UTRECHT UNIVERSITY

Department of Information and Computing Science

Computing Science & Artificial Intelligence

Master Thesis: Classification and Segmentation of photovoltaic and solar thermal systems from aerial imagery

Utrecht University Supervisors:
Itir Önal Ertuğrul
Mang Ning

Second examiner:
Ronald Poppe

Candidate:
Rienk Fidder

In cooperation with:
We-Boost

We-Boost Supervisors:
Marjolein Hordijk
Ward de Hond

Abstract

Photovoltaic (PV) and Solar Thermal (ST) panels mounted on rooftops form a cornerstone in the transition to fully renewable energy generation. However, due to the large gap in data on the number and location of these panels, policymakers have trouble determining the effectiveness of policies and energy network administrators have trouble building efficient networks. In this study, a model is proposed to automatically classify and segment PV and ST panels from aerial imagery to alleviate this issue. A novel dataset of aerial images in the Netherlands, containing image-level and pixel-level annotations of PV and ST panels is presented and made publicly available. A two-stage pipeline consisting of a classification and segmentation stage is proposed, as well as a novel method for weakly-supervised pseudo-label generation based on greedy Class Activation Map (CAM) refinement and Segment Anything Model (SAM) generated segmentations. The model is shown to exhibit strong classification performance, after finetuning models pretrained either on ImageNet or Dutch aerial images. Performance of fully-, semi-, and weakly-supervised segmentation models is evaluated. It is shown that the best performance is achieved by combining a small set of manually annotated mask labels with a larger set of unlabelled data in a semi-supervised manner. This semi-supervised approach leads to an IoU of 73.3% for binary segmentation, and a class-specific IoU of 77.0% and 37.6% is achieved for the PV and ST classes respectively.

Acknowledgements

During the course of this thesis, I have been fortunate to receive the support and guidance of many people, to whom I extend my deepest gratitude. First of all, I would like to thank my direct supervisors Dr. Itir Önal Ertuğrul and MSc. Mang Ning for their guidance, feedback, and support throughout the project, their expertise and critical evaluation have been invaluable. I would also like to thank my second examiner Dr. Ronald Poppe for his time and effort in evaluating this thesis, as well as his immensely useful feedback early on in the project. Furthermore, I would like to thank all my colleagues at We-Boost who welcomed me into their team with open arms and great enthusiasm. Especially, I would like to thank Marjolein Hordijk and Ward de Hond for supervising me during this project. Their knowledge in this field helped me to great extent, and their creativity and interest in the project have been inspiring.

List of acronyms

- **BAG** - Basisregistratie Adressen en Gebouwen (Basic Registration of Addresses and Buildings)
- **CAM** - Class Activation Map
- **CV** - Computer Vision
- **CNN** - Convolutional Neural Network
- **FCN** - Fully Convolutional Network
- **FN** - False Negatives
- **FP** - False Positives
- **FPN** - Feature Pyramid Network
- **IoU** - Intersection over Union
- **lr** - learning rate
- **MAE** - Masked Auto-Encoder
- **mIoU** - mean Intersection over Union
- **PDOK** - Publieke Dienstverlening Op de Kaart (Public Service on the Map)
- **PV** - Photovoltaic
- **SAM** - Segment Anything Model
- **ST** - Solar Thermal
- **TN** - True Negatives
- **TP** - True Positives
- **ViT** - Vision Transformer
- **WSSS** - Weakly Supervised Semantic Segmentation

Contents

1 Introduction **6**

2 Research Questions **10**

2.1 Sub-questions 10

3 Literature review **12**

3.1 Image classification 13

3.2 PV Classification 19

3.3 Image Segmentation 24

3.4 PV Segmentation 32

3.5 Datasets 40

3.6 Gaps in research 42

4 Data **43**

4.1 PDOK Aerial Imagery 43

4.2 BAG 44

4.3 BAG Refinement 45

4.4 Image level labels 46

4.5 Pixel level labels 47

4.6 Unlabelled samples 47

4.7 Dataset overview and comparison 47

5 Background **49**

5.1 Class Activation Maps 49

5.2 ConvNeXt V2 49

5.3 DeepLabV3+ 53

5.4 CorrMatch 55

5.5 Metrics 57

6 Methodology **59**

6.1 Model architecture 59

6.2 Self-supervised Pretraining 59

6.3 Classification 60

6.4 Segmentation 63

7 Results **69**
7.1 Pretraining 69
7.2 Finetuning 71
7.3 CAM Refinement 72
7.4 Segmentation 74

8 Discussion **79**
8.1 Findings 79
8.2 Revisiting the research questions 86
8.3 Limitations and further research 89
8.4 Conclusion 90

Bibliography **91**

1. Introduction

According to a recent report by the United Nations Framework Convention on Climate Change (FCCC), the world is not on track to meet the goals set during the 2015 Paris agreement [1]. Much more effort needs to be done to reduce emissions, by individuals, but especially by governing bodies. Policymakers need to decide on the most effective policies to introduce in order to combat climate change, but designing these policies can be a daunting and uncertain task. Models used to predict the effects of policies can be inaccurate, and data used to monitor the effects of policies that have been implemented is often incomplete or lacking entirely.

In 2022, about 15% of all energy consumed in the Netherlands originated from renewable sources [2]. 3.34% of all energy consumed (about 22% of all renewable energy) was produced from solar energy, a 45% increase from the year before. This increase can predominantly be attributed to the increasing instalment rate of residential photovoltaic (PV) panels, as 80% of all solar panels in the Netherlands are found on rooftops or car parking shades. Policies aim to stimulate the further development of rooftop instalments, but insight in the exact number of installations and date of instalment are not always available. Stedin estimated that 25% of residential PV systems in 2018 were not registered, despite the fact that registration is required by law, although not enforced [3].

In addition to PV panels, solar thermal (ST) panels are also gaining popularity and are increasingly contributing to the renewable energy production. The European Solar Thermal Industry Federation (ESTIF) reported that in 2021, the installed capacity of ST grew by 10.2% to a total of 624 143 square meters, capable of producing 436 900 kW per hour [4]. Despite the increasing utility of these systems, hardly any registries are being kept of the installations, or they are not differentiated from PV panels in solar panel registries. Because of this, it is also hard for municipalities to steer the development of ST implementation in their area further.

The impact of this gap in data on installed panels is twofold. For one, it makes it hard for policymakers in governing bodies to determine to what extent recent policy changes, such as reducing tax on the sale of PV systems, has contributed to

the national PV power generating capacity. Secondly, it makes it harder for energy network administrators to build the most efficient networks to transport and store the energy which is produced. As an example, if the power output generated by PV systems of a residential block is higher than the network surrounding it can transport and store (because it was built based on records underestimating the expected output), the generated energy has nowhere to go and the PV systems will generate less or no energy to compensate. This essentially wastes clean energy, and it is therefore vital to close this gap in PV system registrations as soon as possible.

However, getting this information manually is cumbersome and error-prone. Surveys targeting homeowners can be time-consuming and rely on residents voluntarily providing information, while visual inspections on a large scale are simply infeasible due to the amount of manual labour that would be required. Another problem with these approaches is the fact that PV systems are installed in an increasing rate or might be taken down when they are faulty or have reached the end of their lifetime. Thus existing records require frequent updates to stay up to date. This problem is noticeable in the few governmental datasets that are available. Doing manual surveys or visual inspections can yield a complete registration at a specific point in time, but if registrations are not updated strictly, which up until this point they have not, then these datasets will go out of date quickly. Therefore, one would need to do these inspections intermittently, which increases the amount of effort required even further. Finally, it is also almost impossible to do this kind of data gathering in retrospect, to gain insight in the evolution of the data over time. There is therefore need for an efficient, objective, accurate, and repeatable method which can also be performed to gather data from the past.

Recently, researchers have started to tackle this problem by utilising computer vision (CV) techniques to analyse satellite and aerial images. As CV algorithms, models, hardware, and datasets have improved, they have become increasingly accurate and efficient at recognizing a large variety of objects from image inputs. Especially in the field of Deep Learning for CV, models have been developed trained to recognize objects in images with high accuracy. While still not perfect, these models can also be trained to recognize PV systems in aerial images, an example of the output of such a model can be found in Figure 1.1. Building such an automated system also comes with the advantage of being able to process historic data, as well as novel data efficiently. This approach still has some challenges however, as the models need to be trained on large volumes of data, the labelling of

which requires manual labour. This problem could be alleviated by using semi-, or weakly-supervised learning techniques, which require less manually annotated data, at the cost of potentially lower performance.

Figure 1.1: Example output of a PV segmentation algorithm, taken from [5].

To contribute to this field, this thesis aims to develop a model for accurate classification and segmentation of PV and ST panels. The model should be able to segment PV and ST panels across the entirety of the Netherlands by utilising Dutch aerial imagery. To this end, a novel dataset for training purposes will be proposed, and a method for building and training such a model will be presented. Different variations to the individual components of the architecture will be proposed, and experiments are done to verify the performance of these variations. The results of these experiments will be presented, and the implications of these results will be discussed. The contributions of this thesis can be summarized as follows:

- A novel manually labelled dataset of aerial images in the Netherlands, containing image-level and pixel-level annotations of PV and ST panels is presented and made publicly available.
- A two-stage pipeline for PV and ST panel classification and segmentation is proposed.
- An attempt is made to improve model performance by including building registration data in the form of BAG (Basisregistraties Adressen en Gebouwen) polygons as binary masks.
- A novel method for weakly-supervised pseudo-label generation based on greedy CAM refinement and SAM generated segmentations is proposed, as well as evaluated by training a model on the generated pseudo-labels.

- Application of semi-supervised learning is explored, both in the case of manually annotated data, as well as for weakly-supervised pseudo-labels by partitioning the set of pseudo-labels based on a proposed confidence metric.

This thesis project was performed in cooperation with We-Boost [6]. We-Boost is a Dutch consultancy firm located in Utrecht specialising in tenders, sustainability, and data. To support Dutch municipalities in their transition to 100% renewable energy usage, they aim to provide insight in current solar energy production in these municipalities as well as the growth of PV installations in recent years. The main aim of this research is therefore to produce a model which is as accurate as possible at detecting the total area of photovoltaic and solar thermal panels in a given area.

The structure of the remainder of this thesis is as follows: in Chapter 2, the research questions will be presented. Chapter 3 presents an overview of relevant literature, as well as the gaps that exists within this literature. Next, the dataset created and used in this thesis will be presented in Chapter 4. Background information on the models and techniques used is given in Chapter 5, followed by the methodology used in the experiments in Chapter 6. Results are then presented in Chapter 7, and the thesis is concluded by a discussion of these results in Chapter 8.

2. Research Questions

This thesis aims to answer the following research question:
To what extent can photovoltaic and solar thermal systems be segmented from Dutch aerial imagery?

2.1 Sub-questions

In order to answer this research question, four sub-questions are identified, these sub-questions are formulated as follows:

1. ***To what extent can building location data be utilised in addition to RGB channels to improve the performance of a PV and ST detection model?***
   When training the model for classification of PV and ST panels, two variations will be experimented with. A model utilising only RGB channels as input will be compared to a model that utilises a fourth binary channel with building location information. Performance of the two models will be compared to answer this question.

2. ***What is the effect of self-supervised pretraining on a large domain-specific dataset on the performance of a PV and ST detection model?***
   Encoder weights pretrained on ImageNet are publicly available for the ConvNextV2 model. Performance of a model pretrained utilising only images from the target domain of this study (aerial images) will be compared to the ImageNet pretrained model. The aim of this research question is to determine whether the abundance of unlabelled data in the target domain can be utilised to improve the performance of the model.

3. ***To what extent can photovoltaic and solar thermal systems be distinguished by a machine learning model?***
   A model for classification or segmentation can be trained to perform either binary or multi-label classification and segmentation. One can expect the performance of the binary variant to be higher than the multi-label variant, as determining the presence of either type of solar panel is an easier task than also reporting the distinction. The size of this gap in performance for the cur-

rent problem domain will be investigated by training models to do binary, as well as multi-label, classification and segmentation of solar panels.

4. ***What is the performance impact of choosing a semi-supervised or weakly-supervised segmentation approach over a fully-supervised approach for PV and ST segmentation?***
   While fully supervised segmentation almost exclusively produces better results than semi-supervised or weakly-supervised approaches, data for the latter two approaches is much less labour-intensive to obtain. A method of pseudo label generation based solely on image level labels by utilising class activation map refinement and state-of-the-art methods such as Segment Anything [7] will be proposed. The performance of this weakly supervised method will be compared to a fully supervised approach. Additionally, the performance of a fully supervised segmentation model compared to a semi-supervised approach utilising the same set of labelled image in addition to a larger set of unlabelled images will be examined.

In the following chapter, a detailed review of current literature is given, and the gaps in literature are identified that these questions aim to fill.

3. Literature review

In recent literature, there has been a spike of interest in detecting PV systems via remote sensing. The rising popularity in many remote sensing fields can be attributed to the rapid progress that has been made in the field of computer vision (CV). This progress has, in part, been enabled by the improvements in hardware and neural network quality. CV tasks such as image classification - learning the appropriate class label for an image - and segmentation - learning to segment a relevant section of an image - can be specialised for the task of PV and ST panel detection. In this chapter, an in-depth overview of advances in the field of CV is given – notably in image classification and segmentation – as well as the applications of these novel techniques in PV classification and segmentation.

Similar literature reviews have been published to highlight advances in the field, such as the work of Mao et al. [8]. They gave a comprehensive analysis of frequently used data sources and compared the performance of a large variety of methods, spanning object-based, pixel based, and deep learning methods. They concluded that, overall, deep learning methods provide the best performance for decentralized systems such as rooftop installations, whereas object-based methods sometimes outperform deep learning method when it comes to detecting centralized systems. However, studies do seem to suggest that detecting centralized systems is a slightly simpler task than detecting decentralized systems, as the centralized systems are often much larger in size than e.g. rooftop mounted decentralized systems, and easier to distinguish from their background setting.

Feng et al. [9] employed a text mining approach to gather a large sum of articles in the domain of PV systems and machine learning, and discussed the most relevant articles from the outcome of this process. They found that AI is often employed for solar forecasting, PV fault detection, and PV array detection, the latter of which is often based on deep learning. They identified that the scarcity of data in the field of PV detection compared to solar forecasting was a large reason for the relatively lower interest in PV detection research compared to other areas, but also noted that the application value of PV detection is often underestimated, and that further exploration of its uses might boost research interest.

Highlighting the motivation behind this research as well as many of those discussed in this section, De Hoog et al. [10] identified the stakeholders that would benefit from automatic PV detection from satellite and aerial imagery, and discussed approaches up to that point while also identifying research gaps. It was noted that one of the main challenges in the field is training a model not to confuse non-PV systems with PV systems, such as ST systems, greenhouses, roof windows, or pools. Moreover, ensuring a model recalls all the PV systems in an image is challenging. Factors such as shading, panel colour, or panel size can make systems difficult to recognize even for human annotators. Finally, the consistency of training data is considered to be a large hurdle, since many aerial images are taken at varying tilt angles or times of day. They conclude with an overview of research opportunities for PV detection systems, such as improving model accuracy, but also integrating PV detection systems with solar forecasting systems and existing registrations.

The remainder of this literature review is structured as follows: first, an overview of advancements in image classification is given, mainly by discussing convolutional neural networks and vision transformers. Then the applications of these and other methods to PV classification are discussed. Next, novel methods for image segmentation are reviewed, including fully-, weakly-, and semi-supervised approaches. This will again be followed by an exploration of PV segmentation studies. Additionally, frequently utilised datasets in this particular research field are discussed as a point of comparison for the dataset introduced later in this thesis. The chapter is concluded with an assessment of the gaps in existing literature, and an illustration of how these gaps are addressed by the proposed research questions

3.1 Image classification

The field of CV consists of a large subset of problems, of which image classification is likely the most studied area. While originally many classification algorithms relied on hand-crafted features to train a machine learning algorithm, most recent breakthroughs in classification and other CV problems have been made based on the field of deep learning. The shift towards deep learning has also influenced the field of PV detection greatly, improving the PV detection rates significantly compared to early shallow methods. This section aims to give a brief historical

overview of deep learning application in image classification, and discuss the architectures that were both influential in the general classification task, as well as in the task of PV classification. Additionally, the recent introduction of vision transformers is discussed, as well as how they inspired the development of ConvNeXt and ConvNeXtV2, the latter of which is the backbone of the proposed method in this thesis.

3.1.1 Convolutional neural networks

A broad range of different deep learning network architectures have been proposed. However, for many tasks in CV, convolutional neural networks (CNN) have proven to be one of the most successful. They gained popularity in the 1990s when Lecun et al. [11] presented LeNet-5 and coined the term convolutional neural network. The idea behind CNNs is to use convolutional layers to extract features from an image, and then use fully connected layers to classify the image based on these features. A convolution can be seen as a (learnable) matrix that is slid over the image, where the dot product of the matrix and the image is computed at each position to form a feature map for the next layer. LeNet-5 was built for handwritten digit recognition and consisted of 2 convolutional layers, 2 pooling layers, and two fully connected layers. Although their performance was limited due to hardware constraints, the authors were already able to reach an error rate as low as 0.7% on the handwritten digit recognition task.

In 2010 the first ImageNet large scale visual recognition challenge (ILSVRC) was hosted [12]. This challenge often serves as a benchmark for classification performance, and winners of the challenge are commonly considered as the state-of-the-art solution for the respective year. It only took 2 years for a CNN to win the competition, when Krizhevsky et al. [13] improved the top-5 error rate of the previous year from 25.8% to 16.4% with their submission named AlexNet. AlexNet consists of 8 layers, of which 5 are convolutional and 3 are fully connected. From that point on, every subsequent edition of the challenge that was held has been won by a CNN.

Two years later, the challenge was won by Szegedy et al. [14] of Google with GoogLeNet, an incarnation of their newly proposed Inception architecture. The Inception architecture did not use any fully connected layers, but was instead comprised of many Inception modules in sequence. These Inception modules were devised based on a neuropsychological theory describing the firing mechanics of

neurons in the brain. The Inception module was designed to increase depth and width without significant increases in computation cost. The network had an approximately 10% lower top-5 error rate than AlexNet. One year later, Szegedy et al. [15] revisited their architecture. After discussing some general design principles that aid the performance of CNNs, they proposed a series of modified Inception modules based on these principles. An InceptionV2 network with multiple variants was proposed based on these new modules and experimented with. The highest performing variation was labelled InceptionV3 and decreased the top-5 error rate on the ImageNet challenge set by another percentage compared to the original Inception network. An overview of the InceptionV3 architecture can be found in Figure 3.1.

