Solar Panel Detection From Aerial Images Using Transformers

Paola Reyes, Rushi Chaudhari
Northeastern University

1 Introduction

The solar industry has introduced us to a green and clean source of energy, which in fact is on a 42% average growth spree. The adaptation of this clean resource is on the rise, and we intend to use semantic segmentation techniques to understand solar panel coverage from aerial view images. Whereas we propose a fast and efficient method of detecting solar panels in aerial images (taken from satellite, drones, UAV's or any aerial device) using deep learning methods, which will help us study more about the production and coverage of solar panels in an area/city of interest. This data is highly valuable as companies in the solar space (old or new), can use our research to better target customers, urban planners, open new maintenance workshops, run targeted ad-campaigns, plan for company expansion, etc. Traditional methods of detecting solar panels in highly congregated houses or buildings do not perform well, due to objects that closely resemble a solar panel, and other noise that is present.

Following the success of transformers in natural language processing (NLP) and the introduction of vision Transformer (ViT) [10] for image classification, there has been a surge in the use of transformers for vision tasks with great results. For the task of semantic segmentation in particular, visual transformers have achieved great performance on multiple benchmark datasets compared with state of the art CNNs. In this work, we perform a comparative study of the performance of state-of-the-art CNN models and transformer vision models applied to semantic segmentation of solar panel imagery. For CNN, we implement U-Net and DeepLabv3+ based models. We will compare their performance to that of transformer based methods SegFormer and DPT.

The SegFormer framework proposed in [11] combines a hierarchical transformers encoder with a lightweight decoder with only four MLP layers. It has achieved state of the art results on ADE20K. DPT [2] is a new segmentation model released by Intel in 2021. In DPT ViTs are used instead of CNNs giving more consistent and detailed predictions.

2 Related Work

Traditional algorithm approaches were used in [1] on a small dataset of 50 images and supervised classifiers with manual feature extraction. Modern techniques with CNNs have improved a lot over the traditional image recognition approaches on aerial detection problems [3][4]. In [5] the researchers proposed a method to use CNN based PV detector that performed exceptionally well on single PV cells but faced difficulty in recognizing large PV arrays. To overcome this [6], used CNN network SegNet which substantially outperformed the results of [5] for solar PV arrays. In [7] the researchers employed transfer learning of EfficientNet-B7 for solar panel detection and passed it to U-Net for mask prediction. The work in [5] uses a pretrained VGG net classifier with 6 convolutional and 2 fully connected layers. A post processing method was applied to connect pixels with indirect contact. But since VGG being a classification model, it was difficult to acquire exact shape of the solar panels. Many state of the art deep learning models like U-Net, Deeplab v3+, Dilated net, Dilated ResNet, have been proposed in [8].

