel-2-data-products/collection-1-level-2a

[10] https://wiki.openstreetmap.org/wiki/Tag:generator:source\%3Dsolar

[11] https://zenodo.org/record/5005868

Name	Architecture	mIoU	F1-score
Solis-seg (our)	ResNet+DeepLabV3	**0.9629**	**0.9621**
10k-L	Auto-DeepLab	0.9593	0.9582
ADL-cs	Auto-DeepLab	0.9586	0.9575
10k	Auto-DeepLab	0.9567	0.9555
random	Auto-DeepLab	0.9565	0.9552
Solis-transfer	ResNet+DeepLabV3	N/A	0.89

TABLE II: Top five models ranked by validation mIoU achieved during retraining; Solis-transfer is a reference point.

Name	val mIoU (search)	val mIoU (retrain)	train mIoU (retrain)
2k	0.536	0.9563	0.9637
5k	0.733	0.9550	0.9630
10k	0.741	**0.9567**	**0.9653**
20k	0.785	0.9531	0.9607

TABLE III: mIoU results for different dataset sizes; 10k is considered the best-performing ADL model.

B. Experimental Objectives, Metrics, and Models

1) Objectives and Methodologies: Our research aims to evaluate Auto-DeepLab's performance, particularly focusing on its adaptability to different input data sizes and types. This is directly tied to Experiment 2, which aims to understand how these factors influence the NAS process.

2) Performance Evaluation of Different Models: These are the DNN models that we focus on here:

- **Solis-transfer** and **Solis-seg**: These ResNet-based models serve as points of comparison to the ADL models.
- **2k**, **5k**, **10k**, **20k**, and **20k**: These ADL models result from NAS experiments using corresponding Sentinel-2 dataset sizes.
- **10k-L**: This ADL model results from taking the best-performing model identified via NAS, 10k, and retraining it with a filter multiplier of $\mathcal{F} = 48$, using the Auto-DeepLab-L configuration [21].
- **ADL-cs**: This ADL model, found to be the best-performing by Liu *et al.* during their Cityscapes search [14], provides an external point of comparison.
- **random**: A randomly generated architecture, using ChatGPT, is a second external point of comparison.

The primary metric is validation set mIoU, except for the Solis-transfer model where F1-score is used due to mIoU not being captured during training. Further details are provided in Appendix A as well as accompanying Web sites.[12]

Final models were trained on the complete Solis dataset, adhering to an 80/20 train-test split. This training regimen aligns with our final experiment (see Section IV-F).

3) Performance Evaluation of Final Models: To assess its generalization capabilities, the best-performing model from the early experiments is deployed in a real-world scenario to discover new solar farms. Here, we test the best model on data from untrained regions, the state of New York (see Section IV-F).

C. Experiment 1: Effectiveness of Transfer Learning

The purpose of this experiment is to evaluate the effectiveness of transfer learning, particularly as employed by the Solis-transfer model. Our intention is to investigate if the prevalent approach of transfer learning from classification tasks remains the preferred strategy or if training directly on segmentation tasks from the outset can produce improved results. We also implement a variant of the Solis-transfer model, Solis-seg, trained exclusively on segmentation.

Experimental results are shown in Table II. Contrary to our expectations, not only did the Solis-seg model exhibit a marked performance improvement compared to Solis-transfer by increasing the best F1-score from 0.89 to 0.9621, it even became the best-performing model (even though the differences between the top models are relatively small). With a final validation mIoU of 0.9629, it surpassed all the models obtained through our NAS experiments, emerging as the only model breaching the 0.96 threshold. Table II provides a summary of the top five models, ranked based on the mIoU scores achieved during the retraining phase. It underscores the dominance of Solis-seg in this experiment.

D. Experiment 2: Impact of Dataset Size on NAS

In this experiment, we explored how the size of the dataset influences the outcome of NAS. Due to computational limitations, we opted for smaller subsets of the full dataset, specifically sizes of 2,000, 5,000, 10,000, and 20,000 images, referred to as 2k, 5k, 10k, and 20k respectively. These subsets were considered to be representative samples for the purpose of architecture discovery.