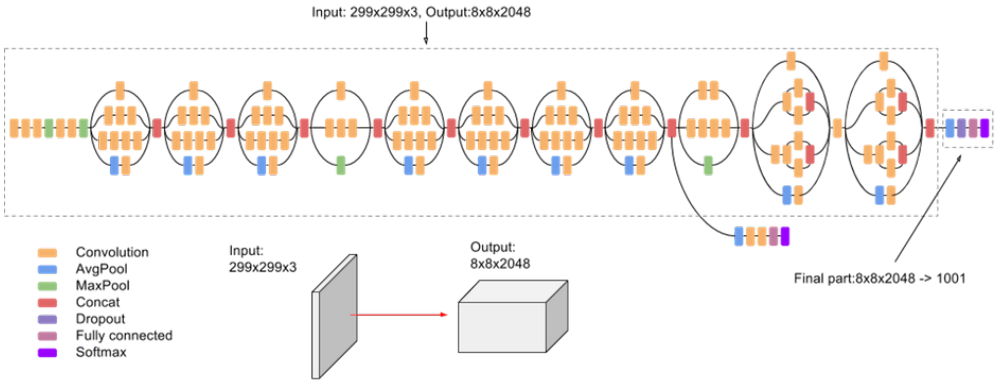

Figure 3.1: InceptionV3 architecture [15]. Image taken from [16]

Another noteworthy winner of the ImageNet challenge was developed by He et al. [17] from Microsoft research group. The network, called ResNet, successfully mitigated the problem of vanishing and exploding gradient, which often becomes a problem as network depth increases, by introducing (identity) shortcut connections that allow the network to retain information over many sequential layers. This allowed the authors to build a network which was 152 layers deep and yet less complex than competing algorithms of its time.

In many CNNs, images are transformed into multiple scales to form an image pyramid. Features are then extracted from each image to create a feature pyramid, of which each layer produces a prediction. With the introduction of these methods, it became easier to classify objects over different scales, but extracting the features from each level in the pyramid can be slow. Alternatively, it is possible - and faster - to build a feature pyramid from a single image, though this decreases the accuracy of the method. To strike a balance between these options, Lin et al. [18] proposed a

method to create a Pyramid Network (FPN) for classification from a single image. Their FPN extracts features for each layer, and then performs upsampling on the resulting feature maps. Each feature map in the upsampled reverse pyramid is then used for prediction (Figure 3.2). With this method, the authors were able to surpass winners of the - at the time recent - COCO 2016 object classification challenge [19].

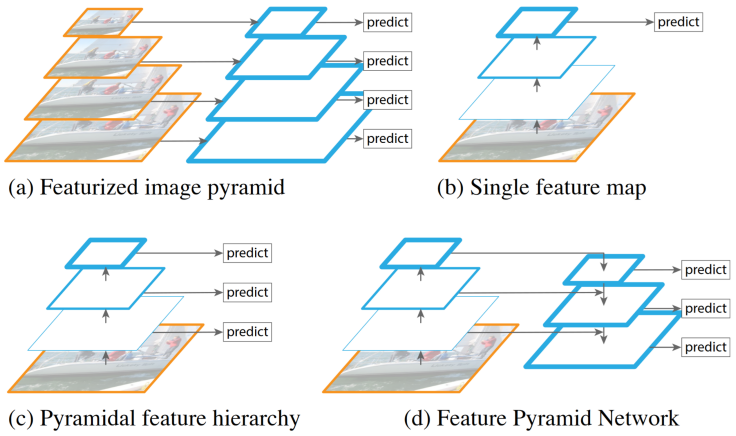

Figure 3.2: Feature Pyramid Networks compared to other approaches. Image taken from [18]

3.1.2 Vision transformers

In 2017, Vaswani et al. [20] revolutionized the field of Natural Language Processing (NLP) with their paper "Attention is all you need". In this paper, they introduced the transformer network architecture, which got rid of the recurrence and convolutions that dominated many state-of-the-art networks in the field. Instead, their architecture relied almost fully on attention mechanisms, using positional embeddings to keep track of the positions of words in a sentence. In short, incorporating the attention mechanism in an architecture allows the network to learn which parts of the full input are important given the currently processed part of the input. Thus, by using attention, a large volume of context can be utilised at each step, while discarding the noisy parts of the input. This created an architecture where the importance of every word in a sentence is learned based on every other word in the sentence, and its own position in the sentence. The novel architecture outperformed other models of its time on translation tasks with only a fraction of the training time and swiftly rose in popularity.

Inspired by the success of Transformers in the NLP domain, Dosovitskiy et al. [21] attempted to apply the transformer architecture to the image domain. But as

the attention mechanism of transformers attempts to assess the importance of each input vector compared to every other input vector, inputting pixels individually would not be computationally feasible. To overcome this issue, the authors opted to divide the input images into patches and embed these patches into input vectors, combined with a positional embedding (Figure 3.3). A key difference between the proposed Vision Transformers (ViT) and CNNs is that ViTs do not make use of the image-specific inductive bias that CNNs have, as the convolution step in a CNN inherently captures locality, two-dimensional neighbourhood structure and translational equivariance by design. While on a smaller dataset this seemed to be a large advantage of CNNs, making them more performant than ViTs, ViTs seem to be able to overcome this disadvantage by processing large enough amounts of data. On larger image datasets, ViTs outperform state-of-the-art CNN architectures, even with a lower training cost. Finally, promise was shown for ViTs to be pretrained in a self-supervised manner utilising masked patch prediction, before finetuning on a specific domain, while achieving a high performance.

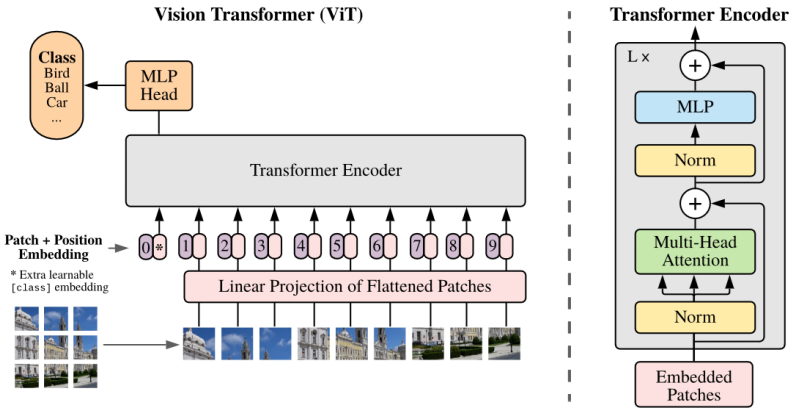

Figure 3.3: ViT model overview [21].

ViTs are able to handle extremely large volumes of data, but manually labelling these large volumes of data can be a labour intensive task. A solution to this problem is to use self-supervised learning methods as a pretraining step, for example by utilising Masked Auto-Encoders (MAE) which are often used to train large language models. He et al. [22] presented MAE as a scalable method of training a ViT in a self-supervised manner. Their method relied on masking a subset of the input patches and training a decoder to reconstruct the image as closely to the original image as possible. The authors showed that vision transformers are able to perform this task well with masking ratios as high as 90%, although a masking ratio of

75% was deemed most suitable for general use cases. An overview of the training strategy and results are visualised in Figure 3.4. After pretraining, the decoder of the vision transformer used for reconstructing the image is discarded. Experiments were then done on a variety of image tasks such as classification, object detection, and segmentation. In all tasks, MAE is shown to be a more effective pretraining method than the current state of the art supervised pretraining approaches.

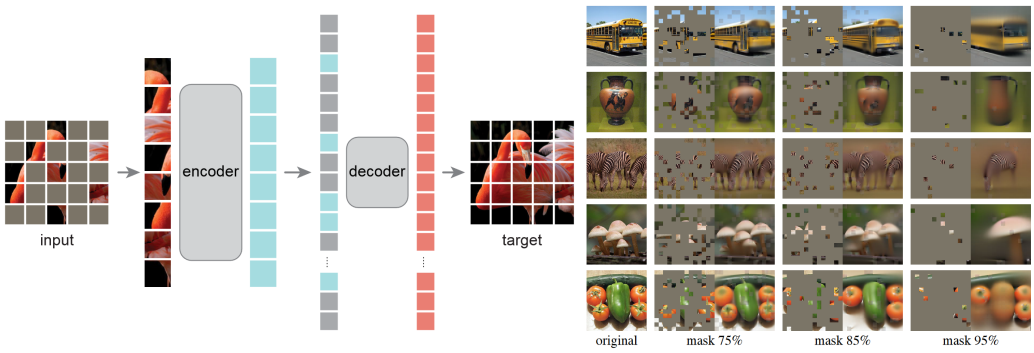

Figure 3.4: MAE training overview and results [22].

3.1.3 A return to CNNs

Multiple extensions to the ViT architecture were proposed in the years following its introduction. One of the most well known of these extensions are Swin transformers [24] by Ze Liu et al., which introduced a hierarchical architecture computed by shifting windows. With the dominant performance of ViTs over CNNs of the time, it seemed that CNNs would soon be replaced fully by ViTs. However, Zhuang Liu et al. [23] showed it was possible to design a CNN architecture based on principles proposed in the ViT research that actually outperforms ViTs. They started by adjusting the training procedure to be close to that of Swin Transformers, and already noted a 2.7% increase in performance of a Resnet-50 model on ImageNet. They then examined other design choices from Swin Transformers, and adjusted ResNet based on these principles. An overview of these choices can be found in Figure 3.5, they will be discussed in further detail in Chapter 5. By combining these principles, that were already researched individually, the authors were able to create a CNN that outperforms ViTs on ImageNet with an accuracy of 82.0% compared to 81.3% for Swin Transformers. This family of models was dubbed ConvNeXt.

Applying another lesson learned from transformers, Woo et al. [25] later applied the concept of Masked Auto-Encoders to ConvNeXt. However, to effectively integrate MAE pretraining into the ConvNeXt architecture, it was necessary to ad-

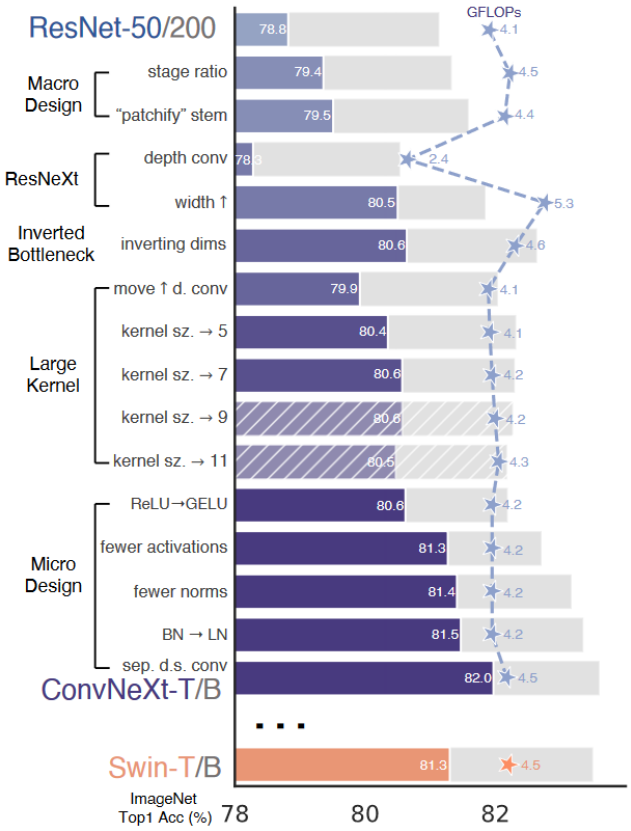

Figure 3.5: Design choices made in the creation of ConvNeXt, and the corresponding performance impact for each choice [23].

just the framework to be fully convolutional. To achieve this, the authors made use of sparse convolutions, such that the 2D image structure is preserved even when a large part of it is masked. The resulting framework, dubbed ConvNeXt V2, scored 84.6% accuracy on ImageNet, outperforming the original ConvNeXt architecture trained only in a supervised manner by 0.8%. The paper highlights the importance of a network architecture designed in parallel with the training strategy. More details on this architecture can also be found in Chapter 5.

3.2 PV Classification

In this section, the application of the aforementioned architectures to the task of PV classification is discussed. A few studies however were published before the era of CNN dominance in image classification, which will be noted first. Then, studies utilising CNNs for PV classification are discussed.

3.2.1 Classical machine learning

One of the earliest works that explored automatic detection of PV systems from satellite imagery using machine learning was carried out by Malof et al. [26]. The proposed algorithm consisted of two steps, a prescreener and a feature processor. The prescreener first converts a candidate image to greyscale and identifies the maximally stable extremal regions. From these regions, a handcrafted set of features was then extracted and used to train a Support Vector Machine (SVM). Finally, overlapping regions that are classified as positive were merged. This approach already showed promising results with a recall of 94%, although the testing set was relatively small, only containing 54 solar panels. Given the low resolution of satellite imagery, however, the research showed potential in automatic detection of PV systems.
A master thesis comparing different classifier architectures found that while SVMs could numerically get accuracies almost identical to those of other approaches, visual inspection showed the classifier was not performing well [27].

In a follow-up to their original paper, Malof et al. [28] continued work on PV detection, this time however from aerial imagery and with the use of an RF classifier. They used a dataset gathered by Bradbury et al. [29] which was created using aerial images, which have a higher resolution than satellite images and are thus expected to yield better results. First, local colour statistic features were extracted from the images. These were then fed to the RF classifier, which outputted a confidence value per pixel indicating the probability of that pixel representing a part of a PV system. This confidence map was then post-processed to extract the most probable regions, which were used for object detection. While this system already did pixel-level classification of PV arrays, the performance was not satisfactory, thus the authors opted to measure accuracy on an object level. Malof et al. [30] later applied an RF classifier again for the task of PV detection, and compared it to a Convolutional Neural Network. This research will be discussed in more detail as part of the next subsection.

Another example of RFs being used for PV detection is the work of Xia et al. [31]. The focus of this work was to detect water photovoltaic (WPV) systems, which are generally large plants either stationary above or floating on the water surface. Pixel-wise training was done similarly to the work of Malof et al. and again post-processing was done to remove the noise that is often generated by pixel-wise clas-

sification methods. Data from multiple years was also used in post-processing to improve classification confidence on systems found in older images that were also detected on more recent images before. The resulting model, which showed an accuracy of 94.2%, was then used to do a historic analysis of WPV system development in China. The authors found that they detected an increase in PV area from 2016 to 2019 of 33.4 square km to 165.0 square km.

Zhang et al. [32] explored the impact of textural features on the training and performance of an RF classifier for PV detection, including a Grey Level Co-occurrence matrix, reflectance, thermal spectral data, and several environmental indexes. They showed that including these features as parameters for the RF to train on can improve an already very performant algorithm even further. While this information might not always be available, the study shows that in many studies based on only visible light there might still be room for improvement if more data becomes available. The authors do not test their findings on other, more popular, classifiers such as CNNs however, leaving it unclear if they would benefit similarly from texture information.

Plakman et al. [33] also trained an RF classifier using not only spectral but also backscatter data from Sentinel-1 and Sentinel-2 satellites. They showed that by first segmenting the data with Simple Non-Iterative Clustering (SNIC) and then classifying the resulting regions an RF classifier is able to learn to predict areas to not be PV systems with near perfect precision and recall, and predict regions that are PV systems with precision and recall of 92.39% and 81.39% respectively. A big advantage of the method is that it requires relatively little labelled training data, however since it works on satellite data with spatial resolutions of 10 to 20 meters it is only really suitable for detection of industrial and commercial PV plants, and not for rooftop PV systems.

3.2.2 Deep learning

As CNNs started to dominate the field of CV, the PV detection field started adapting the best performing algorithms as well. A wide array of studies have been published utilising different networks, most of which showed dominant performance compared to other machine learning methods.

Malof et al. [30] published a new study comparing the use of a random forest classifier to a CNN, as a follow-up to their work on PV detection using a random

forest classifier [28]. utilising a random forest as prescreener, the CNN was used to improve the precision of the model. The model was further improved shortly after, [34] by using only a CNN based mostly on VGG modules [35]. This new CNN model showed very promising results considering the relatively small size of the network, most notably in its precision. They were also one of the first to experiment with transfer learning in the field of PV detection, but due to the simple nature of the network it proved more effective to train from random weights. Later works incorporating larger networks would show that transfer learning can however be quite effective for the given task. Golovko et al. [36] similarly proposed a simple CNN network for PV classification, but utilised lower resolution satellite imagery.

Yu et al. [37] applied InceptionV3 [15] to the task of PV detection in the DeepSolar project. Training was done on a large dataset gathered from over 50 cities spread over the U.S. which was manually labelled at image level. The trained model achieved an accuracy and recall of 93.7% and 90.5% respectively on images in non-residential areas, and only slightly worse performance in residential areas. The trained network was then used to process the entirety of the U.S. and the data collected from this procedure was used to do a demographic study on the deployment of PV systems in various locations in the U.S. The same architecture was also applied by Ioannou and Myronidis [38] for binary PV classification on a Greek satellite image dataset, highlighting the effectiveness of the architecture on different domains.

The original DeepSolar pipeline was later extended by Wang et al. [39] in their DeepSolar++ project. Similarly to the WPV detection project by Xia et al. [31], the pipeline was extended to include information from matching images taken at different periods of time to improve classification performance. The motivation behind this improvement is the increase in image quality that occurred as technology improved, making it easier to detect PV systems. Two branches of the same architecture are trained, one taking high resolution images, and the other taking low resolution images. By comparing the feature maps of the two branches a higher accuracy can be achieved. The InceptionV3 network used in the previous study is also replaced with the newer ResNet architecture [17]. The model was tested by feeding a test set of image sequences with one location per sequence, for which the model was able to predict the correct installation year of the solar panels with an accuracy of 85.9%. The high resolution branch on its own showed a sensitivity of 97.6% and a specificity of 98.5%, while the low resolution branch showed a sensi-

tivity of 91.2% and a specificity of 95.6% applied on a low resolution image with a high resolution reference image.