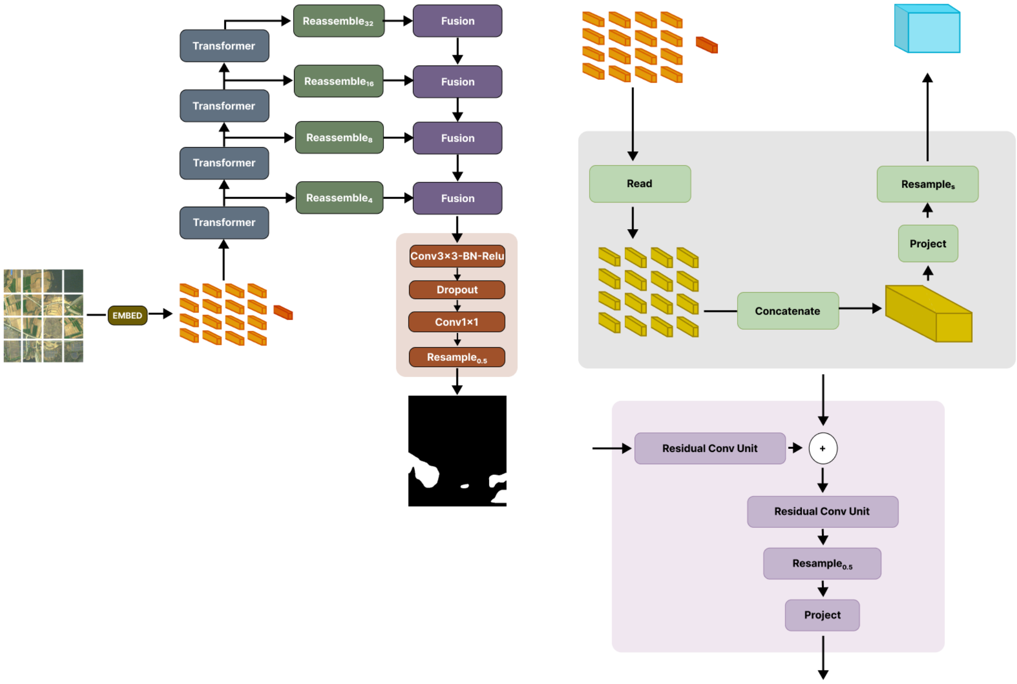


Figure 1: DPT architecture overview

3 Method

To test the performance of transformers for semantic segmentation of Solar PV imagery we performed experiments with two transformer architectures, DPT and SegFormer.

3.1 DPT Transformer

This section shows the experiments performed on Dense Vision transformer (DPT) [2], which leverages a usual encoder-decoder architecture but instead of a CNN, Vision transformers [10] were used as a backbone. In DPT the input image is divided into N overlapping patches of fixed size (16x16). The patches are flattened and turned into embeddings using linear projection. Positional embeddings are concatenated to these embeddings to retain spatial information between tokens. The result of applying applying embedding process to image of shape HxW is N_p tokens $t^0 = \{t^0_0, ..., t^0_{N_p}\}, \quad t^0_n \in \mathbb{R}^D$, where $N_p = H \times W/P^2$ and D is the dimension of each token.

The N_p tokens are passed to Convolution Decoder, where they get assembled at different resolutions which later are progressively fused into final prediction. To assemble the tokens the authors proposed

$$\text{Reassemble}_s^{\hat{D}}(t) = (\text{Resample}_s \circ \text{Concatenate} \circ \text{Read})(t),$$

where s is the recovered representation's output size ratio in terms of the input picture, and D denotes the output feature dimension.

The Read method is used to map the $N_p + 1$ tokens into N_p tokens, since ViT comes from NLP background, it has an extra class token along with the N_p tokens from the image used in classification. Since the class token doesn't make much sense in segmentation, the authors proposed three different Read methods to handle it.

The use of $\text{Read}_{\text{projection}}$ by the authors yielded better performance as compared to using $\text{Read}_{\text{ignore}}$ and Read_{add}. Therefore, we used $\text{Read}_{\text{projection}}$ for our experiments.

$$\begin{aligned}\text{Read}_{proj}(t) = \{&\text{mlp}(\text{cat}(t_1, t_0)), \cdots, \\ &\text{mlp}(\text{cat}(t_{N_p}, t_0))\}\end{aligned} \tag{1}$$