During the search, we observe a correlation between the dataset size and the resulting validation mIoU as seen in Figure 2. The smallest dataset (2k) shows more variability in results, indicating sensitivity to data selection. Most of the searches reached peak performance shortly after 20 epochs, thus we scrutinize the structural components of the resulting architectures. Despite similar performance metrics, the architectures exhibit considerable structural differences.[13]

When these architectures were retrained using the complete dataset, the performance differences noted during the search phase became less significant. The model initially trained on the largest dataset (20k), which exhibited the highest mean Intersection over Union (mIoU) in the architecture search, surprisingly showed the lowest performance upon retraining with the full dataset, see Table III.

The results do not indicate a strong correlation between dataset size and final performance, suggesting that either an element of randomness was at play or that the smaller subsets were sufficiently representative of the full dataset for this application.

[12]The Solis-transfer model can be found in the repository at `https://github.com/TheAtlasRepository/solis` as the fully trained DeeplabV3 with ResNet50 backbone.

Fig. 2: Validation mIoU on y-axis for different dataset sizes during search. The numbers to the right (40,25, 23, and 20) indicate in what epoch the best-performing architecture was found. The x-axis reflects the number of epochs. For each graph, smoothed (strong color) and raw versions (faint color) are shown.

E. Experiment 3: Comparative Evaluation

With the launch of Meta's Segment Anything Model (SAM) [10], we aimed to measure its performance against our best model, Solis-seg. Ideally, we would fine-tune SAM and compare its performance metrics with those of Solis-seg. However, as this exceeds the scope of our current study, we instead used the publicly accessible SAM.[14]

For our comparison, we uploaded RGB images from the validation set, on which Solis-seg was not trained, to SAM. We used the "segment everything" function to analyze the entire image for coherent structures. SAM was not given any specifics about what to identify, nor were any images provided for training. These results are in other words strictly zero-shot, with SAM attempting to segment any structures in the image.

Some experimental results are shown in Figure 3. Three distinct outcomes emerge from this analysis. Notably, SAM's performance varies significantly across different images. In image A, where the solar farm is almost invisible to the naked eye, Solis-seg presumably gains an advantage through the use of spectral bands, as SAM fails to detect it entirely. In image B, SAM clearly distinguishes the solar farm from its surroundings, arguably drawing a more refined boundary than the ground truth. For image C, it not only identifies the solar farm but also segregates the various racks into individual partitions. However, these solar farms are relatively large, and many images depict smaller solar farms that blend into the environment and are challenging to detect even with the human eye. We suspect that a model trained solely on RGB might face increased difficulties with such images given Sentinel's resolution. While it might be possible to fine-tune SAM with spectral bands, it is uncertain whether this would enhance its accuracy [27].

Despite SAM's impressive performance on some images, this task of discovering new facilities might favor a specialized model such as Solis-seg over a generalized zero-shot model. An interesting approach for future work would be to combine SAM with a more specialized model to optimize detection and obtain finer segmentations.

F. Experiment 4: Finding Solar Farms in New York

This experiment aims to deploy a model on novel satellite imagery to identify solar farms, testing its viability as a tool for discovering solar farms on unseen images. We deployed Solis-seg, our best-performing model, to detect new solar farms in satellite imagery covering New York State from 2022. The model identified 874 polygons, which, after accounting for multiple polygons representing single facilities, represent approximately 583 potential solar farms.

Experimental results are illustrated in Figure 4 and Figure 5. Figure 4 depicts a solar farm found by our Solis-seg model. Several of these locations are not documented in publicly available databases such as OpenStreetMap.