Mayer et al. [40] showed the DeepSolar project also has application outside the U.S. by developing DeepSolar for Germany. In this work, they improved the original performance of DeepSolar even further, most prominently by adopting a novel dataset creation strategy. In this strategy, training images are divided into categories relevant to the task problem. A training dataset is created which is heterogeneous with respect to these categories, although urban and rural settings are overrepresented, considering that is where the majority of rooftop PV systems are located. Because of this strategy, the dataset required for a better performance is able to be much smaller than the original labelled dataset, decreasing the required manual annotation labour. The authors compare performance of the original DeepSolar U.S. model on this dataset compared to a model trained on this dataset and show that performance is improved by almost 19 percentage points when using Cohen's k as a performance metric. This shows that training a model on a dataset containing images from the target domain improves performance significantly.

As an extension to the DeepSolar for Germany project, Mayer et al. [41] also published a new project called 3D-PV-Locator. The main addition in this pipeline was the use of 3D spatial data processing to also report azimuth and tilt angles of PV systems. This aids the estimation of energy output of the systems which is useful for tasks such as grid planning. Open datasets containing LiDAR-based point clouds were used to do the estimations and the results were compared with official registries. The azimuth angles between the model output and official registries fell within the same or next closest class in 88% of all tested cases. Tilt angle estimation showed an accuracy of 64%, although the angle estimation is done categorically (tilt $\in [0°; 20°), [20°; 40°), [40°; 60°)$).

The most recent addition to the DeepSolar family of projects was made by Lindahl et al. [42]. They created a dataset guided by official Swedish registries and compared the performance of the model trained on this dataset with the results from earlier studies, although performance decreased compared to other DeepSolar projects. This might have been caused by the very low representation of positive samples in the dataset (0.09%), but the authors also give an analysis of false negatives. They show that a substantial portion of the undetected systems were frameless modules placed on a black rooftop. This indicates that the network most

likely learned to recognise the frames of the PV systems, as well as their contrast to the background rooftop. The authors also train the model to recognise ST systems, but are not able to train the model to distinguish them from PV since there was not enough training data to learn from.

With the wide variety of architectures that are available in CV, it might become difficult to select the most effective one for the task of PV classification. The Dutch CBS [43] therefore conducted a study comparing multiple architectures. In the study, labelled "Deep Solaris", they compared InceptionV3, InceptionResNetV2, DenseNet, and Xception, and found Xception to be the best performing architecture. Furthermore, they found that full transfer learning yielded the best results, yet do not elaborate exactly what is meant by full transfer learning.

Han van Leeuwen [44] later conducted a follow-up study called "DeepSoLim", in which he mainly compared InceptionResNet and VGG16 on images of the Dutch province of Limburg. Additionally, a variety of scenarios testing the impact of factors such as noise, sample size, and class imbalance on the performance of the network were examined. It was found that training on a training set with as little as 353 images can yield and accuracy of 90.5%, while training on the full dataset containing 17669 images improved accuracy to 98.1%. It was also found that transfer learning has a positive impact, while data augmentation did very little to improve performance, or even degraded it in some cases.

3.3 Image Segmentation

While classifying an image to determine if an object is present or not on its own is a useful tool, more precise recognition is often required. In many cases the separation of classes is desired at pixel-level, this pixel-level classification task is referred to as image segmentation. In this section, some of the most influential methods in this field, which are also often utilised in PV segmentation studies, are discussed.

The first of these networks often utilised is Mask R-CNN [45]. Although Mask R-CNN is a model for image segmentation, it is built as an extension of a framework developed for object detection. Object detection is defined as drawing a bounding box around a target object. A family of CNNs named Region-based Convolutional Neural Networks (R-CNN) was developed by Girshick et al. [46]–[48] to solve this problem. The main idea behind the object detection R-CNN networks was to do an initial pass over the image to generate regions of interest, and then

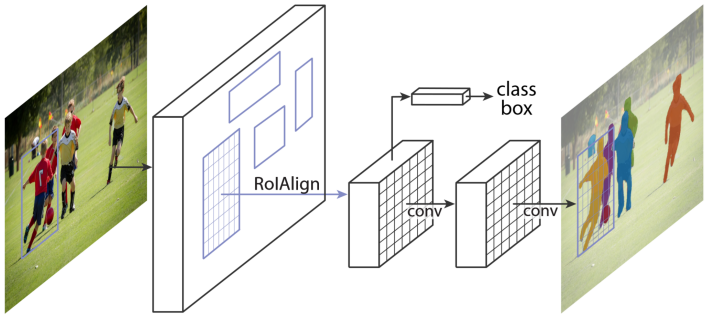

Figure 3.6: Mask R-CNN framework [45].

pass these regions to a CNN classifier to determine whether they actually contain a target object. Improving upon this object detection architecture by adding segmentation output, Mask R-CNN was developed [45]. Mask R-CNN simply extends the last R-CNN iteration for object detection, Faster R-CNN [48], by adding a new branch parallel to the bounding box prediction branch. This new branch was responsible for mask prediction, see Figure 3.6. This means that classification is still independent of mask extraction, which was in contrast to other image segmentation approaches at the time. Performance was not impacted much by this new branch, as the algorithm is able to segment images at 5 frames per second.

Two other influential methods in this field, which are also applied for PV detection are U-Net by Ronneberger et al. [49], and EMANet by Li et al. [50]. While originally developed as a network for segmentation of biomedical images, U-Nets have since been used for many generic segmentation tasks. U-Net is a CNN that can be divided into two distinct paths, a contracting path that reduces the feature dimensions, and an expanding path that increases the dimensions of the features. Intermediate features from the contracting path are also fed forward directly to the corresponding layer in the expanding path, similar to how feature pyramid networks operate. The final layer of the expanding path has the same dimensions as the original image and is used to predict a segmentation mask. EMANet on the other hand was built by extending the well-known Expectation-Maximization (EM) algorithm with an attention component, which was shown to be useful in image segmentation. Li et al. used a variation of the EM algorithm to generate attention maps for an input image, which were used to predict segmentation masks. The algorithm is embedded in a reusable Expectation-Maximization Attention (EMA) unit, so it can be used inside various neural networks. This unit is then used on top

of a ResNet encoder, to create EMANet.

Chen et al. [51]–[53] utilised atrous (also known as dilated) convolutions, Conditional Random Fields, and later an encoder-decoder design to build DeepLab. Atrous convolutions work similar to regular convolutions, but allow a kernel to work over positions that are not strictly neighbouring. The latest iteration of the architecture, DeepLabV3+, set the new state-of-the-art performance on the PASCAL VOC 2012 and Cityscapes datasets, with a mIoU of 90.0% and 82.1% respectively. This was achieved by extending the DeepLabV3 architecture with a decoder module. More details on the DeepLab family of architectures will be given in Chapter 5.

In recent years, approaches for prompted segmentation tasks have also been created. That is, using for example a click location, textual or image input as a prompt, a network is trained to find the corresponding object segmentation. One of such approaches is Segment anything by Kirillov et al. [7] of Meta AI Research. The Segment Anything Model (SAM) consists of a ViT image encoder pretrained by a MAE as described in [22], a prompt encoder that encodes points, boxes, text and masks, and a transformer mask decoder block. The decoder outputs three different potential image masks to deal with ambiguity that can arise when using prompts. By using a grid of input prompts per image, the authors then use their model to create SA-1B, a dataset containing 11 million images and over a billion masks (Illustrated in Figure 3.7). The dataset quality was verified by randomly selecting images and asking human annotator to improve the segmentations. Of the improved images, 94% still had an IoU with the original image greater than 90%.

3.3.1 Weakly supervised semantic segmentation

While CNNs and ViTs have been shown to very capable to the task of image segmentation, they often require a large amount of pixel-level labelled data to perform well. This data is labour-intensive to obtain, forming a bottleneck in progress for many domains. Similarly, data collection can be a large hurdle in the domain of solar panel detection. A possible remedy to this problem is to make use of Weakly Supervised Semantic Segmentation (WSSS) techniques. In WSSS, the aim is to train a model to perform image segmentation using only image-level labels. Class Activation Maps (CAM) of a CNN trained on image level classification are often utilised to localize objects, and a refinement method is introduced to improve the quality of

Figure 3.7: SAM generated segmentations [7].

the segmentation. These refined masks can then be used as pseudo labels to train a segmentation model in a (fully-) supervised manner.

One of such methods of refining CAMs was introduced by Xie et al. [54] who showed the importance of utilising the correct activation function for class-specific feature activation. Their method, called ReCAM relied on first training a regular multi-label classifier, and utilising the CAMs from this trained network to create class-specific feature maps. By then training a new network head to classify these feature maps in a multi-class classifier (predicting only one class at a time), the authors show that the resulting CAMs from this network are more precise in separating classes. Figure 3.8 depicts the pipeline utilised in this method, the authors further note that the resulting pseudo masks can be further refined by methods such as IRN [55] or AdvCAM [56].

In a similar work, Li et al. [57] showed it is also possible to perform CAM refinement without training a network twice. By utilising the CNN-Transformer hybrid Conformer network, they successfully combined CAMs from the CNN branch with attention maps from different layers of the transformer branch. The approach is motivated by the ability of CAMs to precisely locate an object through activation, and the global receptive field inherent to the attention mechanism of transformers. Attention maps at different levels of the transformer branch are shown to capture different similarities, such as textural similarities at a low level, and semantic similarities at a higher level. Pseudo labels are generated by first training a network on a classification task, and then refining the resulting CAMs by doing a matrix multiplication with the average of all attention maps. An overview of the full pipeline,

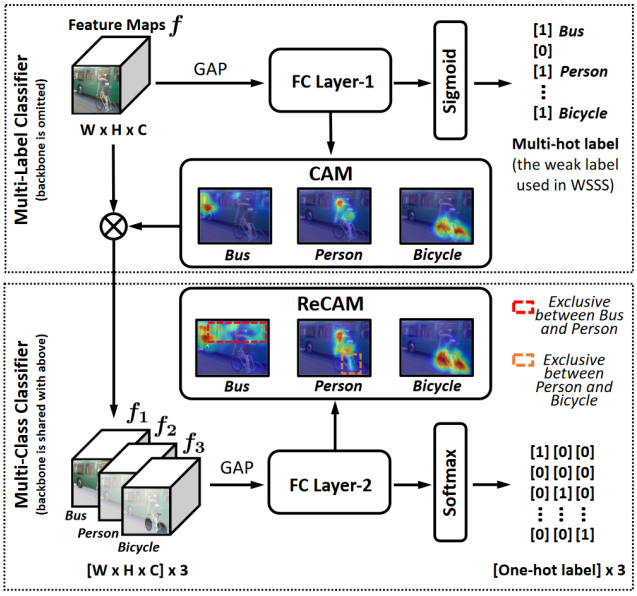

Figure 3.8: ReCAM pipeline [54].

named TransCAM, can be found in Figure 3.9.

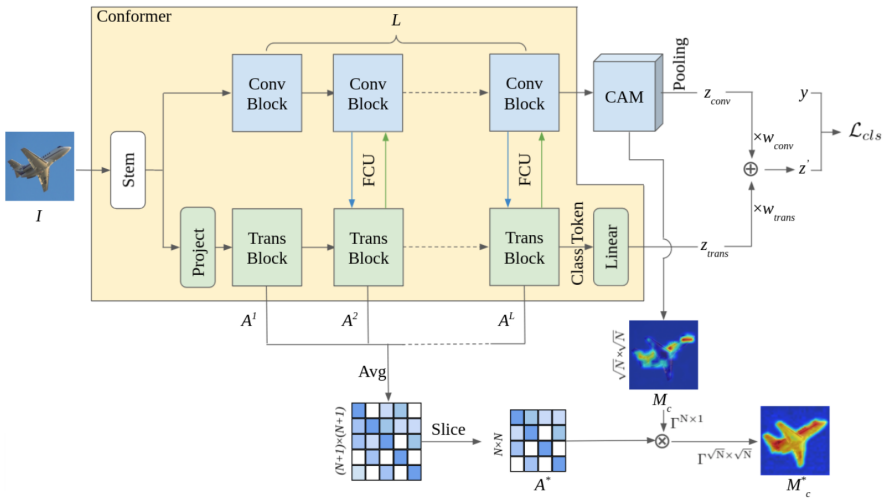

Figure 3.9: TransCAM pipeline [54].

Recently, interest has grown in the use of SAM [7] in this refinement process. By utilising CAMs to localize target objects, SAM can be used to generate class-agnostic segmentation masks to be used as pseudo labels. A study by Jiang et al. [58] showed that SAM can be prompted utilising different forms of weak image labels including image level labels, point labels, scribble labels, and bounding box labels. For image level labels, a trained classification network was used to extract CAMs, from which high confidence points were sampled. These points were then

used as an input prompt to SAM to generate pixel level pseudo labels.

Alternatively, Chen et al. [59] used SAM to generate all possible masks for an image first by utilising the point grid prompting method. They then used the CAMs from a trained classification network to assign masks to different classes based on the overlap between each mask and the CAMs per class. The assigned masks where then filtered based on the total overlap between the CAM and the mask, where only masks with a high enough overlap, and thus high confidence, were kept as pseudo labels to train a segmentation network, the full pipeline can be found in Figure 3.10.

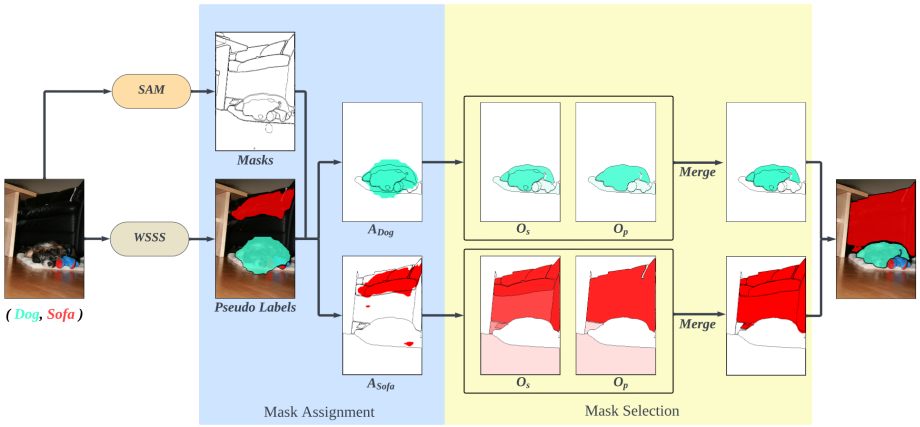

Figure 3.10: SAM mask selection pipeline by [59].

3.3.2 Semi-supervised semantic segmentation

Instead of either requiring the full dataset to be annotated for segmentation, or only requiring image-level annotations, it might also be effective to utilise a large set of unlabelled data paired with a small set of labelled data. This idea is known as semi-supervised semantic segmentation and heavily relies on designing a method of training that can effectively utilise the unlabelled data to improve the segmentation model trained on the labelled data.

The two most common approaches in this field are entropy minimization and consistency regularization. In entropy minimization, a teacher network is trained on the labelled data to generate pseudo labels for the unlabelled data. The student network is then trained on the labelled data and the pseudo labels generated by the teacher network, after which it is again used as a teacher for a new student network. By iterating this process the quality of the pseudo labels increases and the student network becomes more accurate over time.

Consistency regularization is based on the idea that the predicted label for a given input should be invariant to any input perturbation. In these approaches, a loss function is usually designed to contain the loss on the unperturbed labelled data, as well as the loss on the perturbed unlabelled data.

FixMatch by Sohn et al. [60], one of the most well known consistency regularization approaches, showed how a very simple implementation of this approach can already yield promising results. The proposed approach revolved around a simple loss function combining standard cross-entropy loss on the labelled data, and novel loss function on unsupervised samples. The loss function for unlabelled data is as follows:

$$l_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \leq \tau) H(\hat{q}, p_m(y|A(u_b))) \tag{3.1}$$

Here, μB is a batch of unlabelled images, q_b is the prediction on a weakly perturbed image, H is cross entropy, $\hat{q}$ is the prediction on a strongly perturbed image, and τ is a threshold used to determine which pseudo labels to use. Only pseudo labels are used where the confidence of the prediction of the weakly perturbed image is above a set threshold. The total loss is computed by summing the supervised loss and the unsupervised loss weighted with a scaling factor. A benefit of this approach is that the network first trains almost fully on the labelled images, as the confidence on unlabelled image predictions is low, and starts utilising the unlabelled images only when this becomes useful. Simple flip and shift operations were used for weak augmentation, and two specialised augmentation algorithms (RandAugment and CTAugment) were used for strong augmentation. The full approach is visualised in Figure 3.11. While FixMatch was designed and tested on the task of image classification, the approach could easily be transformed to be applied on image segmentation.

Yang et al. [61] revisited FixMatch, and showed that its relatively simple approach of supervising the strongly perturbed image prediction by a weakly perturbed image prediction can already yield state-of-the-art results on the task of image segmentation. High confidence predictions are now computed at a pixel level, allowing for loss to be computed only over the high confidence regions of the pre-

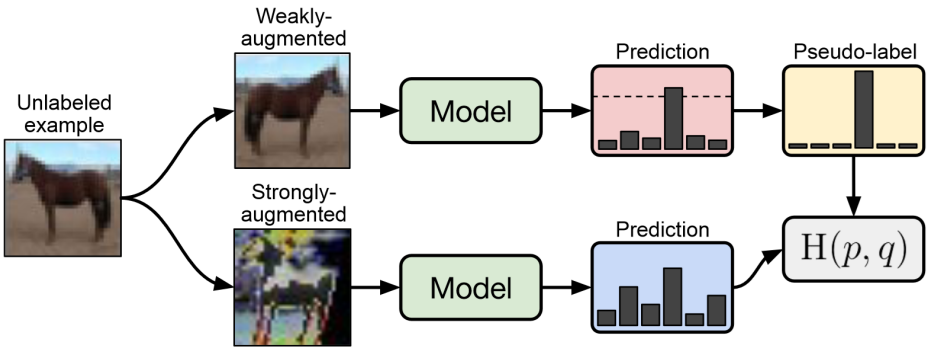

Figure 3.11: FixMatch architecture [60].

dicted segmentation mask. However, they note that domain-specific knowledge might be essential in many cases to reach the best performance on the segmentation task. To mitigate this, the authors propose two modifications to the FixMatch algorithm. The first of these is an additional perturbation stream on the weakly perturbed image. After weak perturbation of the image, the encoded features are passed to the decoder in one branch, and perturbed again in a second branch before also passing through to the decoder. Additionally, the authors note that recent works have shown the benefit of multiple strong perturbation streams, and propose to utilise two streams of strong perturbations. The prediction on the weakly perturbed image without feature perturbation is then used as a pseudo label to compute an unsupervised loss on the remaining three labels predicted on the other perturbed images. A comparison this approach and FixMatch can be found in Figure 3.12. A combination of the two adjustments to FixMatch, named Unimatch, significantly outperformed existing state-of-the-art methods on the PASCAL VOC 2012, COCO, and Cityscapes datasets.