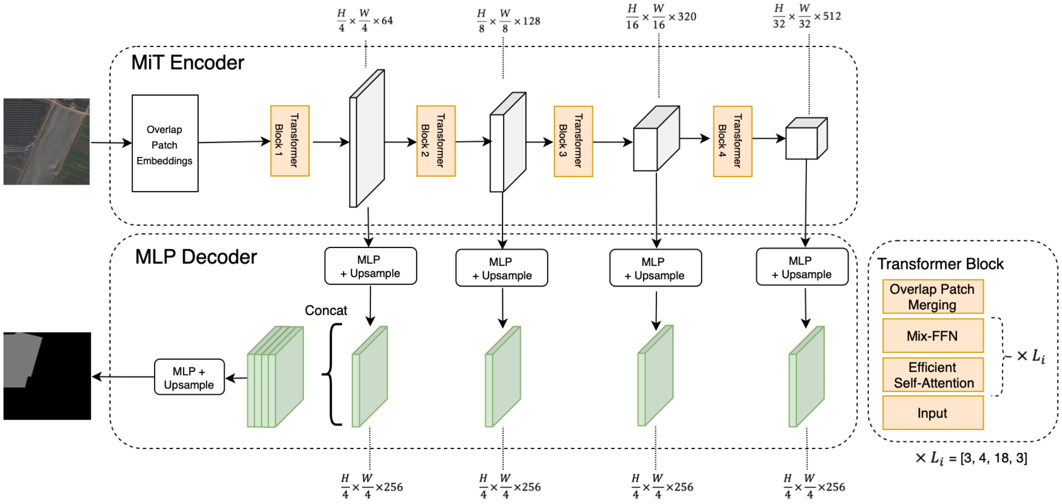

$\frac{H}{4} \times \frac{W}{4} \times 64$
$\frac{H}{8} \times \frac{W}{8} \times 128$
$\frac{H}{16} \times \frac{W}{16} \times 320$
$\frac{H}{32} \times \frac{W}{32} \times 512$
MiT Encoder
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Decoder
MLP + Upsample
MLP + Upsample
MLP + Upsample
MLP + Upsample
Concat
MLP + Upsample
Transformer Block
Overlap Patch Merging
Mix-FFN
Efficient Self-Attention
Input
$\times L_i$
$\times L_i$ = [3, 4, 18, 3]
$\frac{H}{4} \times \frac{W}{4} \times 256$
$\frac{H}{4} \times \frac{W}{4} \times 256$
$\frac{H}{4} \times \frac{W}{4} \times 256$
$\frac{H}{4} \times \frac{W}{4} \times 256$

Figure 2: **Overview of the SegFormer Model.** The encoder consists of four transformer blocks. Each transformer block as multiple layers of Efficient Self Attention and Mix-FFN modules. The All-MLP decoder fuses the multi-level features to produce a segmentation mask. The feature dimensions shown correspond to SegFormer size MiT-B3 which was used.

which concatenates the class token and passes to multi layer perceptron. After the Read block, the N_p tokens are reshaped into image like representations using Concatenate block.

$$\text{Concatenate} : \mathbb{R}^{N_p \times D} \to \mathbb{R}^{\frac{H}{p} \times \frac{W}{p} \times D}. \tag{2}$$

And finally these representations are scaled to size $\frac{H}{s} \times \frac{W}{s}$ in the Resamples block. Resample involves a 1×1 convolution followed by a 3×3 strided convolution/transpose convolution for upsampling/downsampling respectively.

The scaled representations at successive stages are joined using refinenet's fusion algorithm [14], and each fusion stage is upsampled by a factor of two. Figure 1 shows an overview of the Dense Vision transformer architecture.

3.2 SegFormer

Xie et al. [11] proposed the SegFormer transformer model as a variant of ViT optimized for semantic segmentation. The input image is split into 4×4 patches which are then organized into a linear sequence that is used as input to the encoder. Figure 2 shows an overview of the SegFormer architecture.

3.2.1 Mix Transformer Encoder (MiT)

SegFormer introduces a hierarchical transformer encoder which they call Mix Transformer (MiT). It consists of 4 transformer blocks, each of which generates features at a different level of resolution. Given an input image with resolution $H \times W \times 3$, patch merging is performed at each transformer block to obtain a hierarchical feature map with a resolution of $\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i$, where $i \in \{1, 2, 3, 4\}$ is the block number.

A single transformer block consists of two transformer encoder layers followed by overlap patch merging. The transformer encoder layer stacks an Efficient Self-Attention layer followed by a Mix-FFN layer.

Efficient Self-Attention. The original multi-head self attention introduces three heads, Q, K, V, with dimension $N \times C$, where $N = H \times W$ is the length of the sequence and attention is formulated as:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^\top}{\sqrt{d_{head}}})V. \tag{3}$$

Efficient self-attention uses a sequence reduction process [13] to reduce the length of the sequence as follows:

$$\begin{aligned}\hat{K} &= \text{Reshape}(\frac{N}{R}, C \cdot R)(K) \\ K &= \text{Linear}(C \cdot R, C)(\hat{K}),\end{aligned} \tag{4}$$

where R is a reduction ratio, K is the sequence to be reduced. The new K has dimensions $\frac{N}{R} \times C$. This reduces the complexity of the self-attention layer by R.