While Solis-seg was effective in identifying numerous solar farms, its performance was not as robust in the New York dataset as it was with the solar farms in our validation set. We noticed that the model detected some solar farms and entirely missed others, suggesting challenges in generalizing to new regions. A related challenge is the verification of the model's predictions due to the absence of up-to-date, high-resolution imagery. As illustrated in Figure 5, this makes it difficult to determine whether certain polygons are solar

[13]Due to limited space we are not showing the architectures in this paper and refer to the MS Thesis [21].

[14]`https://segment-anything.com/demo`

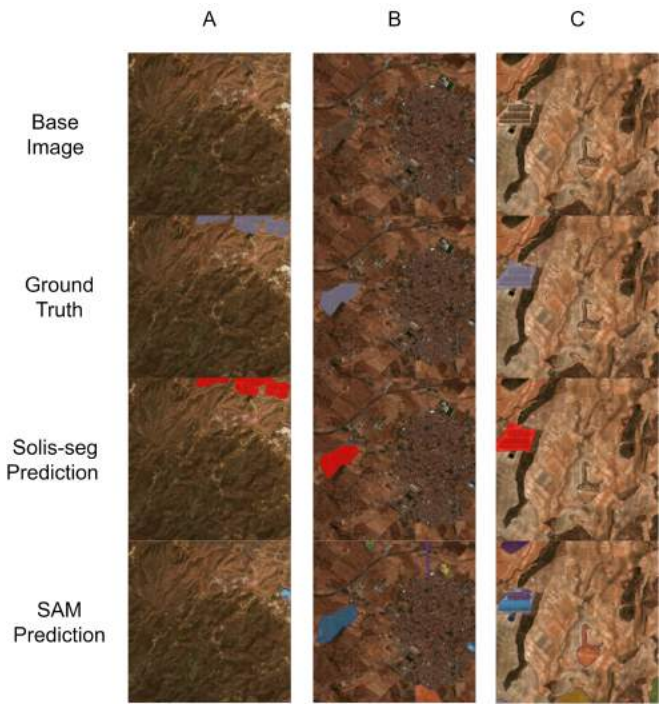

Fig. 3: A comparison of predictions from our Solis-seg model (third row) versus SAM [10] (bottom row) for three different images A, B, and C.

Fig. 4: Example of a solar farm detected in New York state.

farms or false positives. Despite these challenges, Solis-seg's real-world deployment was largely successful. We have made the dataset of detected solar farms in New York publicly available in a GitHub repository.[15]

V. Discussion of Experimental Results

We now discuss the RQs identified in Section II in light of the experimental results presented in Section IV.

A. Re-evaluating the Efficacy of Transfer Learning (RQ2)

The Solis-seg and Solis-transfer models differ solely in their training methodology as detailed in Section IV-C. Solis-seg is dedicated to the exclusive task of semantic segmentation of solar farms, whereas the ResNet component of Solis-transfer is initially trained to identify whether an image does or does not contain a solar farm (classification), and only thereafter it is trained for the task of segmentation.

Despite numerous trials with Solis-transfer, it has yet to surpass an F1 score of 0.89 as seen in Table II. In contrast, the single experiment conducted with Solis-seg yields a significantly superior F1 score (0.962). This highlights the effectiveness of Solis-seg's task-specific training. The increase in performance is attributable to the switch in training strategy, as no other alterations were made during training.

This surprising outcome, in light of previous research [28], suggests that the methods employed by the classification model differ considerably from the pixel-wise recognition performed during semantic segmentation. The competencies required for these tasks might diverge to the extent that proficiency in one (classification) could potentially impede the ability to learn the other (segmentation).

Moreover, the experimental results reported in Table II and Table V highlight how the benefits of transfer learning are not universally applicable, but are contingent upon various factors including the degree of similarity between the source and target tasks, and the specific nature of these tasks.

In summarizing our findings in Table V, we note that our best-performing model surpassed the IoU score of 0.9 obtained by Kruitwagen *et al.* [11]. While an apples-to-apples comparison between their and our DNNs using the same datasets is infeasible, our results are notable given the markedly higher relative score on our dataset.