Recently, Sun et al. [62] proposed a novel consistency regularization framework heavily relying on correlation maps to relate pairs of locations across weakly and strongly augmented images. Their approach also builds upon FixMatch [60] by enhancing pseudo labels with correlation maps, and introduces a novel correlation loss. Details of the implementation are discussed in chapter 5. The authors report a new semi-supervised state-of-the-art performance on the Pascal VOC 2012 dataset on every common partition of labelled and unlabelled images.

3.3.3 Segmentation refinement

Segmentation networks are often trained on relatively low-resolution datasets, and might not be able to translate this to precise segmentation of high resolution im-

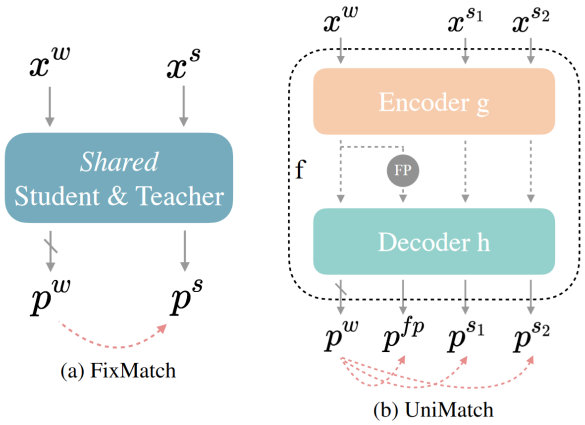

Figure 3.12: FixMatch compared to Unimatch [61]. Here, FP denotes feature perturbation, and the red arrows denote computation of cross entropy loss between predictions.

ages. To combat this issue, Cheng et al. [63] proposed CascadePSP: a refinement model taking an image and low resolution segmentation as input, and producing a high resolution segmentation as output. The model is based on a trained refinement module, consisting of PSPNet with ResNet as the backbone. These refinement modules are then used both on a global level, to repair structure of the whole image, as well as a local step which refines multiple image crops at a more detailed level. The authors show that the model is able to improve the IoU performance of DeepLab V3+ by 2.5% on the PASCAL VOC test set.

3.4 PV Segmentation

PV segmentation studies almost exclusively show a great reliance on CNNs. Many of the studies use network architectures proven to be successful on the general image segmentation task such as Mask R-CNN [45] or U-Net [49]. Additionally, the majority of works on PV segmentation make use of a strongly labelled dataset and fully supervised training, which will be discussed in this section. Initial exploration of weakly supervised segmentation approaches to PV segmentation has been carried out in a handful of studies, which will also be discussed. Finally, some works highlighting the usefulness of transfer learning in this domain are explored, as well as a some studies that are unique in their approach to PV segmentation.

3.4.1 Fully supervised PV segmentation

One of the earliest attempts to do segmentation of PV systems was made by Yuan et al. [64]. They used a basic CNN with an integration stage to do pixel-wise prediction of PV systems, after training on a manually labelled dataset covering 5 cities in the U.S. and containing around 5,000 PV systems. Though the algorithm produces segmentation maps, the authors do not compute IoU scores. By taking the centre of the produced segmentation maps, however, they are able to report precision and recall scores of 85.5% and 87.3% respectively.

Improving upon [34], Camilo et al. [65] replaced the VGG-based network used for segmentation with a SegNet-based implementation. By utilising a network specifically designed for segmentation, they showed it was possible to attain a significant performance improvement. After training on a subset of the Duke California dataset [29], both pixel-based and object-based detection performance was measured. For both of these performance measurements, the SegNet-based approach showed a substantial improvement in the precision-recall curve compared to the VGG-based approach, although the maximum accuracy is not reported.

Malof et al. [66] have also done further work on semantic segmentation and presented their improvements in a project titled SolarMapper. While not specifying the exact architecture of the network, they present IoU scores of 66%-69% and apply the model to a new location as a case study. Before applying the model directly, the model is finetuned on a small, hand-labelled training set taken from the new area in Connecticut. With this finetuned model, the entire state of Connecticut is processed, and the results are used to make an energy capacity estimation. The authors find that, comparing their estimation with a manually collected energy capacity dataset for Connecticut, they achieve a Pearson correlation coefficient of 0.91.

Mask R-CNN

Fully Convolutional Networks (FCN) such as U-Net and Mask R-CNN were shown in multiple studies to be a viable method for PV segmentation, for example by Sizkouhi et al. [67]. They trained a Mask R-CNN network with a VGG16 network as backbone, pretrained on ImageNet. The model was trained on the "Amir" dataset, a large and heterogeneous dataset comprised of images from 12 countries across the globe. The authors report an accuracy of 96.93%, although it is unclear and unlikely that this accuracy is computed at pixel-level. This might be due to the

fact that the aim of the study was to do boundary extraction of large-scale PV installations, such that route planning of autonomous aerial monitoring robots could be optimized. For this task, pixel level precision is not required, and rough estimates of the boundaries should provide enough data to plan routes with.

Creating a larger and more precise dataset is in almost all cases beneficial to the eventual performance of the model, but time and resource constraints often make it difficult to create such large datasets. To combat this issue, Li et al. [68] used a combination of data augmentation and data generation to increase the amount of varied training they could do on the model. They first used public map APIs to segment only the rooftops of houses from satellite data, then used a variation of data augmentation methods such as flipping, rotating, adding noise, and increasing brightness to extend the dataset. Besides that, they leveraged a Deep Convolutional Generative Adversarial Network to learn from the segmented samples they created and generate more rooftops to train on. See Figure 3.13 for examples of the generated sampled. These samples were then used to train a Mask R-CNN model to segment PV system arrays. Results show that while only making a small difference (0.003 improvement of Matthews Correlation Coefficient), data augmentation and generation can improve the accuracy of a model.

Figure 3.13: Samples of training images generated by the Deep Convolutional Generative Adversarial Network trained by Li et al. [68]

While segmentation performance can be improved by utilising more sophisticated segmentation networks or improving the quality of the dataset it is trained on, one could also add a domain-specific post-processing step. This is what Liang et al. [69] did, as they employed a right-angle polygon fitting algorithm after segmenting using a Mask R-CNN segmentation network. The segmentation network was trained using the test set from DeepSolar, extended with more manually labelled mask annotations. The combination of a more specialized segmentation algorithm and a post-processing step that ensures the output matches the shapes of PV systems allowed the authors to improve the precision and recall of the segmen-

tation algorithm to 96% and 95% respectively.

Thus far, all the techniques that have been discussed have been aimed at binary segmentation of PV systems. Schulz et al. [70] however, worked on a more general tool that could segment 6 different types of renewable energy systems from aerial imagery. The model they presented, called DetEEktor, was based on Mask R-CNN and was able to detect 63 to 75% of the systems, dependent on the type of plant. They showed it was possible to distinguish PV and ST systems, even if they have a similar appearance. Despite the dataset only consisting for 0.55% and 4.8% out of biomass plants and wind power plants respectively, the system was still able to detect these types of power plants with an F1 score of 0.71 and 0.8 respectively. The model was applied to analyse the city of Chemnitz, and the results were compared to official German registries for renewable power plants. For each of the 6 power plant types, the model was able to detect more instances than were known in the official registry, highlighting the usefulness of automatic detection systems for renewable power plant registration.

U-Net

Another example of FCNs is the work by Zech and Ranalli [71] who trained a U-Net on a manually labelled dataset of German aerial images. They noticed a variety of similarly looking objects that were probable to hinder performance of the networks, including ST systems and greenhouse-like rooms connected to a house. The authors experimented with different variations of ResNet as a backbone and found ResNet-50 to slightly outperform the other variants. Despite the similar looking objects in many training and test images, the model was still able to reach an IoU score of 69% on the test set.

Parhar et al. [72] worked on HyperionSolarNet and showed that it is possible to train FCNs with relatively little training data. They manually annotated 836 images for training and validation of a U-Net, which was used on all positively classified images outputted by a trained EfficientNet-B7 classification network. The classification network was trained by finetuning a pretrained network, and the segmentation network was trained from scratch. The segmentation network achieves an IoU score of 0.82 on the test set, suggesting that a dedicated segmentation network for PV segmentation can be trained with relatively little training data.

Kausika et al. [5] applied a variation of U-Net to Dutch aerial images to segment PV systems. They applied TernausNet, a network built by replacing the encoding

part of the U-Net with a VGG16 encoder, to True Ortho (TO) images of the Netherlands. The images are created by adjusting for the angle at which aerial images were taken, such that the image appears to be taken from straight down at every point, which is not normally the case with aerial images. This aligns images perfectly with coordinates, but the downside of this approach is that the method is never able to determine the correct pixel for every point on the map, leaving blank spots. See Figure 3.14 for an example. Despite this, the algorithm was still able to achieve a precision and recall of 94% and 91% respectively. This is partly due to the post-processing step, which removed false positives such as shadows or greenhouses by cross-referencing the detection locations with topographic datasets.

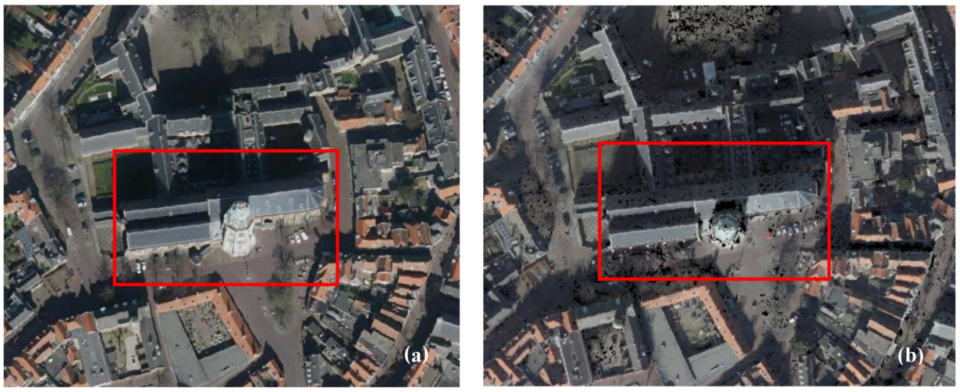

Figure 3.14: An illustration of the True Ortho transformation from the original image (a) to the adjusted image (b). Black spots are points where no pixel colour could be determined. Illustration taken from [5].

Another rather large-scale study was conducted by Kruitwagen et al. [73] who used a U-Net in combination with a Recurrent Neural Network (RNN) to create a global inventory of large-scale PV plants. The U-Net was trained to maximize recall as an initial stage of the pipeline, false positives were then removed by the trained RNN. The model was trained primarily using data from OSM, but the authors made some manual annotations as well. They then used the pipeline to process satellite images on areas with a human population covering the entire globe and located 68,661 facilities, including their estimated power generation capacity. It was noted that most PV facilities are located on croplands, aridlands, and grasslands, while only a small portion of PV systems were detected in built-up areas, most likely due to the low spatial resolutions that were used which do not allow the model to detect small scale PV systems well that are often found in urban areas.

Zhuang et al. [74] used an ensemble learning method where a community of U-Nets is trained at once, and periodically exchanges information to improve per-

formance of the other networks. The U-Nets are trained in a set of epochs, in some of which the best U-Net is selected and used to transfer weights to all other U-Nets if the performance improves after this transfer. While this is computationally much more expensive compared to training a single network, it helps to prevent U-Nets reaching a local maximum performance during training. To train the networks, they used images with 30 cm spatial resolution and achieved an IoU of nearly 75%, which is not as high as many other methods have shown when training on data with a higher resolution. This seems to suggest that while ensemble methods can help to find the best parameters in a specific setup, it is not as important as training on the best possible dataset.

3.4.2 Weakly supervised PV segmentation

The DeepSolar project by Yu et al. [37] took a weakly supervised approach to segmenting the PV systems detected by the classification algorithm. Instead of training a segmentation model on labelled data, a segmentation branch was added after one of the initial layers of the classification model. This branch contained only two convolutional layers, which are again trained on image-level classification of PV systems. However, only one layer is trained at a time, and all other weights in the network remain fixed during training of the segmentation branch. This greedy approach allows the network to focus on low level features without creating a lot of noise. An illustration of this training approach can be found in Figure 3.15. During segmentation, the CAM of the final convolution layer of the segmentation branch is inspected for a given input image, and used to determine which pixels are most likely to constitute a PV system, thresholded by a constant probability value. IoU scores of the segmentation approach are not presented, but the authors report an area-based mean relative error of 3.0% and 2.1% for residential and non-residential areas respectively. That is, the area of the segmented PV systems is on average 3.0% and 2.1% off from the actual area of the PV systems.

This approach to segmentation is used in all DeepSolar follow-up projects up until the 3D-PV-Locator project by Mayer et al. [41]. In this work, they instead replace the weakly supervised segmentation branch with a dedicated segmentation network relying on the DeepLab-v3 architecture with ResNet-101 as a backbone. They showed that with the use of transfer learning the segmentation branch can be finetuned for the task of PV segmentation with only about 4,000 labelled images. Performance does not suffer much from this small finetuning dataset, as the

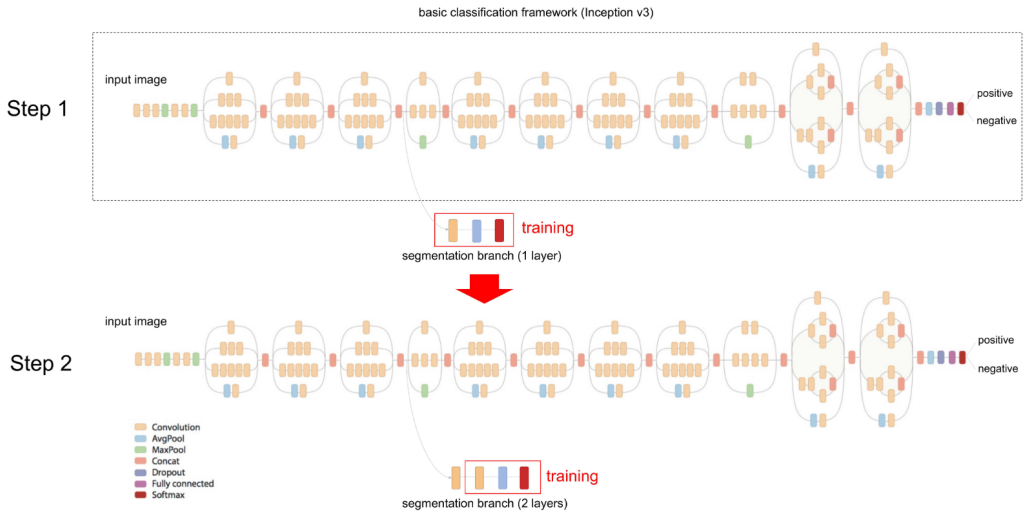

Figure 3.15: An illustration taken from [37] depicting the semi-supervised training strategy for the segmentation branch. A branch is created after one of the initial layers of the trained classification network, which is retrained greedily on a classification task by keeping all weights except for the target layer fixed. After training the two layers of the segmentation branch, the classification head of the new branch is discarded, and the CAM of the last convolutional layer is used to create segmentation masks.

segmentation branch segmented PV systems with a mean IoU of 74.1%

One of the most recent methods for PV segmentation was published by Yang et al. [75], utilising a combination of weakly supervised semantic segmentation methods and semi supervised segmentation methods. They utilise EigenCAMs extracted from a trained classification network to filter SAM [7] generated segmentations. However, as the filtered masks are not always informative, masks with a low total surface area are discarded, and their samples are treated as unlabelled. This set of labelled and unlabelled samples are then used to train UniMatch. With this architecture they are able to achieve an IoU of 73.5%, approaching the fully supervised model IoU performance of 84.1%.

3.4.3 Transfer learning and other approaches

Transfer learning performance for CNNs as PV segmentation networks was investigated by Wang et al. [76] by comparing performance of networks trained on two cities in the Duke California dataset [29]. They found that networks trained using test data from the same city as it was trained on often exhibit optimistic performance. In a more realistic scenario where the model is tested on data from a different city, the performance drops significantly [76]. As noted by other studies in this

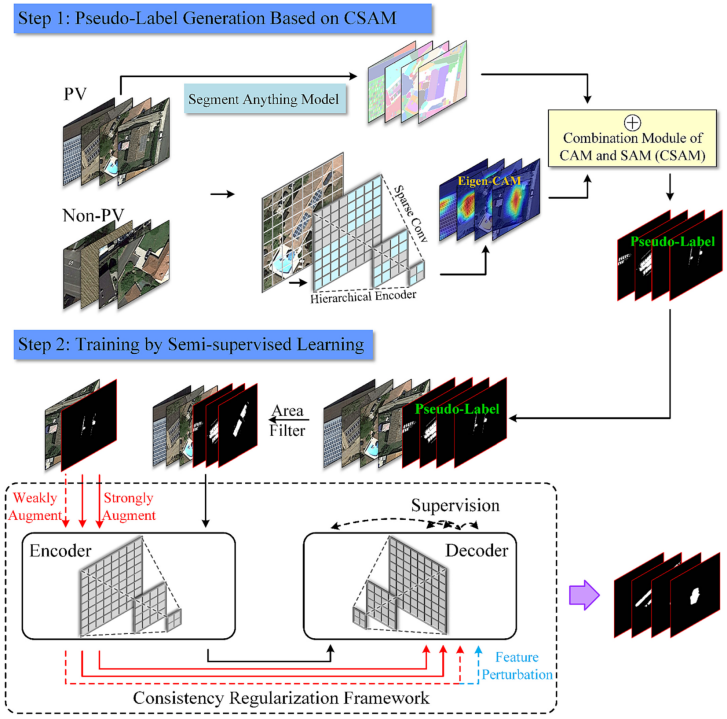

Figure 3.16: Weakly supervised segmentation approach overview from [75].

review, this problem can be alleviated greatly by finetuning the model on the target city using much fewer data than would be required for full training, although it still requires notable manual labelling work.