Mix-FFN. Different from a feed-forward network (FFN), a Mix-FNN introduces a 3 × 3 convolution to encode positional information. The Mix-FFN layer is formulated as:

$$\mathbf{x}_{out} = \text{MLP}(\text{GELU}(\text{Conv}_{3\times3}(\text{ MLP}(\mathbf{x}_{in})))) + \mathbf{x}_{in}, \tag{5}$$

where $\mathbf{x}_{in}$ is the feature from the self-attention module.

3.2.2 MLP Decoder

The SegFormer decoder consists of four main steps which can be formulated as:

$$\begin{aligned}\hat{F_i} &= \text{ReLU}(\text{Norm}(\text{Linear}(C_i, C)))(F_i), \forall i \\ \hat{F_i} &= \text{Upsample}(\frac{W}{4} \times \frac{W}{4})(\hat{F_i}), \forall i \\ F &= \text{ReLU}(\text{Norm}(\text{Linear}(4C, C)))(\text{Concat}(\hat{F_i})), \forall i \\ M &= \text{Linear}(C, N_{cls})(F),\end{aligned} \tag{6}$$

where F_i are the multilevel features from the MiT encoder, $\text{Linear}(C_{in}, C_{out})(\cdot)$ refers to a linear layer with C_{in} input and C_{out} output dimension, and N_{cls} is the number of classes.

3.2.3 Local Emphasis

Following the work in [16], we experimented with changing SegFormer's simple decoder. The main idea is to refocus the attention weights obtained from the encoder feature maps and emphasize local features in the decoder. To this end, we replace the MLP layer that transforms each feature map with a convolution operation and activation function. The new decoder is defined as

$$\begin{aligned}\hat{F_i} &= \text{ReLU}(\text{Norm}(\text{Conv}(C_i, C)))(F_i), \forall i \\ \hat{F_i} &= \text{ReLU}(\text{Norm}(\text{Conv}(C, C)))(F_i), \forall i \\ \hat{F_i} &= \text{Upsample}(\frac{W}{4} \times \frac{W}{4})(\hat{F_i}), \forall i \\ F &= \text{ReLU}(\text{Norm}(\text{Linear}(4C, C)))(\text{Concat}(\hat{F_i})), \forall i \\ M &= \text{Linear}(C, N_{cls})(F).\end{aligned} \tag{7}$$

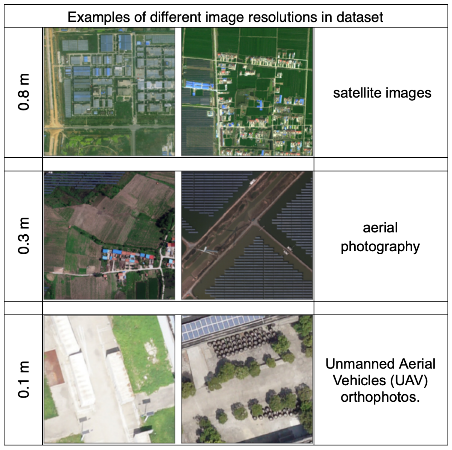
4 Experiments

4.1 Dataset And Evaluation Metrics

All experiments were conducted using the dataset introduced in [8]. This dataset consists of satellite and aerial imagery collected in Jiangsu Province, China. The PV samples were collected at three different spatial resolutions: 0.8m resolution from satellite imagery, 0.3m resolution from aerial photography and 0.1m resolution from Unmanned Aerial Vehicles (UAV) orthophotos. Each training sample was manually annotated to generate segmentation masks to be used as ground truth during training. Figure 3 shows the number of images as well as some sample images in each resolution. We split the dataset into 60% training, 20% validation and 20% test.

We evaluate semantic segmentation performance using mean Intersection over Union (mIoU). As a loss function we use Dice Loss[12].

Mean Intersection over Union (mIoU). For an individual class, Intersection over Union (IoU) measures the percent overlap between the target mask and the predicted output. At the pixel level,


Examples of different image resolutions in dataset
0.8 m
satellite images
0.3 m
aerial photography
0.1 m
Unmanned Aerial Vehicles (UAV) orthophotos.