B. Robustness for Satellite Image Segmentation (RQ1)

We now discuss our study of finding solar farms in images for which the model was not trained. The results indicate that we successfully identified solar farms in these untrained images. An interesting finding from our experiments, reported in Table V, is that out of 14 NAS trials, only a single architecture outperformed any of the benchmarks, excluding Solis-transfer. This raises questions about the effectiveness and cost-benefit value of DARTS and Auto-DeepLab in this context, which will be further elaborated in Section V-C.

Surprisingly, the randomly sampled architecture produced by ChatGPT outperformed almost all of the architectures identified via NAS (see the model referred to as "random" in Table V). While this might be an outlier event and additional random samples should be examined for validation, it raises questions about the consistency and effectiveness of NAS in yielding superior architectures for certain use cases.

Furthermore, in Table II, we observe that the performance of most models was closely aligned with that of the random

[15] https://github.com/eolweus/autodeeplab.

Fig. 5: High-resolution image of an object the model thought was a solar farm. It appears to be a gray rooftop.

model. This suggests that the search space may be densely populated with models that deliver comparable performance, making it difficult to continually progress toward an optimal solution. This hypothesis is supported by studying the search graphs, particularly the observation that most searches peaked early. This pervasive challenge is credited by Chen and Hsieh [3] to DARTS' tendency to reach strong local minima in the search space.

The influence of spectral bands in Sentinal-2 images on NAS emerged as a significant factor. Separate trials were conducted with architectures identified using a 10,000-image dataset.[16] A model trained solely with RGB data underperformed compared to models that utilized additional spectral bands. Further trials are needed to conclusively attribute this performance discrepancy to spectral band usage, but this hints at Auto-DeepLab's potential to leverage this extra information effectively.

Overall, the top-performing NAS model, 10k-L, only slightly lags behind the best-performing model, Solis-seg (see Table II). This suggests that under appropriate conditions, NAS can generate architectures that approach or even match the state-of-the-art, even in specialized applications such as satellite imagery segmentation. The robustness and adaptability of NAS, despite its complexities and challenges, underscore its potential.

C. Computational Trade-offs in NAS Application (RQ3)

In evaluating the efficiency of NAS, two main aspects come into play: the potential performance gain and the importance of this gain for the specific application. In our study, NAS proved to be less time-efficient when compared to traditional methods (see Table IV). Specifically, the Solis-seg model took 46 hours to train, while the average training time for NAS-derived architectures was around 59 hours. These figures do not yet account for the additional search time required by NAS, as shown in Table IV. When considering both the search and training times, the total computational time for NAS architectures vastly exceeds that for Solis-seg. This casts doubt on the cost-effectiveness of NAS, particularly when an off-the-shelf model like ResNet50-DeepLab (Solis-seg) performed best on our dataset after 14 NAS trials (see Table II).

[16]We refer to the MS Thesis [21] for details.

Dataset size	Search time (h)
2k	20
5k	41
10k	62
20k	104

TABLE IV: Dataset size and search time.

Name	Architecture	mIoU	F1-score
Solis-seg	ResNet+DeepLabV3	0.9629	0.9621
10k-L	Auto-DeepLab	0.9593	0.9582
ADL-cs	Auto-DeepLab	0.9586	0.9575
random	Auto-DeepLab	0.9565	0.9552
2k	Auto-DeepLab	0.9563	0.9550
Solis-transfer	ResNet+DeepLabV3	N/A	0.89

TABLE V: The top 5 models ranked by validation mIoU obtained during retraining. The model 10k is omitted as it has the same architecture as 10k-L. It would have been placed between ADL-cs and random, see Table II.

Reflecting on the top five models derived from our study, as shown in Table V, three out of the five top performers are baseline models that we originally proposed for comparison. Interestingly, even a randomly suggested model outperformed all but one model discovered through NAS.