Segmentation methods have also been developed to segment large scale centralized PV plants, one example of such methods is the work by Hou et al. [77]. They used ResNet-101 as a backbone to extract features and used an EMAU module for self-attention. Training was done on only 819 manually labelled images, although the resolution of these images ranged from 512×512 to 10000×10000, meaning it is probable these images were cut into multiple smaller images. Data augmentations in the form of cropping, rotation, scaling, and flipping were also utilised. Interestingly, they show that their method not only performs well on Chinese imagery, but also on the DeepSolar dataset, and the networks embedded with EMAU modules show improved performance compared to a standard ResNet and U-Net combination. The high performance on the DeepSolar dataset is surprising considering the fact that the authors do not mention training on that dataset, and other studies seem to suggest that direct application of a model trained on one domain to images from another domain usually does not yield performance over approximately 50% mean IoU [34], [76].

Besides satellite and aerial top-down images, there also exist datasets containing videos taken by unmanned aerial vehicles of large solar plants. These videos are collected with infrared or thermal cameras, which can also be used to segment PV systems. Greco et al. [78] applied YOLOv2 and YOLOv3 to these thermal videos to segment PV systems, with the aim of later using these segmentations for anomaly detection. They showed that while performance with training performed only on the training set was already enough to achieve high performance, finetuning on a small part of a testing video could improve performance significantly. Since the system is built upon YOLOv3, an architecture known for its speed, the algorithm is able to produce detections in real time. This would enable the recording aircraft to report in real-time where anomalies lie, assuming these anomalies are also detected in real time.

As an alternative to most deep learning methods discussed above, Karoui et al. [79] proposed a method based solely on non-negative matrix factorization. They used known solar panel surface spectra as a ground truth and used hyperspectral satellite imagery to separate these spectra from. To achieve this, they considered each pixel to be a linear mixture of reflectance spectra. Then, by developing a method to unmix these spectra one can decide which pixel spectrum mixtures are most likely to originate from solar panels, and therefore do pixel-based segmentation on these pixels.

3.5 Datasets

Training deep learning networks requires large volumes of data to train on, and both the quantity and the quality of this data are essential to building a well performing model. Across the globe, efforts have been made to create publicly available datasets tailored to training PV classification and segmentation algorithms, some of which will be discussed in this section.

Perhaps the most often-utilised dataset in early PV classification approaches was created in 2016 by Bradbury et al. [29] at Duke university in California. Four cities in the state of California were selected, from which 601 images of 5000×5000 pixels representing an area of 2.25 square kilometres each were gathered. Every image was annotated manually by two annotators who drew polygons around each PV array. After comparing the results of the two annotators, it was found that only 70% of PV systems were found by both annotators, which shows that even for

human annotators it can be challenging to accurately find all PV systems. This dataset was used by studies such as those by Zhuang et al. [74], Golovko et al. [36], Wang et al. [76], Camilo et al. [65], and Malof et al. [28], [30], [34], [66].

Stowell et al. [80] created a similar dataset based on the United Kingdom. However, instead of doing the annotations themselves, they made liberal use of crowdsourcing. Members of the OpenStreetMap (OSM) community volunteered to locate and label PV installations, which led to a dataset containing 264,641 installations. A big advantage of crowdsourcing is the large volume of data that can be gathered, but a big disadvantage is the difficulty of validating the data. To combat this, the authors created a pipeline for processing the data, which included automated correction of spelling errors in manually written labels, data merging and deduplication, as well as some manual validation.

Often models are trained on datasets of a single spatial resolution, but handling a variety of spatial resolutions can also be beneficial in some scenarios. For example, when doing historical analysis, there might only be data available of lower spatial resolution than that of imagery of the present day. Jiang et al. [81] therefore collected imagery from the Chinese province of Jiangsu at three spatial resolutions: 0.8 m, 0.3 m, and 0.1 m. Images were annotated with polygons by two individual annotators and verified and combined by a third annotator. They used this dataset to compare performance of U-Net, RefineNet and DeepLabV3 on all datasets. Results showed that overall DeepLabV3 had the best performance, and that all networks were able to achieve accuracies of over 95% on every dataset. Accuracy slightly improved as spatial resolution increased, but not by much. Pixel level precision was measured by computing IoU scores and interestingly showed that the highest IoU was achieved on the dataset with spatial resolution of 0.3 m. Experiments with cross learning were also carried out, which showed that while direct application of a network trained on a different spatial resolution than the target dataset resulted in extremely poor performance, finetuning on a small set of data (20% of the training set) with the target resolution resulted in performance that was close to, if not better, than performance resulting from direct training on the target resolution.

3.6 Gaps in research

As illustrated in this review, PV detection based on satellite or aerial imagery has been addressed abundantly. However, with the exception of a handful of studies such as [79], multi-modal data extending the standard RGB channels of an image have not been explored. This gap in research is addressed with the first research question: *"To what extent can building location data be utilised in addition to RGB channels to improve the performance of a PV and ST detection model?"*.

Self-supervised training methods have been shown to be effective in many domains, but have known little attention in solar panel detection research. With the large volume of aerial images openly available, it seems sensible to explore the possible benefits of self-supervised learning on this data. Therefore, the second research question *"What is the effect of self-supervised pretraining on a large domain-specific dataset on the performance of a PV and ST detection model?"* targets this gap in research.

Additionally, while much work has been done on segmenting PV systems, only a small subset of articles (such as the DetEEktor project [70]) explore the detection and segmentation of ST systems. This might be due to the fact that PV systems are more popular than ST systems, because they might be hard to differentiate from PV systems, or because of a lack of data regarding the locations of these systems. This is the gap in research aimed to be filled by answering the third research question: *"To what extent can photovoltaic and solar thermal systems be distinguished by a machine learning model?"*.

Finally, apart from the DeepSolar projects [37], [39] and the recent work by Yang et al. [75], all studies on solar panel segmentation have focussed on fully supervised training. To decrease the need for manual labelling and explore the most recent advancements in weakly supervised and semi-supervised learning, the fourth research question *"What is the performance impact of choosing a semi-supervised or weakly-supervised segmentation approach over a fully-supervised approach for PV and ST segmentation?"* aims to investigate the potential of further work on weakly- and semi-supervised learning in PV and ST segmentation.

4. Data

An annotated dataset for PV and ST detection on Dutch aerial imagery was not publicly available at the time of this research. Therefore, a new dataset was manually annotated. This section will cover the source of the created dataset, the sampling strategy used, how building information was incorporated, the labelling process, and conclude with a comparison with datasets used in similar research.

4.1 PDOK Aerial Imagery

The Dutch government provides a large variety of geographical datasets which can be used freely. Among these are aerial images by PDOK (Publieke Dienstverlening Op de Kaart). PDOK is a Dutch public platform that can be used to access datasets provided by the Dutch government, specifically in the field of geographical data [82]. PDOK has published nation-covering annual aerial imagery since 2016 at a spatial resolution of 0.25 m [83], see Figure 4.1 for a visualisation of the imagery. From 2021 onwards the aerial images are also taken at a spatial resolution of 0.08 m. In this research, the 2023 high resolution version of the aerial images was used. Images were sampled at a resolution of 224x224 pixels, spanning an area of 16x16 meters per sample, therefore the sampled spatial resolution of the images is approximately 0.0714 m.

Figure 4.1: PDOK aerial images

If the dataset were to be randomly sampled, it can be expected to be extremely imbalanced, as only a small portion of the Dutch surface is covered by PV or ST panels. To mitigate this, a dataset containing locations of PV and ST panels was

used in a sampling strategy aimed to create a more balanced distribution of positive and negative samples. The dataset contained addresses throughout the Netherlands with an indication of either PV or ST presence, or both. Unfortunately, at some time between utilising the dataset for this study and the time of writing for this thesis the dataset has been made private. However, while the dataset was used to find locations with PV and ST panels and provide initial labels, all images were manually reviewed to ensure the correctness of the labels.

4.1.1 Sampling strategy

To sample images containing PV and ST panels, the dataset was used as follows: A random subset of all addresses containing either PV or ST panels was selected. For each address, 4 tiles were downloaded, each with the centre of the address set to a different corner of the tile. To make sure the dataset also contained enough negative samples, two random locations between a range of 150 and 650 meters from the address were selected, and a tile with this location as the upper left corner was also exported. For the tiles selected from the known addresses, the corresponding PV or ST presence was marked as an initial label for the image. The tiles selected from a random unmarked location remained unlabelled. Thus, for each address, 6 tiles were exported, 4 with a label, and 2 without. After these initial labels were set, all instances were reviewed manually and adjusted accordingly. The manual review of the labels caused more samples initially labelled as positive to be corrected to negative than the other way around, since panels often do not cover the entire roof of an adress, resulting in a balanced distribution of labels.

4.2 BAG

PV and ST panels are often installed on rooftops. To potentially allow the model to learn this pattern, publicly available building information was used in the form of BAG (Basisregistraties Adressen en Gebouwen) polygons. BAG is a registration provided by the Dutch government containing the addresses and shapes of buildings in the Netherlands [84]. In this study, only the top-down building shapes were used, other information such as addresses was discarded.

For every sample in the image dataset, the corresponding BAG polygons were retrieved as binary masks with a resolution and location corresponding to the images. Since polygons in this set were actually made available as greyscale images,

with addresses in the same building usually separated with a thin black line, a preprocessing step was applied to compensate. BAG polygon masks were first thresholded at 128 to create binary masks and remove the anti-aliasing effect from the WMS server. A 3 by 3 convolution of ones was then applied to remove lines separating addresses housed under the same building, after which another threshold of 75 was applied. Figure 4.2 shows the results of preprocessing.

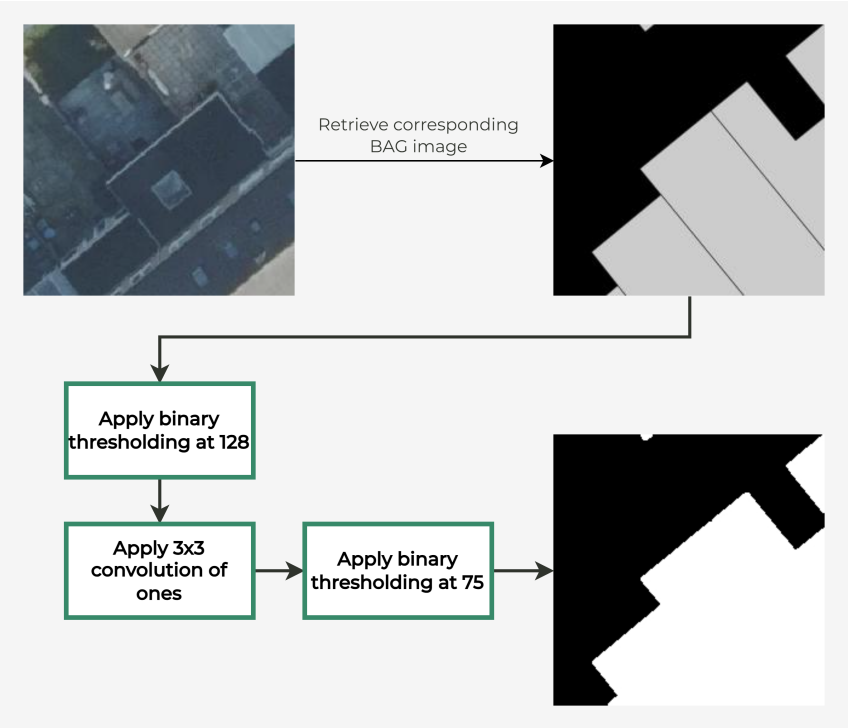

Figure 4.2: BAG processing steps

4.3 BAG Refinement

Since the aerial images in the dataset are not truly orthogonal to the surface, the images might not properly align with the BAG polygons. BAG polygons assume true orthogonal projection on the surface, but the aerial imagery is often taken at a slight angle as flying precisely above every part of the landscape would be infeasible. Parts of the rooftop on the aerial images then fall outside the BAG polygon as a result, and at other locations BAG polygons contain a part of the image that does not depict a roof. Figure 4.3 shows an example of such misalignment.

To mitigate this issue, BAG polygons were refined using CascadePSP [63], utilising the segmentation-refinement Python package [85]. The BAG binary masks were used as low resolution masks in CascadePSP with the corresponding aerial

Figure 4.3: Misalignment between BAG polygons and aerial image. BAG polygon in grey.

image as input, to generate new refined masks fitted to the buildings on the image. This was done for all 50,000 images in the labelled dataset. This set of refined masks will from here on be referred to as refined BAG. Figure 4.4 highlights a comparison between the original fitting of BAG masks and the fitting after refinement.

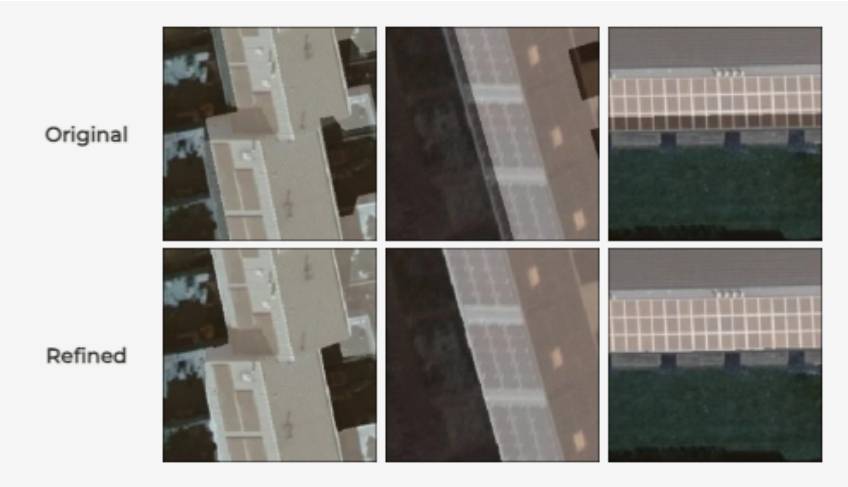

Figure 4.4: Examples of BAG refinement results with CascadePSP. Mask highlighted in white.

4.4 Image level labels

50,000 images were imported from the PDOK dataset using the described sampling method. As the initial labels might contain errors, all images were manually reviewed to correctly label whether each image contains PV or ST panels. Manual correction was done utilising Computer Vision Annotation Tool (CVAT) [86]. The

resulting dataset was randomly split into a training set of 45,000 images and a test set of 5,000 images. The resulting training set contained 16,466 images with PV panels, 1,692 images with ST panels, and 27,402 images without any panels. The test set contained 1,830 images with PV panels, 173 images with ST panels, and 3,104 images without any panels. Both sets have a slight imbalance towards negative samples, although the distribution is much less skewed than can be expected from a randomly selected area of Dutch areal imagery. It is therefore expected that the model will be able to properly learn from this dataset. The distribution of PV and ST panels in the positive samples is however rather skewed, due to the fact that PV panels are much more common than ST panels in the Netherlands.

4.5 Pixel level labels

For the purpose of testing the segmentation models, positive samples from the test set were also manually annotated at a pixel level. This labelling was also performed in CVAT.

Finally, to train fully-supervised and semi-supervised models, pixel-level annotated training samples are required. Therefore, a subset of 3,000 images was randomly selected from the positive samples in the training set and manually annotated as well. For the semi-supervised model, the remaining positive samples from the training set were used as unlabelled samples.

4.6 Unlabelled samples

For the masked auto-encoder pretraining stage of the classification network, a large set of 500,000 new images were also sampled from the PDOK dataset. The images were sampled in an identical manner as was done for the labelled dataset, although the initial labels were discarded. Corresponding BAG polygons were also retrieved for these images, and preprocessing of the polygons was again performed.

4.7 Dataset overview and comparison

An overview of this dataset in comparison to datasets used in similar research can be found in Table 4.1

While studies have shown that good classification or segmentation results can be achieved with datasets much smaller than the proposed dataset, there have also

Table 4.1: Statistics of datasets used in similar research

Study	Samples	Image size	Spatial resolution
This Study	**50,000**	**224x224**	**0.07 m**
Bradbury et al. [29]	601	5000x5000	0.45 m
Jiang et al. [81]	763 (0.8 m); 2,335 (0.3 m); 645 (0.1 m)	1024x1024 (0.8 m & 0.3 m) 256x256 (0.1 m)	0.8 m; 0.3 m; 0.1 m
Yu et al. [37]	472,953	320x320	0.3 m
Mayer et al. [40]	38,304 (Google); 70,673 (OpenNRW)	320x320	0.05 m (Google); 0.1 m (OpenNRW)
Lindahl et al. [42]	57,839	299x299	0.0615 m
Zech et al. [71]	1,325	630x640	0.2 m
Parhar et al. [72]	2,455	416x416, 600x600	*not given*
Zhuang et al. [74]	1,414	256x256	0.3 m
Yang et al. [75]	28,484	256x256	0.1-0.2 m

been studies that show the importance of larger volumes of data. Spatial resolution of this dataset is among the highest in recent literature.

The dataset of labelled images, including BAG and refined BAG masks, is made publicly available via the Utrecht University Research Data Management system Yoda, and can be found at `https://public.yoda.uu.nl/science/UU01/NRFYSC.html` (or: `https://doi.org/10.24416/UU01-NRFYSC`). Labels are provided both at image level, and at pixel level for the applicable samples.

5. Background

In this chapter, important techniques and architectures that are utilised in the proposed method of this thesis will be discussed in more detail. The aim is to provide the reader with an understanding of the inner workings of these techniques, such that the focus of the methodology chapter can be on the implementation of these techniques in the proposed method. This chapter will discuss the computation of Class Activation Maps, the ConvNeXt V2 architecture, the DeepLabV3+ architecture, the CorrMatch algorithm, and conclude with an explanation of the metrics used to evaluate the performance of the proposed method.