Spatial Resolution	Number	Size
0.8m	763	1024 x 1024
0.3m	2,308	1024 x 1024
0.1m	645	256 x 256
Total	3,716	

Figure 3: Summary of the PV Dataset

IoU is defined as the number of true positives (TP) divided by the sum of true positives (TP), false positives (FP) and false negatives (FN).

$$IoU = \frac{\textit{Area of Overlap}}{\textit{Area of Union}}$$

$$IoU = \frac{TP}{TP + FP + FN} \tag{8}$$

Dice Loss. Dice Loss is based on the dice coefficient which is a statistic used to measure the similarity between two samples. It was first proposed in [12] and it is defined as:

$$DL = 1 - \frac{2 * \sum p_{true} * p_{pred} + \epsilon}{\sum p_{true}^2 + \sum p_{pred}^2 + \epsilon} \tag{9}$$

where p_{true} is the binary target and p_{pred} is the predicted binary segmentation and ϵ is a small number added to prevent errors in edge cases. Dice loss can be particularly helpful when we have highly imbalanced classes where the background pixels are a lot more prominent than the foreground pixels as is often the case with solar panel segmentation data.

4.2 Implementation Details.

For implementation of SegFormer and DeepLab v3+ we used the *mmsegmentation*[1] codebase. In the training pipeline we used random crop of 512 x 512, random flip, and photometric distortions as data augmentation operations. All three image resolution groups were used together for training. In the case of the 0.1 resolution images that are smaller than 512 x 512, padding is used. For UNet we used the *segmentation_models_pytorch* [2] codebase. The results for DPT of PV01 are trained solely on PV01 instead of all resolutions to compare with the results in [8]. We used the AdamW optimizer and polynomial learning rate policy.

4.3 Baseline CNN Models

We compare the performance of the transformer models to two of the best performing models that have been used on this dataset[8].

UNet. We used a UNet with ResNet50 as the backbone and encoder weights pretrained on ImageNet. Our model has achieved almost comparable results to the best model proposed in the research paper [8]. The authors achieve an IoU score of 84.5 using DeepLab v3+ on the PV08_ Ground data. Our

[1] https://github.com/open-mmlab/mmsegmentation

[2] https://github.com/qubvel/mmsegmentation_models.pytorch

model achieves 92.5 IoU score at 15th epoch using the U-Net Segmentation model trained on the PV08_Ground data. Additionally, we track the dice loss during the training performance to improve the model performance. Our U-Net Segmentation model achieves 0.163 dice loss, and this metric can serve as a baseline for other researchers.

Table 1: Overall test results

Model	mIoU	Loss
U-Net	86.24	0.16
DeepLabv3+	92.7	0.17
DPT	92.1	0.06 *
SegFormer	93.9	0.16

Table 2: Results by spatial resolution

Model	0.8m	0.3m	0.1m
DeepLabv3+	88.4	93.2	90.6
DPT	88.03	93.3	**95.1**
SegFormer	**92.2**	**94.1**	94.9

* *Dice loss did not perform well on DPT so cross entropy loss was used.*

** *U-Net does not perform well on lower resolution images and is omitted in the comparison by resolution.*

DeepLab v3+. The best results that have been previously reported on the dataset[8] we obtained using DeepLab v3+. We used a ResNet-101 as the backbone with dilation set to (1, 1, 1, 2) and strides set to (1, 2, 2, 1). The decoder is a depthwise separable ASPP head with dilations set to (1, 6, 12, 18). We initialize with weights pretrained on Cityscapes[17] and train for 2000 iterations. The model achieves dice loss of 0.18 and overall mIoU of 92.9. As shown in table 2, there is significant difference in performance between different resolutions. DeepLab v3+ performs much better on 0.3m resolution images which were more abundant in the dataset. The 0.8m resolution group gets the lowest mIoU of 88.4 which are the images with the highest resolution where the pixel areas with PV panels is much smaller than the pixel areas without them. This suggests that, for this dataset, DeepLab v3+ may be biased toward the background class and it is not able to accurately detect the finer details in high resolution images. This problem is evident in the qualitative results shown in figure 4 where DeepLab v3+ works really well for the 0.1m and 0.3m resolution images, but struggles to properly identify the PV panel in the 0.8m resolution image.