While the search outcomes reported in Table III might not seem particularly outstanding—failing to surpass a ResNet-based model, marginally exceeding a model found by searching on a different dataset, and the curious case of a random model outperforming all but one NAS architecture—it is important to recognize that the top model found through the search, 10k-L, does not lag significantly behind the best model, Solis-seg.

In Table III and V we note that all models outperform Solis-transfer, implying that the DARTS search space is replete with viable architectures. Additionally, given the low-resolution nature of the images in this study, this presents a relatively unconventional segmentation problem. Considering this, the results speak to the robustness and versatility of the models derived from the DARTS search space.

Moreover, the high computational cost of NAS, see Table IV, could potentially deter researchers with constrained computing resources. Without access to a computing cluster, this research project would have likely spanned well over a hundred continuous training days on an NVIDIA RTX-3090 GPU. All these considerations should be factored in when deciding whether to employ NAS, further emphasizing the need for a case-by-case approach to the application of this

technology.

D. NAS versus Foundation Models (RQ4)

NAS offers a mechanism to craft models optimized for particular tasks or datasets. This specialization, as our Experiment 4 suggested, can exploit additional image information like Sentinel-2's spectral bands, typically overlooked by broader models like SAM. This ability to tailor architectures to specific problems pushes performance boundaries, provides valuable insights into the nature of tasks, and can lead to efficient models adept at solving unique problems. However, this comes at a substantial computational cost, and the solutions may lack generalizability across diverse tasks.

Conversely, generalized models such as GPT-4 and SAM are designed to perform well across a broad range of tasks within a specific domain. These models leverage large amounts of diverse data, becoming proficient in multiple areas. They offer a holistic approach, handling various tasks without task-specific customization. However, their vast size may not result in the peak task-specific performance achievable by a NAS-generated model. Additionally, their large sizes often translate to high resource requirements and substantial environmental impact, restricting who can train these new networks. Once trained, many of these models become openly available and can be used for various tasks.

The balance between specialized and generalized models will likely continue to shift as technological advances and computational resources evolve. Future research may explore hybrid strategies, blending the customization of NAS with the broad applicability of large-scale generalized models, or new approaches may emerge. The trade-offs between these paradigms suggest potential integration in hybrids. It is plausible that NAS could design future massive generalized models. While large, generalized models have proven proficient, the ability of NAS to tailor architectures to specific problems could refine such models, ensuring efficiency and improving performance.

VI. Conclusion and Future Work

Addressing the global need for renewable energy monitoring, this work introduces Solis-seg, a DNN for solar farm segmentation in Sentinel-2 satellite imagery. Solis-seg has a strong mean Intersection over Union (IoU) of 96.26% on a continental-scale dataset. We also demonstrate the practical application of NAS in semantic segmentation of Sentinel-2 satellite imagery, a largely unexplored domain for NAS. Our results suggest that NAS methodologies, specifically Auto-DeepLab [14], can leverage additional image data, such as spectral bands, offering avenues for creating data-rich models in specialized tasks.

Contrary to popular practice, our results lead us to question the efficacy of transfer learning from classification to semantic segmentation, suggesting that this approach may compromise performance. Our study also emphasizes the need to weigh the benefits of NAS against practical constraints like computational resources, particularly when computing resources are limited. Finally, we contribute an open dataset of New York solar farms, enriching publicly available resources for further research in this field.

The decision of whether or not to use NAS hinges on the importance of incremental performance improvement and the available alternatives to increase the performance of the model. In our case, it might be more productive to allocate resources toward enhancing other aspects of the model, such as augmenting the quality and volume of data [12] or investigating the optimal combination of spectral bands.

Future research could combine our models with Kruitwagen *et al.*'s dataset. This would enable apples-to-apples evaluation of our models in a more expansive and diverse setting. Unfortunately, developing a data pipeline, akin to the one employed by Kruitwagen *et al.*, that integrates their data with one of our trained models, is a substantial undertaking. This is due to the complex nature of these pipelines. This complexity is why we have not tried to perform this integration in our current study. The challenges uncovered in the New York pilot study, discussed in Section IV-F, underscore the importance of diverse training data. The model's struggle to generalize indicates that it could benefit from a more diverse dataset that includes various architectural styles, landscapes, and environmental conditions. Future work on creating and distributing such datasets would be fruitful.