5.1 Class Activation Maps

Class Activation Maps (CAM) are a method of computing the importance of each pixel in an input image of a CNN to the final classification decision. The method relies on combining the output of the final feature map with the class importances learned by the final fully connected layer on the last global average pooling layer.

A global average pooling layer is a layer that outputs the average of every feature map in its input. The layer is used to reduce the spatial dimensions of the feature maps to a single value which is then often fed into the final fully connected layer. The fully connected layer then learns a weight for each feature map, determining the importance of each feature for the classification of the specific class. The weight of each feature can then be fed back and multiplied with the corresponding feature map before global average pooling to obtain the localized importance for a class on each feature map. The weighted feature maps are then stacked into a single image and upsampled to the original image size to obtain the CAM. Figure 5.1 illustrates the process.

5.2 ConvNeXt V2

In this thesis, for both classification and segmentation, ConvNeXt V2 is used as the backbone. ConvNeXt V2 is the successor to the original ConvNeXt architecture proposed by Liu et al. [23]. The aim of this research was to build a modern CNN

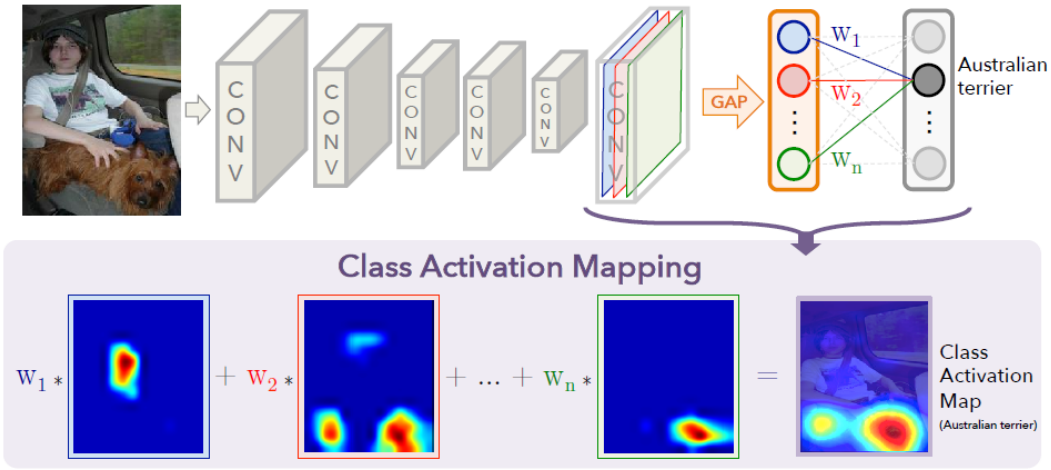

Figure 5.1: Visualisation of Class Activation Map computation. The learned weights w_1-w_n are multiplied with the corresponding feature maps and stacked to obtain the CAM for a given class.

architecture based on principles learned in the development of vision transformers.

The first of these principles was adjusting the stage computation ratio used in the network. Often CNN architectures are build by repeating a specially designed "block" of layers a certain amount of times. In between a set of blocks, downsampling layers are then introduced to reduce the spatial dimensions of the network gradually. The series of blocks between downsampling layers are often referred to as stages, the amount of stages and the amount of blocks per stage are then referred to as the stage computation ratio. Where the original ResNet architecture had the following ratio of computations per stage: 3:4:6:3, the authors adopted the 1:1:3:1 ratio from Swin Transformers and fitted the network with a block distribution per stage of 3:3:9:3 for the smaller variants of the network, and a 3:3:27:3 ratio for the larger variants.

Next, the network stem was adjusted. Traditionally, a CNN stem downsamples the input image at the root of the network by applying an aggressive overlapping convolution. In ViTs however, the input is split into non-overlapping patches which are processed separately. This idea is implemented by utilising a 4x4 convolution with a stride of 4 in the first layer of the network. Meaning the convolutions on the input are non-overlapping.

Another notable design choice transferred from transformers is to use an inverted bottleneck at the start of each block, meaning that the dimensions of the hidden layers in the block are wider than the input and output of the block.

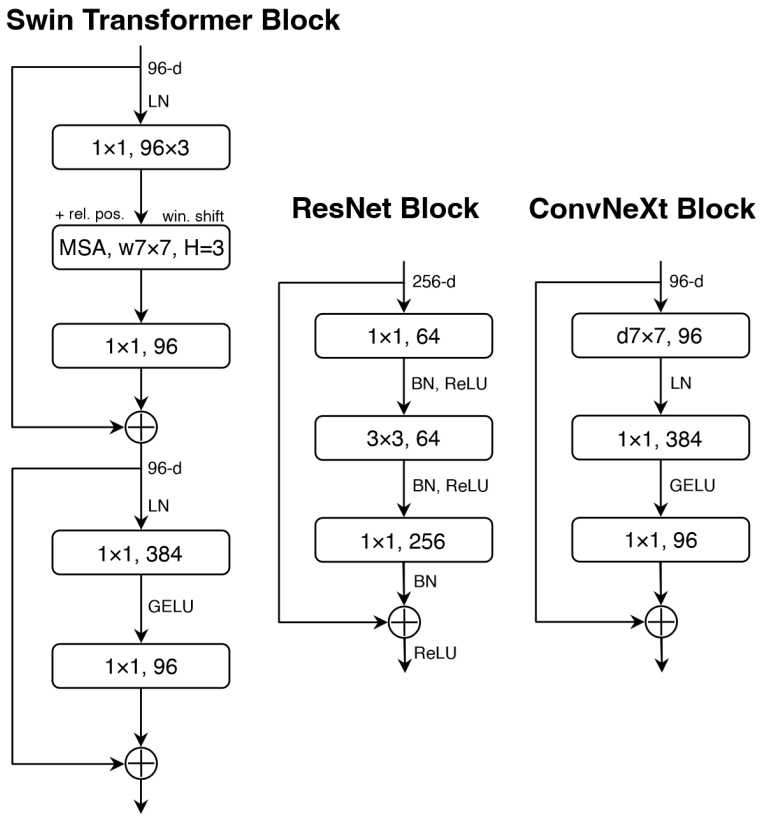

Figure 5.2: Comparison of block design between Swin Transformers, ResNet, and ConvNeXt [23]. Here, BN is Batch Normalization, LN is Layer Normalization, GELU is Gaussian Error Linear Unit activation, and MSA is Multi-head Self Attention.

The final architecture design adjustment was the increase in kernel size. To mimic the global attention mechanism pivotal in transformers, the authors increased the kernel size of depth wise convolutions in each block from 3x3 to 7x7.
A few more adjustments were made at a micro level. These include replacing ReLU activation with Gaussian Error Linear Unit (GELU) activation, reducing the number of activation functions in a block, reducing the number of normalization layers, adding separate downsampling layers, and finally substituting Batch Normalization with Layer Normalization. Layer normalization computes a normalization factor based on all the inputs of a hidden layer, instead of based on all the input in a batch. A comparison between Swin Transformer, ResNet, and ConvNeXt blocks is illustrated in Figure 5.2, and the full architecture of ConvNeXt is illustrated in Figure 5.3.

ConvNeXt V2 [25] improved upon this architecture even further by introducing Masked Auto-Encoder pretraining. An adjustment to the convolutional nature of the networks was needed however, as regular convolutions are not able to deal with

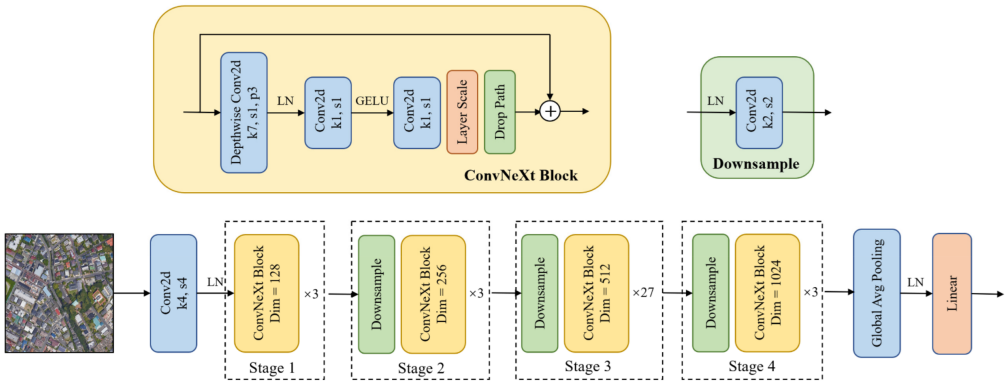

Figure 5.3: ConvNeXt architecture, Figure taken from [87].

masked regions. To solve this issue, submanifold sparse convolutional layers were utilised instead of the regular convolutional layers during pretraining. Submanifold sparse convolutions work similarly to regular convolutions, but only return an output for positions where the centre of the kernel covers an unmasked pixel, all other points in a feature map are simply forwarded as masked values for the next sparse convolutional layer. Thus, when computing the dot product of the kernel and the sparse patch, only unmasked positions are included in the computation.

However, it was found that even after these adjustments, applying MAE pretraining on the original ConvNeXt architecture produced unsatisfying results. This was because the network suffered too much from feature collapse, meaning a heightened amount of irrelevant features were learned. To mitigate this issue, the authors introduced a novel Global Response Normalization (GRN) layer. This layer takes a feature map, computes an aggregate vector of the feature map, then calculates a normalization factor based on this aggregate vector, and finally calibrates the input feature map based on this normalization factor. This normalization is incorporated into a learnable linear function with a residual connection. The final learnable function is added to the ConvNeXt block after the GELU activation. Finally, LayerScale was considered, which was introduced to the network in the original ConvNeXt architecture to perform normalization after the residual blocks which improves training dynamics. The layer was however found to be redundant after the addition of the GRN layer, and therefore removed. A comparison of the ConvNeXt V1 and V2 blocks can be found in Figure 5.4.

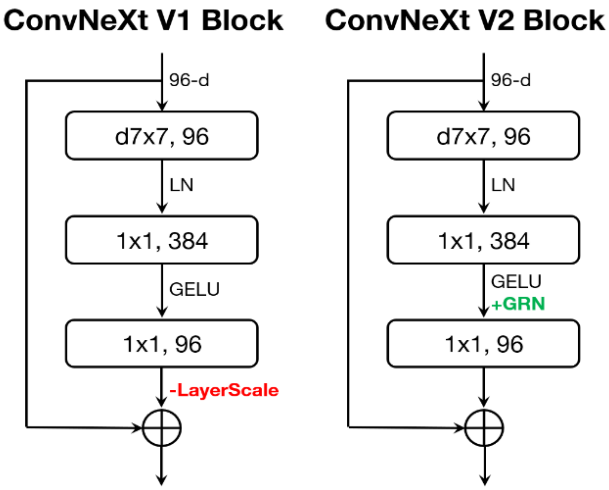

Figure 5.4: ConvNeXt V1 and V2 block comparison [25].

5.3 DeepLabV3+

The first iteration of DeepLab [51] introduced the use of atrous convolutions in semantic segmentation with the aim of capturing a large field of view when convolving without increasing computational costs. For a one-dimensional convolution, the formula to compute the output of a position i is then given by:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k] \tag{5.1}$$

Here, x is the input signal, K is the filter length, w is the kernel, and r is the stride with which the input signal is sampled. A larger r means larger gaps in the convolution. DeepLabV2 then introduces atrous spatial pyramid pooling (ASPP), meaning that for classification of a pixel, multiple atrous convolutions with different rates are applied. In DeepLabV3 [52], the ASPP layer is further investigated, and extended by also incorporating image features in the ASPP layer. This is achieved by applying global average pooling to the last image feature map and feeding the result into a 1x1 convolution which is then upsampled to the proper spatial dimension to match the other convolutions.

As is illustrated in Figure 5.5, the DeepLabV3 architecture mostly relied on the use of atrous spatial pyramid pooling near the end of the convolving part to capture features with different context frame sizes. These features were then used in

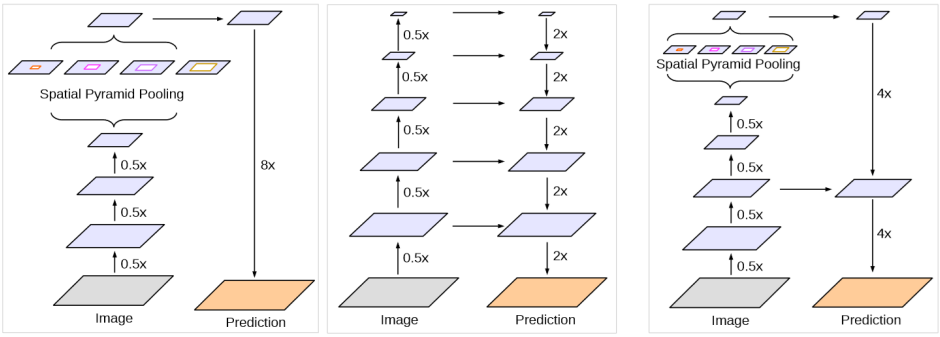

Figure 5.5: DeepLabV3 architecture with spatial pyramid pooling (left), Encoder-Decoder architecture (middle), and DeepLabV3+ architecture (right). [53].

a single upsampling operation to make a final prediction. However, many studies have shown having an asymmetrical encoder-decoder architecture with intermediate connections from the encoder can boost performance for semantic segmentation. In DeepLabV3Plus [53], a simple but effective decoder module is therefore presented to work in conjunction with the encoder. The full pipeline including the proposed decoder can be found in Figure 5.6. The decoder module works by Applying a 1x1 convolution on low level features to reduce dimensions, and combining the results with an upsampled result from the final ASPP layer. The combined features are then once more passed through a 3x3 convolution and upsampled to obtain the final result.

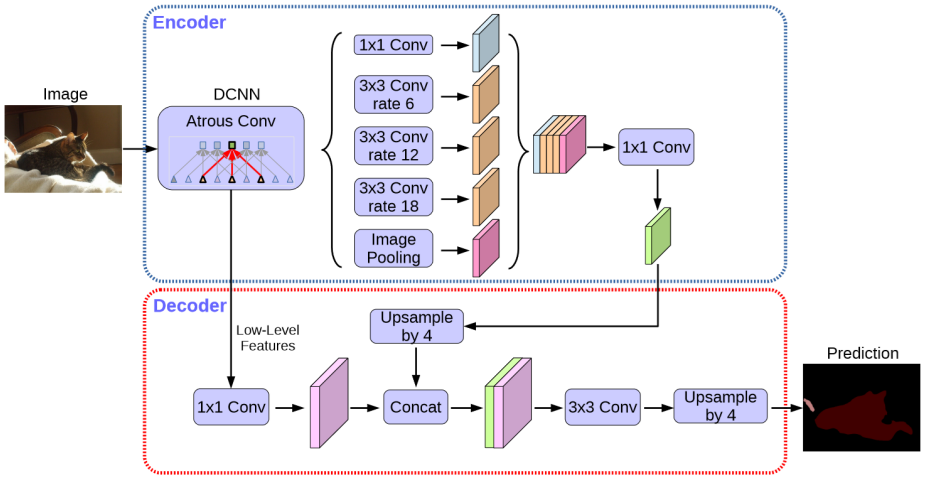

Figure 5.6: The full DeepLabV3+ architecture. [53].

5.4 CorrMatch

CorrMatch [62] is an approach for semi-supervised semantic segmentation which builds on FixMatch [60], but utilises correlation maps to compute an additional correlation loss, as well as refine pseudo labels. The correlation map C is computed between two learnable linear layers after the encoder of the network by performing a matrix multiplication between the features extracted by these layers:

$$C = Softmax(w_1^\top \cdot w_2)/\sqrt{D} \tag{5.2}$$

Where w_1 and w_2 are the feature maps extracted by the linear layers, and D is channel dimension. The correlation map is then applied to the model logits outputs on the unperturbed unlabelled image to produce a new representation of the prediction:

$$z_i^u = f_1(\hat{F}(x_i^u)) \cdot C \tag{5.3}$$

f_1 here is bilinear interpolation for shape matching, and $\hat{F}$ is the model output logits. To obtain the correlation loss L_u^c, this representation is compared to the high confidence pseudo labels computed on the weakly perturbed images:

$$L_u^c = \frac{1}{|N|}\sum_{i=1}^{N}(l_c(z_i^u, F(x_i^w))) \odot M_i \tag{5.4}$$

here, l_c is the cross entropy loss, and M_i is a binary map of high confidence pixels.

Besides building a correlation loss, the correlation maps can also be utilised to enhance the pseudo labels. To achieve this, a sampled row c from the correlation map is thresholded and turned into a binary map $\hat{c}$. With this binary map, the overlap of a class l with this map and the high confidence binary map is computed

with function G:

$$G(l) = \sum_{HW} \mathbb{1}[(F(x_i^w) \odot M_i \odot \hat{c}) = l] \tag{5.5}$$

The class for which G is maximised is set as k^*, if this k^* overlaps sufficiently with a high confidence shape, the class is expanded into the high confidence region. This is repeated for a number of samples, and only if the correlation row overlaps more than a given threshold with the high confidence region map. The full pipeline for unlabelled images is visualised in Figure 5.7.

Using this refinement method to refine pseudo labels computed from the weakly perturbed images, two more losses are introduced. A hard supervision:

$$L_u^h = \frac{1}{N} \sum_i^N l_c(F(x_i^s), F(x_i^w)) \odot M_i \tag{5.6}$$

where l_c is again cross entropy loss (note also that $F(x_i^w)$ here should be refined with correlation as described above), and a soft supervision:

$$L_u^s = \frac{1}{N} \sum_i^N KL(\hat{F}(x_i^s), \hat{F}(x_i^w)) \odot M_i \tag{5.7}$$

Where KL is the Kullback-Leibler divergence which operates on the model logits output.