4.4 Transformer Models Implementation and Results

DPT. We used ViT-Base as the transformers in DPT which gave similar results compared to the baseline models. We used layers 2, 5, 8, 11 to reassemble the tokens and a patch size of 16. We attained an mIoU of 92.14 and a Cross entropy loss of 0.06. DiceLoss, like other models, didn't achieve positive results with DPT. Interestingly, fine-tuning with a train-validation-test split of 5% 5% 90% gave nearly similar results when compared with other models. The transformer encoder does a great job in capturing fine-grain details compared to the CNN models in [8].

SegFormer. We used the MiT-B3 version of SegFormer where the number of layers for each transformer stage is set to (3,4,18,3). We set the channel dimension of the decoder to 256. We do pretrained weight initialization using Cityscapes and train for 2000 iterations. SegFormer performs best out of all models considered achieving mIoU of 93.9 and dice loss of 17. It also performs exceptionally well for each individual image resolution group as shown in table 2. This is evidence that the hierarchical transformer encoder does a great job of capturing relevant features at different levels of resolution, especially when compared to the results of DeepLab v3+.

SegFormer + Local Emphasis. We tested SegFormer with several variations of the Local Emphasis decoder. We tried same convolutions with (K=3,P=1,S=1) and (K=5,P=2,S=1). We also experimented with the stepwise feature aggregation strategy proposed in [16] where, instead of concatenating all four feature maps at once, we first concatenate two of them and combine the result with each remaining feature map one at a time applying a linear layer in between each add. We did not find significant improvement in performance using Local Emphasis. We obtained the a best mIoU of 94.0 using two (K=3,P=1,S=1) convolutions which is virtually the same at the original SegFormer performance of 93.9.

We find that transformers can capture much fine details from features because of the multi-head attention. Table 1 compares the performance metrics for all the models. Since the data was biased

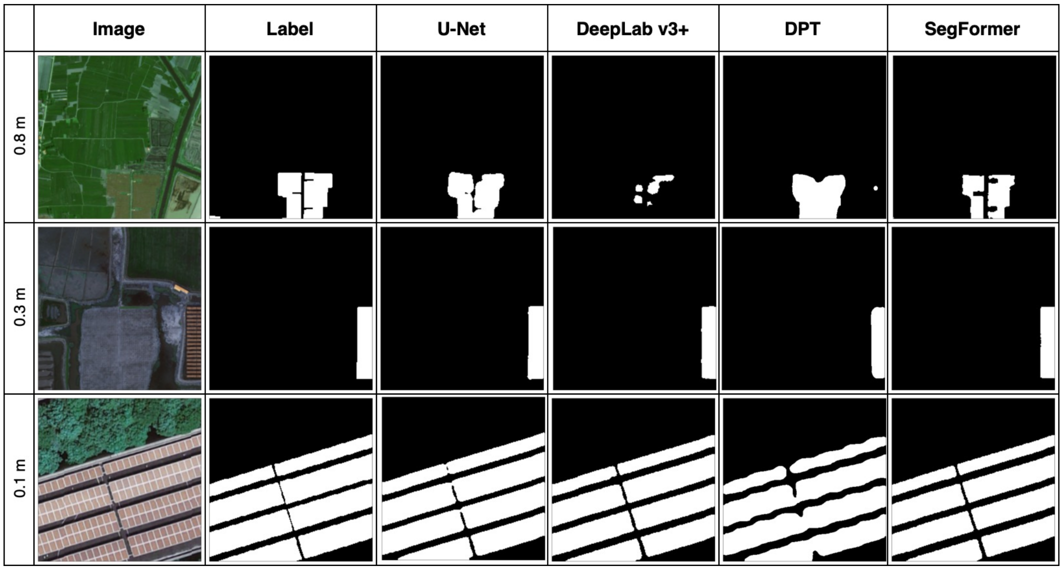


Figure 4: Qualitative results for each model in each spatial resolution.

towards PV03, in 2 U-Net didn't perform well on lower resolution datasets like PV-01, but the transformers based models achieved an mIoU of 95.104 and 94.9 (DPT and Segformer respectively).