Finally, it is crucial to remember that NAS is a relatively nascent field, despite much progress [6], [29]. As with many emerging technologies, it will likely undergo considerable refinement and become more efficient and accessible in the coming years. Future advancements might mitigate many of the current limitations, enabling more widespread and accessible usage.

Appendix

A. Training Environment and Data

Our experiments were conducted on a Computing Cluster equipped with NVIDIA A100 and V100 GPUs. Some tests also utilized an NVIDIA RTX 3090.

A collection of over 200,000 Sentinel-2 level-2A images, serves as the empirical foundation of our research. We will refer to this dataset as the Solis dataset. Each image is a 224x224 pixel chip with 12 bands, and approximately half are positive examples featuring solar farms. The masks are either hand drawn, sourced from OpenStreetMap,[17] or generated by prior deployments of our Solis-transfer model. To counter potential biases and overfitting, we employed a diverse set of images from various geographical regions. Data augmentation techniques, including random horizontal and vertical flips, were applied to enhance model robustness.

B. Implementation and Parameter Selection

We use Auto-DeepLab (ADL) in experiments, specifically we study the impact of how ADL enhances a cell-centric search space [15] via a hierarchical component to manage spatial resolution during search [19]. In line with DeepLab conventions, the architecture search concludes with

[17] https://wiki.openstreetmap.org

an Atrous Spatial Pyramid Pooling (ASPP) module [1]. However, unlike traditional DeepLab models, ADL utilizes only three branches in the ASPP module instead of the typical five [14].

Our research utilized a PyTorch adaptation of the original ADL model[18] modified to work with our data loaders and with minor enhancements to memory usage, code readability, checkpointing, and model monitoring. This codebase serves as the foundation for all our experiments and is available for public scrutiny. In terms of parameter settings, we followed Liu *et al.* [14], with modifications to suit our specific hardware. For instance, we adjusted the batch sizes to 22 or 12 depending on the available GPU memory.

C. On the Random Model

Auto-DeepLab architectures are represented by two arrays, detailing macro- and micro-architecture, each bound by specific constraints. To illustrate, here's the prompt given to ChatGPT:

> Give me two random arrays that look kind of like this: [0 0 0 1 2 1 2 2 3 3 2 1] [[0 7] [1 4] [2 4] [3 6] [5 4] [8 4] [11 5] [13 5] [17 5] [19 7]] For the first array, the length should be 12, and the numbers have to be in range 0-3, also, the difference between subsequent numbers cannot be larger than 1. For the second array, the right number in each instance is between 0-7; the left side is between 0 and 19.

The constraints for the micro-architecture (the second array) are a bit stricter in reality. Still, after making it retry a few times, ChatGPT generated arrays that, with the modification of just one out of 20 numbers in the micro-architecture array, conformed to these constraints. This encoding system is not described in the original paper [14].

REFERENCES

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE PAMI*, 40(4):834–848, 2017.

[2] W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang. FasterSeg: Searching for faster real-time semantic segmentation. *arXiv preprint arXiv:1912.10917*, 2019.

[3] X. Chen and C.-J. Hsieh. Stabilizing differentiable architecture search via perturbation-based regularization. In *ICML*, pages 1554–1565, 2020.

[4] M. V. C. V. da Costa, O. L. F. de Carvalho, A. G. Orlandi, I. Hirata, A. O. de Albuquerque, F. V. e Silva, R. F. Guimarães, R. A. T. Gomes, and O. A. de Carvalho Júnior. Remote sensing for monitoring photovoltaic solar plants in Brazil using deep semantic segmentation. *Energies*, 14(10), 2021.

[5] X. Dong and Y. Yang. NAS-Bench-201: Extending the scope of reproducible neural architecture search. In *ICLR*, 2020.