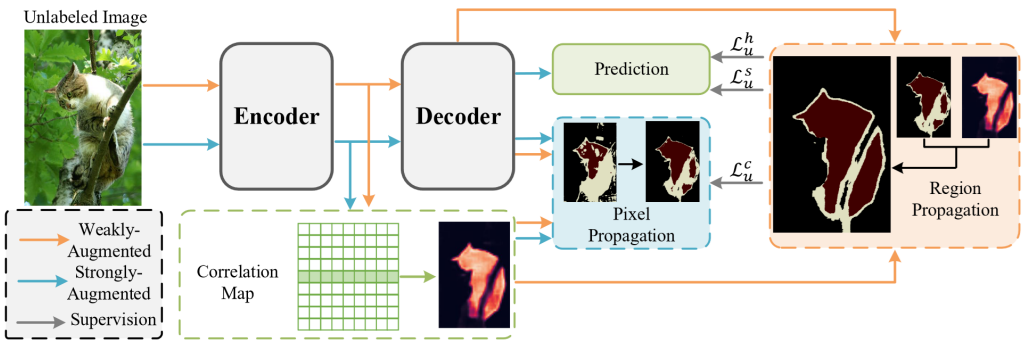

Figure 5.7: CorrMatch architecture [62].

5.5 Metrics

Evaluation of classification and segmentation performance is done utilising a variety of metrics. For classification, precision, recall and F1 score are used. These metrics are computed utilising the following equations[1]:

$$Precision = \frac{TP}{TP + FP} \tag{5.8}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.9}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.10}$$

Precision is a measure of the degree to which samples classified as positive truly belong to the positive class. Recall instead measures the degree to which samples from the positive class are classified as such. The F1 score is the harmonic mean of precision and recall, and is a measure of the balance between the two.

Intersection over Union (IoU) is defined as the pixel count in the intersection of ground truth and predicted mask, divided by the pixel count of the union of ground truth and predicted mask. The metrics is used as the main metric to evaluate segmentation performance. Figure 5.8 illustrates the IoU metric.

IoU could be seen as a combination of precision and recall. If precision is low, the positively classified pixels will not intersect with the ground truth, thus the union increases and the IoU decreases. If the recall is low, the intersection decreases, and the IoU decreases as well.

[1]TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives

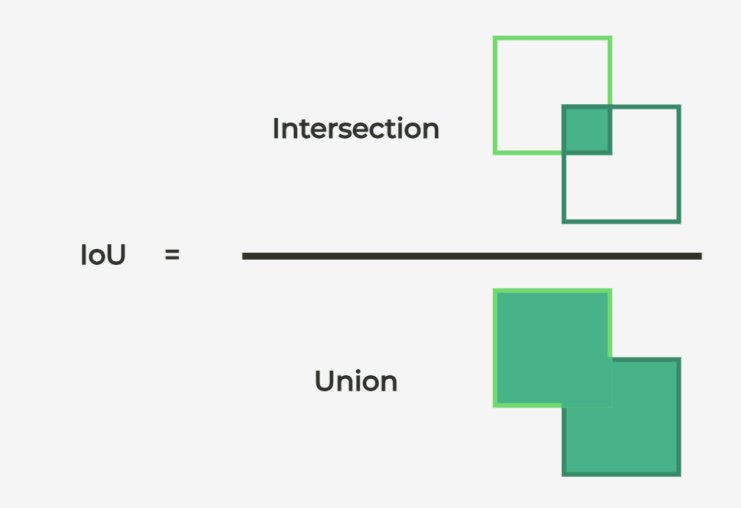

Figure 5.8: Visualisation of Intersection over Union metric. Here, one green square can be seen as the ground truth, and the other as the predicted mask.

6. Methodology

In this chapter, the methodology applied in this thesis will be described. The proposed model architecture and networks that were used will be detailed. Then, the pretraining strategy used to find good baseline encoder weights will be explained. Next, the classification task and training setting are described. Finally, the fully, semi-, and weakly-supervised segmentation approaches are elaborated on, including a detailed description of the pseudo-label generation and selection strategy.

6.1 Model architecture

To analyse the capacity for renewable energy generated by solar panels in a municipality from aerial imagery, large volumes of images need to be processed. However, many of the images to be processed will not contain solar panels. As segmentation networks are generally more computationally expensive to apply to an image, it appears wasteful to apply them to the large amount of images not containing solar panels. Motivated by this observation, a two-step architecture was developed consisting of a classification step that processes all images, and a more computationally expensive segmentation step that only processes images classified to contain solar panels by the classification network. An added benefit of this approach is that the classification network can be used to generate CAM-based pseudo-labels for the segmentation network, and the backbone weights can be reused as well.

ConvNeXt V2 [25] was used as the backbone for both classification and segmentation. Multiple network variants of ConvNeXt V2 exist, varying in dimensions and stage sizes. In this project, the 'base' model was used, with a block distribution of 3:3:27:3, and a dimension distribution of 128:256:512:1024, both relative to stages 1-4. To take full advantage of the ConvNeXt V2 architecture designed around masked auto-encoder pretraining, the network was pretrained on the unlabelled dataset.

6.2 Self-supervised Pretraining

While manually labelling a dataset is a labour-intensive task, a very large sum of unlabelled data can easily be obtained, as the entirety of the Netherlands has been

captured in the aerial imagery. In an attempt to take advantage of this large sum of data, a masked auto-encoder (MAE) pretraining strategy was utilised. As described in Chapter 4, 500,000 unlabelled images were collected in a similar manner to the data that was collected for manual labelling. Pretraining was done on this unlabelled dataset of 500,000 images for 50 epochs with 5 warm-up epochs. A masking ratio of 0.6 was utilised, as it was shown by Woo et al. [25] to provide the best results for ConvNeXt V2. The base learning rate was set to 1.5e-4, and a linear lr scaling rule was used similar to [25]: $lr = base\,lr * batchsize/256$. A batch size of 128 was used, and weight decay was set to 0.05. AdamW was used as the optimizer. A horizontal flip applied with a probability of 0.5 was used as data augmentation.

For classification, experimentation was done with both 3-channel (RGB) and 4-channel (RGB+BAG) models. As the encoder in the 4-channel case should also learn to extract features from the 4th channel, the pretraining stage was performed separately for the 3-channel and 4-channel cases.

Thus, in total 2 encoders were pretrained:

- RGB encoder (3-channel)
- RGB+BAG encoder (4-channel)

The second encoder was built and pretrained by modifying the default ConvNeXt V2 encoder stem to accept 4 channels instead of 3, and by modifying the decoder to predict 4 output channels instead of 3. The BAG data was concatenated to the RGB data for each pretraining image. Thus, the network is tasked not only with reconstructing the missing RGB patches, but also with predicting the BAG polygon shapes in the missing patches.
Only the original BAG polygons were used for pretraining, as refining the large number of masks in the unlabelled dataset with CascadePSP was computationally infeasible. See Figure 6.1 for an overview of the pretraining process for the different variations.

6.3 Classification

After MAE pretraining, multiple variants of the classification network were fine-tuned on the labelled dataset. Of the 50,000 labelled images, 45,000 were used for training, the remaining 5,000 were used for testing.

The classification networks were built by adding a classification head to a pre-

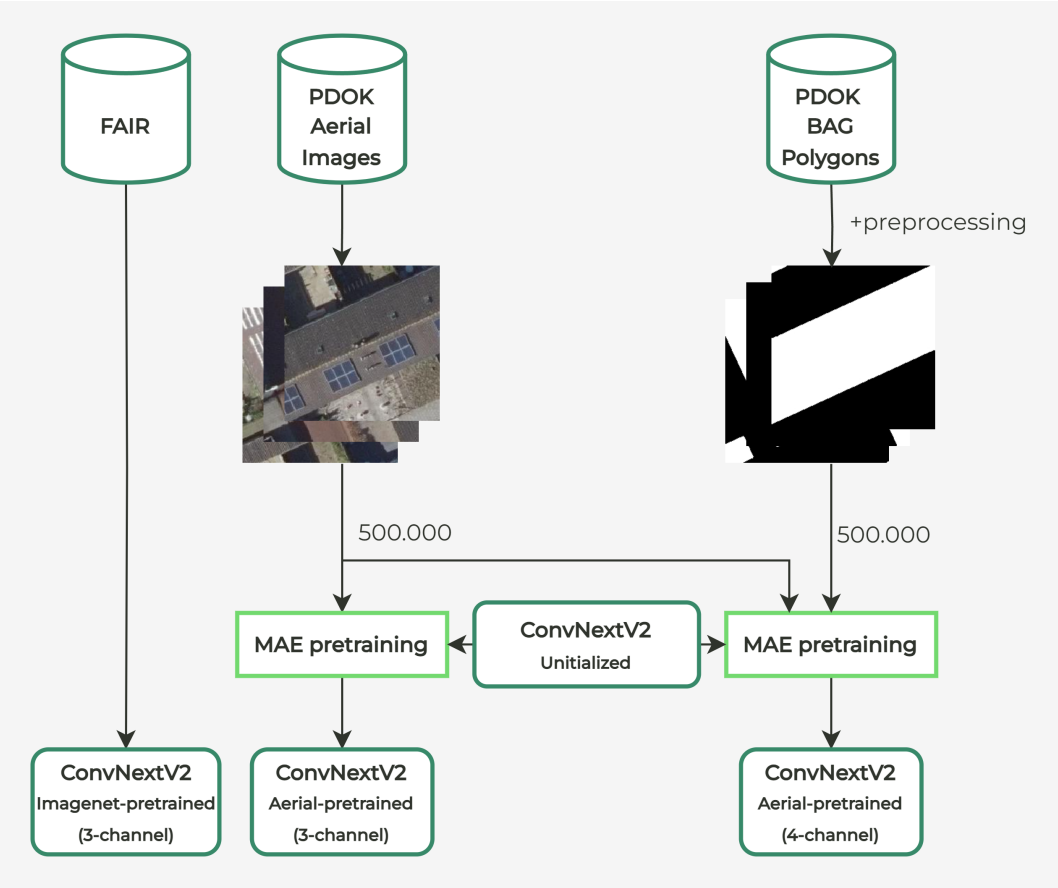

Figure 6.1: An overview of the origin of the 3 sets of pretrained weights used in later stages, and how these weights were acquired.

trained encoder. The classification head consisted of a normalisation layer, and a fully connected layer with 2 output nodes. In the binary task of solar panel detection, the 2 classes indicated the absence and the presence of a solar panel (either PV or ST). In the multi-label task, the first class indicates the presence of PV panels, and the second class indicates the presence of ST panels.

Both 3-channel (RGB) and 4-channel (RGB+BAG) variants of the classification network were finetuned on the labelled dataset. For the 3-channel variant, finetuning was performed based on the weights pretrained using aerial imagery, as well as on the public weights from [25] pretrained on ImageNet. For the 4-channel variant, finetuning was performed based on the weights pretrained using aerial imagery combined with BAG data only. The 4-channel variant, however, was finetuned with both the original BAG-polygons, as well as the BAG-polygons refined with CascadePSP.

Finally, finetuning was also performed for binary classification, where PV and

ST panel labels were combined into a single label. For binary classification, the 3-channel network was again finetuned both using aerial image pretrained weights, and ImageNet pretrained weights. The 4-channel network was also again finetuned from the 4-channel pretrained weights, both on the original and refined BAG-polygons. An overview of all the network variants trained, and the resources used per network can be found in Figure 6.2. The finetuning process resulted in a total of 8 trained classification networks, 4 networks for 3 channel data and 4 networks for 4 channel data. The 8 networks can also be divided into 4 binary classification networks, and 4 multi-label classification networks.

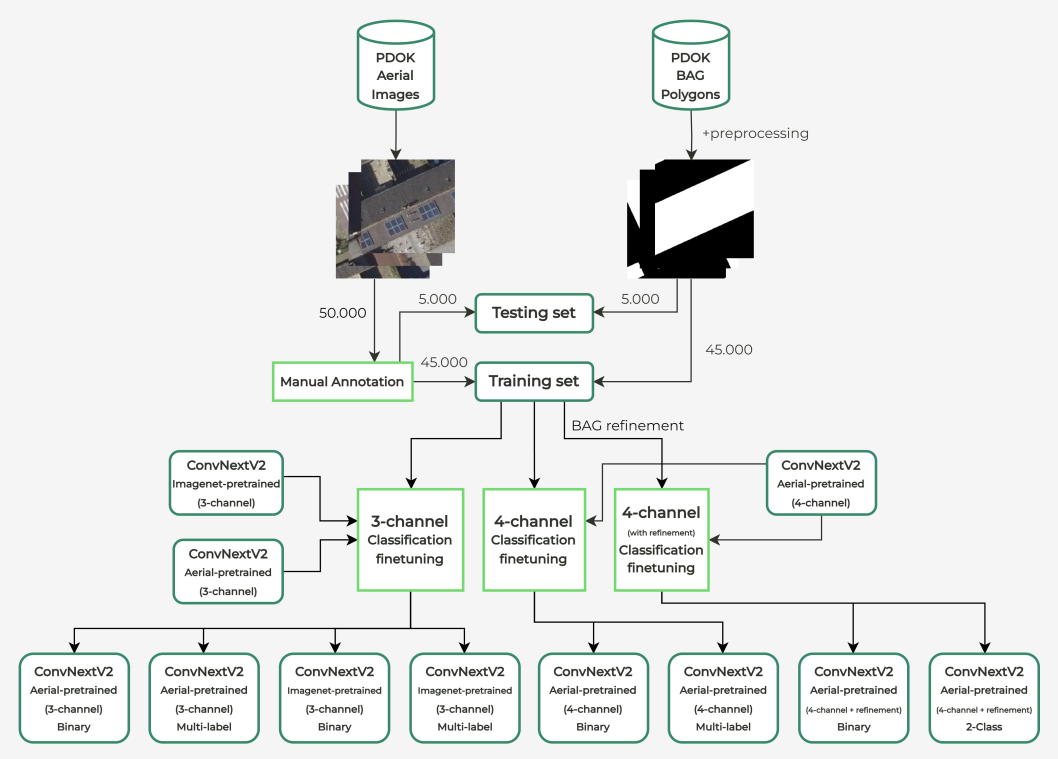

Figure 6.2: The classification network variations that were trained, and the resources used per finetuning task. The pretrained weights used for finetuning are taken from the pretraining phase.

For all finetuning tasks, the same hyperparameters were used. Base learning rate was set to 6.25e-4, with the same linear lr scaling rule as was used in pretraining. Furthermore, weight decay was set to 0.05, and a grouped layer decay of 0.6 was used, again in accordance with [25]. AdamW was used as the optimizer, and a batch size of 64 was used. The networks were trained for 15 epochs with 1 warm-up epoch.

6.4 Segmentation

The second step in the proposed pipeline is the segmentation of positively classified images. In order to answer research sub-question 4, this segmentation network was trained in a fully-, semi-, and weakly-supervised manner. The segmentation network used for all experiments consisted of the encoder taken from a classification network, with a DeepLabV3 segmentation decoder on top. Encoder weights were taken from a trained classification network, motivated by the findings of Woo et al. [25] which showed that a segmentation network exhibits better performance when the initialized encoder weights are taken from a finetuned classification network, instead of directly from the MAE pretrained network.

In order to more accurately answer research sub-question 3, the segmentation network was also trained for both binary and multi-label segmentation. For the binary task, 2 output classes were predicted per pixel: background and solar panel. For the multi-label task, 3 output classes were predicted per pixel: background, PV panel, and ST panel.

As it was not feasible to train all the variants of the segmentation network multiple times based on the different finetuned classification network, only a set of 2 finetuned classification encoders were utilised for segmentation training. The 4-channel encoders exhibited inferior classification performance to the 3-channel encoders, thus were not chosen. For the 3-channel encoders, inspection of the resulting CAMs showed the encoders finetuned from the aerial image based pretraining to have learned better image representations (details about these results will be given in Chapter 7). For this reason, the segmentation networks for binary and multi-label segmentation were all trained based on the 3-channel encoders finetuned from the aerial image based pretraining for binary and multi-label classification respectively. Thus, in total 2 segmentation networks were trained, one for binary segmentation and one for multi-label segmentation. Both networks were trained in a fully-, semi-, and weakly-supervised manner.

While Chen et al. [53] showed the segmentation performance of a network with a DeepLabV3 head can be improved by replacing standard convolutions in the encoder with atrous convolutions, this was not implemented in the trained networks. This choice was motivated by the relatively short training time that was feasible in this project, which was not expected to be sufficient to allow the transferred encoder weights to adjust well to the new atrous convolutions.

6.4.1 Fully-supervised segmentation

The baseline to compare semi-supervised and weakly-supervised models with was set by training the segmentation networks in a fully supervised manner. To this end, the 3,000 manually labelled images from the training set were used as input. For a fair comparison to semi- and weakly-supervised training, the models were trained utilising the training strategy from CorrMatch [62], except without utilising any unlabelled data, and a loss function based only on supervised samples. Therefore, the loss function becomes a simple cross entropy loss between the ground truth mask and predicted mask. The network was trained for 100 epochs, which is significantly more than the semi-supervised and weakly-supervised training, although the iterations per epoch are lower. The total iterations per training strategy were therefore kept similar. A batch size of 16 was used, the base learning rate was set to 1e-3, and the same learning decay as in CorrMatch was used: $(1 - \frac{iter}{total\_iter})^{0.9}$. Stochastic Gradient Descent (SGD) was used as the optimizer, again in accordance to the training of CorrMatch.

6.4.2 Semi-supervised segmentation

While only 3,000 training samples were manually labelled at a pixel level, the full training dataset contains 16,466 positive samples. To make use of the 13,466 remaining positive samples with no pixel-level annotations, a semi-supervised training approach was utilised. CorrMatch [62] was used as the semi-supervised training strategy, with a single partition of 3,000:13,466 or roughly $1 : 4.5$. Iterations per epoch are based on the maximum of labelled and unlabelled sample count. Therefore, there were 13,466 iterations per epoch in this case. As the supervised case was trained for 100 epochs, corresponding to 300,000 iterations[2], the semi-supervised case was trained for 22 epochs, corresponding to 296,252 iterations[2]. A batch size of 8 was used, and all other hyperparameters were kept the same as in the fully-supervised case.