5 Conclusion

Based on the dataset, we investigated the performance of transformer models compared to state of the art CNN models for solar panel segmentation. Out of the models Segformer had a significantly higher mIoU. Our experiments showcase SegFormer's ability to capture multi-level features that proved useful when dealing with the multi-resolution dataset. UNet and DeepLabV3+ performed acceptably. DeepLab v3+ performed better for lower resolution images. Considering the speed, CNN models would be of more practical use if the image is not of lower resolution. Thus urban planners/ government can use these techniques by feeding in the satellite images and conducting surveys to improve solar panel campaigns reducing the reliance on fossil fuels and combating climate change. Further research could include ways to boost the performance of transformer based models and apply such techniques to similar questions of interest.

References

[1] J. M. Malof, R. Hou, L. M. Collins, K. Bradbury, and R. Newell, “Automatic solar photovoltaic panel detection in satellite imagery,” in International Conference on Renewable Energy Research and Applications (ICRERA), 2015, pp. 1428–1431.

[2] ReneRanftl, AlexeyBochkovskiy, and VladlenKoltun. Visiontransformers for dense prediction. In ICCV, pages 12179–12188, October 2021.

[3] I. Sevo and A. Avramovic, “Convolutional Neural Network Based Automatic Object Detection on Aerial Images,” IEEE Geosci. Remote Sens. Lett., vol. 13, no. 5, pp. 740– 744, 2016.

[4] K. Nogueira, O. a B. Penatti, and J. a Dos Santos, “Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification,” arXiv Prepr. arXiv1602.01517, 2016

[5] J. M. Malof, L. M. Collins, and K. Bradbury, “A deep convolutional neural network, with pre-training, for solar photovoltaic array detection in aerial imagery,” Int. Geosci. Remote Sens. Symp., vol. 2017-July, pp. 874–877, 2017, doi: 10.1109/IGARSS.2017.8127092.

[6] Camilo, J.; Wang, R.; Collins, L.M.; Bradbury, K.; Malof, J.M. Application of a Semantic Segmentation Convolutional Neural Network for Accurate Automatic Detection and Mapping of Solar Photovoltaic Arrays in Aerial Imagery. arXiv 2018, arXiv:1801.04018.

[7] Parhar, P., Sawasaki, R., Todeschini, A., Vahabi, H., Nusaputra, N. and Vergara, F., 2022. HyperionSolarNet: Solar Panel Detection from Aerial Images. arXiv preprint arXiv:2201.02107.

[8] Jiang, H.; Yao, L.; Lu, N.; Qin, J.; Liu, T.; Liu, Y.; Zhou, C. Multi-resolution dataset for photovoltaic panel segmentation from satellite and aerial imagery. Earth Syst. Sci. Data 2021, 13, 5389–5401.

[9] Wani MA, Mujtaba T. Segmentation of Satellite Images of Solar Panels Using Fast Deep Learning Model. International Journal of Renewable Energy Research (IJRER). 2021 Mar 30;11(1):31-45.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv, 2020.

[11] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” arXiv:2105.15203, 2021.

[12] Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." In 2016 fourth international conference on 3D vision (3DV), pp. 565-571. IEEE, 2016.

[13] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv, 2021. 2, 3, 4.

[14] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In CVPR, 2017.

[15] Chen, Liang-Chieh, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation." In Proceedings of the European conference on computer vision (ECCV), pp. 801-818. 2018.

[16] Wang, Jinfeng, Qiming Huang, Feilong Tang, Jia Meng, Jionglong Su, and Sifan Song. "Stepwise Feature Fusion: Local Guides Global." arXiv preprint arXiv:2203.03635 (2022).

[17] Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. "The cityscapes dataset." In CVPR Workshop on the Future of Datasets in Vision, vol. 2. sn, 2015.

[18] Wang, Jinfeng, Qiming Huang, Feilong Tang, Jia Meng, Jionglong Su, and Sifan Song. "Stepwise Feature Fusion: Local Guides Global." arXiv preprint arXiv:2203.03635 (2022).