[6] T. Elsken, J. H. Metzen, and F. Hutter. Neural architecture search: A survey. *JMLR*, 20(55):1–21, 2019.

[7] S. Garcia-Garcia, A.and Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[8] X. Hou, B. Wang, W. Hu, L. Yin, and H. Wu. SolarNet: a deep learning framework to map solar power plants in China from satellite imagery. *arXiv preprint arXiv:1912.03685*, 2019.

[9] Xin Hou, Biao Wang, Wanqi Hu, Lei Yin, and Haishan Wu. Solarnet: A deep learning framework to map solar power plants in china from satellite imagery, 2019.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023.

[11] L Kruitwagen, K. T Story, J Friedrich, L Byers, S Skillman, and C Hepburn. A global inventory of photovoltaic solar energy generating units. *Nature*, pages 604–610, 10 2021.

[12] C. Layman. Using satellites to track solar farm growth, 2019.

[13] P. Lin, P. Sun, G. Cheng, S. Xie, X. Li, and J. Shi. Graph-guided architecture search for real-time semantic segmentation. In *CVPR*, pages 4203–4212, 2020.

[14] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei. Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation. In *CVPR*, pages 82–92, 2019.

[15] H. Liu, K. Simonyan, and Y. Yang. DARTS: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[16] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan. A survey on evolutionary neural architecture search. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2021.

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[18] G. F. Miller, P. M. Todd, and S. U. Hegde. Designing neural networks using genetic algorithms. In *International Conference on Genetic Algorithms*, 1989.

[19] R. Mohan, T. Elsken, A. Zela, J. H. Metzen, B. Staffler, T. Brox, A. Valada, and F. Hutter. Neural architecture search for dense prediction tasks in computer vision. *International Journal of Computer Vision*, 131(7):1784–1807, 2023.

[20] G. Ochoa and N. Veerapen. Neural architecture search: A visual analysis. In *PPSN XVII*, pages 603–615, Cham, 2022. Springer International Publishing.

[21] E. Olweus. Deep neural network architectures for detection and segmentation of solar farms in satellite imagery. Master's thesis, Norwegian University of Science and Technology (NTNU), 2023.

[22] OpenAI. Gpt-4 technical report, 2023.

[23] V. Plakman, J. Rosier, and J. van Vliet. Solar park detection from publicly available satellite imagery. *GIScience & Remote Sensing*, 59(1):461–480, 2022.

[24] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[25] A. Shaw, D. Hunter, F. Landola, and S. Sidhu. SqueezeNAS: Fast neural architecture search for faster semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*.

[26] M. Själander, M. Jahre, G. Tufte, and N. Reissmann. EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure, 2019.

[27] C. Tao, Y. Meng, J. Li, B. Yang, F. Hu, Y. Li, C. Cui, and W. Chang. MSNet: multispectral semantic segmentation network for remote sensing images. *GIScience & Remote Sensing*, 59(1):1177–1198, 2022.

[28] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3, 5 2016.

[29] C. White, M. Safari, R. Sukthanker, B. Ru, T. Elsken, A. Zela, D. Dey, and F. Hutter. Neural architecture search: Insights from 1000 papers. *arXiv preprint arXiv:2301.08727*, 2023.

[30] D. Xuanyi and Y. Yang. Searching for a robust neural architecture in four GPU hours. In *CVPR*, pages 1761–1770, 2019.

[31] J. Yu, Z. Wang, A. Majumdar, and R. Rajagopal. Deepsolar: A machine learning framework to efficiently construct a solar deployment database in the United States. *Joule*, 2:2605–2617, 2018.

[32] X. Zhang, H. Xu, H. Mo, J. Tan, C. Yang, L. Wang, and W. Ren. DCNAS: Densely connected neural architecture search for semantic image segmentation. In *CVPR*, pages 13956–13967, 2021.

[18] https://github.com/NoamRosenberg/autodeeplab