6.4.3 Weakly-supervised segmentation

Finally, a weakly-supervised approach was proposed and tested. The weakly-supervised approach combined CAMs from greedily retrained classification net-

[2]The actual amount of iterations is different, due to batch size. But since the batch sizes differ between the fully- and semi-supervised case, a batch size of 1 is assumed here to simplify calculations.

works with SAM segmentations to generate pseudo-labels. The pseudo-labels were then scored based on confidence estimated by finding the IoU between the pseudo-mask and CAM binary mask. Based on this confidence, different partitions of the pseudo-labels were created partitioning in labelled high-confidence pseudo-labels and images used as unlabelled in a semi-supervised training manner. This combination of weak pseudo-label generation and semi-supervised learning with subsets of the pseudo-labels could be referred to, as done by [75], as weakly-semi-supervised segmentation. The remainder of this subsection will detail the pseudo-label generation, selection strategy, and semi-supervised training approach. The full pseudo-label generation strategy is visualised in Figure 6.3.

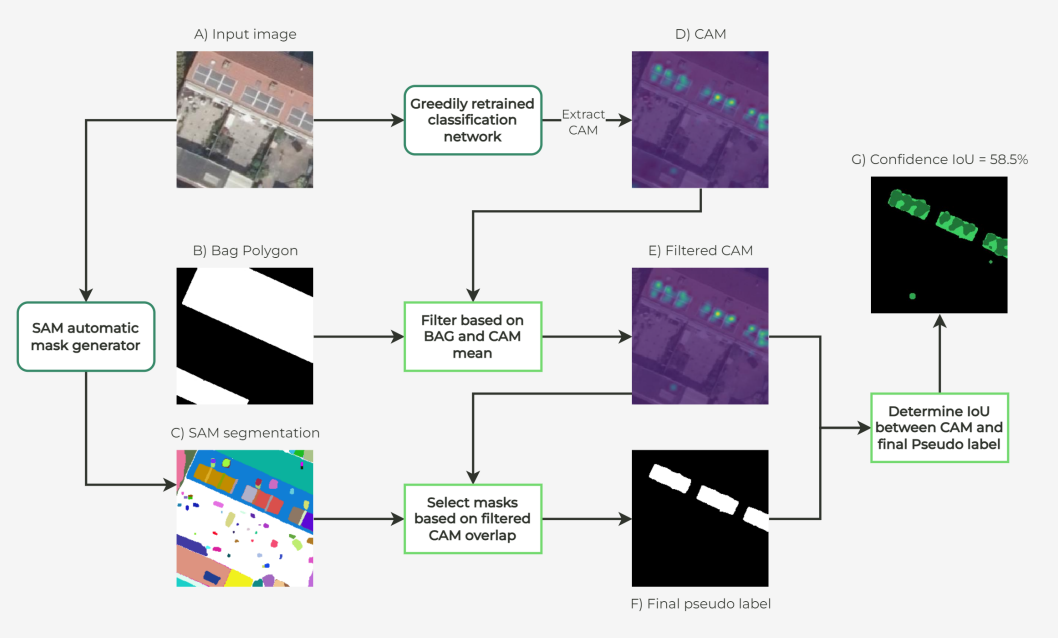

Figure 6.3: An overview of the pseudo-label generation strategy. An input image (A) is classified with the greedily retrained network. From this prediction, a CAM is extracted (D). The CAM is then filtered such that only activation within BAG polygon (B) remains which is higher than 1.5 times the mean of the CAM values (E). This CAM is then used to select masks from the SAM generated masks (C) to create the pseudo label (F). Finally, to compute a confidence score for each pseudo-label, the filtered CAM is compared to the final pseudo-label, and the IoU is computed (G).

Class Activation Map refinement

As demonstrated by Yu et al. [37] in DeepSolar, CAMs computed near the final layers of a network are often not precise enough to locate solar panels. Computing the CAMs at an earlier stage produces finer grained results, but also tends to produce a lot of noise. By adding one or multiple freshly initialized layers after one of the earlier layers, and training this layer on a classification task while keeping the other

layers frozen, the CAMs produced by the network become more precise, while still being able to take advantage of the low level features to locate every individual solar panel across the image. This approach was applied to the classification network with some adjustments discussed below.

The ConvNeXt V2 backbone consists of 4 stages, with a downsampling layer between each stage. It was found experimentally that the best trade-off between higher-resolution feature maps at the beginning of the network and an increased learning capacity at the end of the network could be found by inserting a refinement layer after the second stage, before downsampling for the third stage. The simple convolutional layer used in DeepSolar was tested, but adding a ConvNeXt block was favoured as it produced better quality CAMs. Another change made to the DeepSolar approach was the insertion of only a single greedily trained layer, instead of two. This greedy network was then trained for 25 epochs on the classification training set with a fixed learning rate of 5e-4, other hyperparameters were identical to the classification training setting. All weights in the network were frozen except for the newly added layer, in accordance with the DeepSolar method.

Some experimentation was done to further explore methods of refining the CAMs. For example: adding an upsampling layer after the second stage before the greedy layer, using kernel widths of 3 or 5 for the greedy layer, and combinations of these. Based on qualitative evaluation it was decided that besides the standard refinement method, adding an upsampling layer also had potential to improve the quality of the CAMs. Therefore, 3 refined networks were used for generation of pseudo-masks: multi class with standard refinement, binary with standard refinement, and binary with upsampling refinement. Details on the assessment of these refinements can be found in Chapter 7.

SAM mask selection

With the refined CAMs from the greedily trained segmentation branches, a mask selection strategy was designed to select SAM generated masks.

For each image, the SAM automatic mask generator [88] was run. IoU prediction threshold was lowered to 0.8 to generate additional masks and improve recall, and the refined BAG polygons were used to filter out all generated masks that did not overlap with a BAG polygon.

Next, a selection strategy was designed to select the masks from the SAM gen-

erated set that were most likely to cover a solar panel. To select masks, the refined CAMs were utilised. The strategy was as follows: for each image, a CAM was computed, and filtered based on the BAG mask corresponding to the image (Equation 6.1). The resulting CAM was thresholded, such that all values lower than 1.5 times the mean of the remaining values were set to zero (Equation 6.2). This was done to ensure that only high confidence regions from the CAM were used to select masks. Next, for each mask in the SAM generated set, an intersection ratio with the remaining CAM was computed (Equation 6.3). Note that this intersection is weighted based on the amount of activation. The value of the intersection was then divided by the area of the mask to compute an intersection ratio. All masks for which this ratio was higher than 0.05 were selected. All selected masks were combined into the final pseudo-label for the image.

$$FCAM = CAM \odot BAG \tag{6.1}$$

$$TCAM_{i,j} = \begin{cases} FCAM_{i,j}, & \text{if } FCAM_{i,j} \geq 1.5 * mean(FCAM) \\ 0, & \text{otherwise} \end{cases} \tag{6.2}$$

$$IntersectionRatio = Sum(SAMMask \odot TCAM) / Area(SAMMask) \tag{6.3}$$

In the above equations, $FCAM$ is short for Filtered CAM, and $TCAM$ is short for Thresholded CAM.

Adding a semi-supervised segmentation network

As shown by [75], pseudo-labels generated utilising CAMs to select masks from SAM generated masks can be of poor quality in many cases. To mitigate this issue, semi-supervised segmentation approaches can be utilised to use only the high quality pseudo-labels to train, and discard other pseudo-labels. This combination of semi-supervised learning and weakly generated pseudo labels will be referred to as weakly-semi-supervised learning. CorrMatch [62] was also utilised here as it showed better performance than UniMatch, which was utilised in a similar fashion in [75].

Pseudo-label selection strategy

Not all pseudo-labels are of equal quality, but since the ground truth is not available in weakly supervised learning, a metric is required to determine the confidence per pseudo-label. The metric used is as follows: the CAM of the image was once again utilised, but this time to compare to the full pseudo-label. The thresholded CAM (Figure 6.3 E) from the SAM mask selection strategy was binarised such that all values higher than 0.1 are set to 1, and an IoU of the resulting binary CAM mask was computed with the full pseudo-label (Figure 6.3 F) to determine a confidence value (Figure 6.3 G). This process was repeated for all masks, and all images were sorted in descending order based on this CAM IoU. Depending on the partition ratio, the top x% of the images were then selected as labelled instances for semi-supervised training, and the rest were selected as unlabelled samples.

Partitions

With the proposed weakly-supervised approach, the entire set could be utilised to generate pseudo-labels. With the pseudo-label selection strategy discussed above, the set is instead partitioned in a high-confidence pseudo-labelled set, and an unlabelled set. Different partition ratios were tested, to investigate the optimal trade-off between the quality of the labels and the quantity of the labels. The following partition distributions were tested (defined as *labelled:unlabelled*) : 2:1, 1:1, 1:2, 1:4.

Training

Semi supervised training on weakly generated pseudo-labels was performed with the same setup as used in the semi-supervised training described in Section 6.4.2. Epochs were calculated again such that the overall iteration count was roughly equal between all experiments. The epoch count per partition was therefore 25, 33, 25, and 21 for the 2:1, 1:1, 1:2, and 1:4 partitions respectively. For the binary cases, a batch size of 8 was used, and for the multi-label cases a batch size of 4 was used.

7. Results

This chapter aims to report the findings of this thesis project, divided into the different components of the proposed architecture. Results will be presented in the order in which components of the architecture were developed and trained. First, results from the pretraining stage will be presented by qualitative analysis of a sample of images reconstructed by the trained network from a masked image. Next, results of finetuning of the pretrained network on the classification task will be discussed. A qualitative analysis of the different approaches to CAM refinement from these finetuned networks will be given, highlighting the motivation for choosing only a subset of the refined methods to be used for weakly supervised pseudo label generation. Finally, the result of the segmentation task will be presented on all three training approaches: fully-, semi- and weakly-supervised segmentation.

7.1 Pretraining

Pretraining was done in a self-supervised manner with the use of masked autoencoders. The full unlabelled dataset was utilised for pretraining, with the network learning to reconstruct the masked regions of the images. Pretraining performance itself is not of interest, as instead the impact of pretraining on classification and segmentation performance is the metric with which the importance of pretraining can be measured. However, a set of reconstructed images for both pretraining regiments will be given to provide an impression of the degree to which the network has learned valuable representations of the data.

Figure 7.1 illustrates the 3-channel network reconstructing the masked regions of the RGB images. Visual inspection of the reconstructed images shows the network is able to extrapolate large shapes such as rooftops, and local patters such as a grid of solar panels. It struggles however to create an image that is convincing as a real aerial image, as regions with no unmasked patches nearby are noisy in the output. Similar results can be found in Figure 7.2, depicting the 4-channel network reconstructing both the RGB image and corresponding BAG polygon. In the 4-channel case however, it is clear that the shape of the BAG polygon can be

Figure 7.1: Results of the 3-channel MAE pretraining stage.

predicted very convincingly, with the network able to predict the shape of the polygon with high accuracy. Since the BAG channel is a simple binary mask, it can be expected that the network would be able to solve this task well. The difference between the RGB and BAG channel predictions therefore indicates the reason for inferior prediction performance on the RGB channels can most likely be attributed to the complexity of the target images. Aerial images can contain a large variety of textures and patterns, making prediction of regions with no close unmasked patches difficult.

Figure 7.2: Results of the 4-channel MAE pretraining stage. RGB channels have been separated from the BAG channel for visualisation purposes. The BAG channel is shown subsequently to each RGB sample, although in the network, all four channels are inputted and predicted simultaneously.

7.2 Finetuning

With the MAE pretrained networks, the classification network could now be finetuned on the labelled dataset. This section contains the results gathered from the finetuned classification networks. To evaluate the networks, an F1 score is computed per target class and used as the main metric. Additionally, precision and recall are presented per class.

7.2.1 Binary Classification

Table 7.1: Results of the binary classification task training. Highest performance is underlined.

Channels	Pretraining	Precision	Recall	F1
RGB	ImageNet	<u>0.964</u>	<u>0.945</u>	<u>0.954</u>
	Aerial images	0.953	0.938	0.946
RGB + BAG	Aerial images + BAG	0.916	0.909	0.913
RGB + Refined BAG	Aerial images + BAG	0.919	0.915	0.917

Table 7.1 shows results of the binary classification finetuning on the labelled dataset. The first notable result is the inferior performance of the two networks utilising BAG and refined BAG data as a 4th channel of the network. While still able to classify solar panels with a high F1 score, it is clear that dividing the learning capacity of the network over 4 channels instead of the default 3 RGB channels deteriorates performance. Secondly, the network utilising the ImageNet pretrained weights slightly outperforms the aerial image pretrained network. Finally, it can also be noted that the network trained utilising refined BAG images does not show a significant improvement over the network trained utilising unmodified BAG images.

7.2.2 Multi-label Classification

The multi-label classification results presented in Table 7.2 show similar result trends as were found for the binary classification task. Once again, the ImageNet pretrained network outperforms the aerial image pretrained network, and the 4-channel networks perform worse overall. Furthermore, it is clear that all networks are less capable of classifying images with the ST label than those with a PV label. In this multi-label task, it is also clear that refinement of BAG polygons has no significant impact on the performance of the network, and in this case even modestly

Table 7.2: Results of the multi-label classification task training. Highest performance is underlined.

Channels	Pretraining	Class	Precision	Recall	F1
RGB	ImageNet	PV	<u>0.963</u>	<u>0.952</u>	<u>0.957</u>
		ST	0.846	0.673	0.749
	Aerial images	PV	0.952	0.933	0.943
		ST	0.730	0.602	0.660
RGB + BAG	Aerial images + BAG	PV	0.912	0.894	0.903
		ST	0.675	0.450	0.540
RGB + Refined BAG	Aerial images + BAG	PV	0.909	0.893	0.900
		ST	0.617	0.461	0.528

decreases performance compared to using the unrefined BAG polygons.

7.3 CAM Refinement

The results from the finetuning step on the classification task showed that utilising BAG as a 4th input channel did not improve performance. Therefore, the greedy retraining step was performed utilising only the 3-channel networks.

Of the 3-channel networks, greedy retraining was tested both on the network finetuned from ImageNet weights and the network finetuned from aerial image weights. Figure 7.3 illustrates a qualitative analysis of the difference in CAMs generated by the two networks with and without greedy retraining. While the CAMs generated directly from the finetuned networks clearly show activation appearing over areas of the image containing solar panels, the greedily retrained networks show a much more focused activation. The greedily retrained networks are also able to more evenly divide the activation over the different parts of the image, meaning more solar panels are covered by at least a portion of activation. This is a desirable trait for pseudo label generation from SAM, as it should be possible to recall a large mask if only a part of contains activation, but solar panel masks without activation will not be able to be located.

Comparing the two greedily retrained networks, it is clear that the network finetuned from ImageNet weights has less focussed CAMs after retraining compared to the network finetuned from aerial image weights. This is likely due to the fact that the low level features learned during pretraining on ImageNet are more general and therefore less specialised for this target domain. For example, the network finetuned from ImageNet weights shows larger activation near the borders of solar

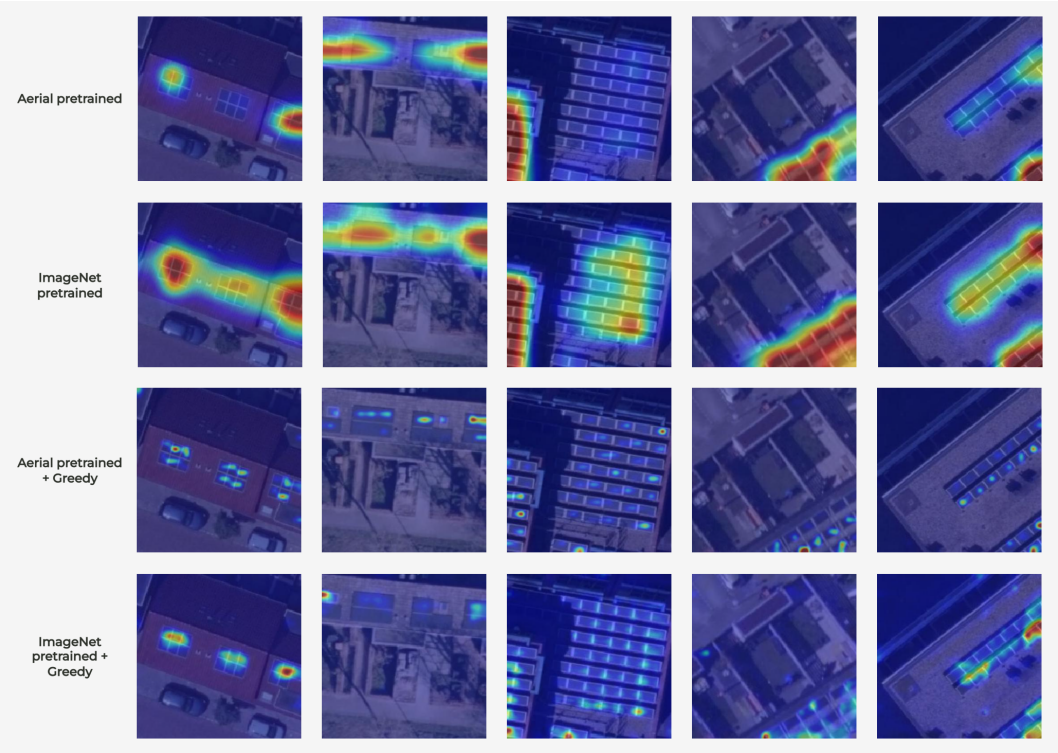

Figure 7.3: Class Activation Maps generated from the non-refined networks and the greedily retrained networks. For both refined and non-refined networks, results are shown for the network based on ImageNet pretrained weights and the network based on aerial image pretrained weights.

panels in an image with many panels than on the actual panels themselves. The network retrained from aerial image weights however shows activation at the centre of many solar panels, indicating the network has learned to focus on the actual target object.

Based on the above results, it was decided to continue with the network fine-tuned from aerial image weights for further experimentation. Different modifications to the network were experimented with. Figure 7.4 illustrates results of some of these modifications. Lowering the kernel size of the first convolutional layer in the greedy layer added on top of the first two stages of the network showed a slight improvement in the localization of panels. It can be observed that the network is able to locate individual panels with a slightly higher accuracy than with the original kernel width of 7. Adding an upsampling layer before the greedy layer also showed a modest improvement, especially in the location of panels in an image with numerous panels. For this reason, weakly supervised pseudo label generation was performed based on two greedily retrained networks: a network retrained with a greedy layer consisting of a single standard ConvNeXt block, and a network

retrained with a greedy layer consisting of an upsampling layer and a standard ConvNeXt block.

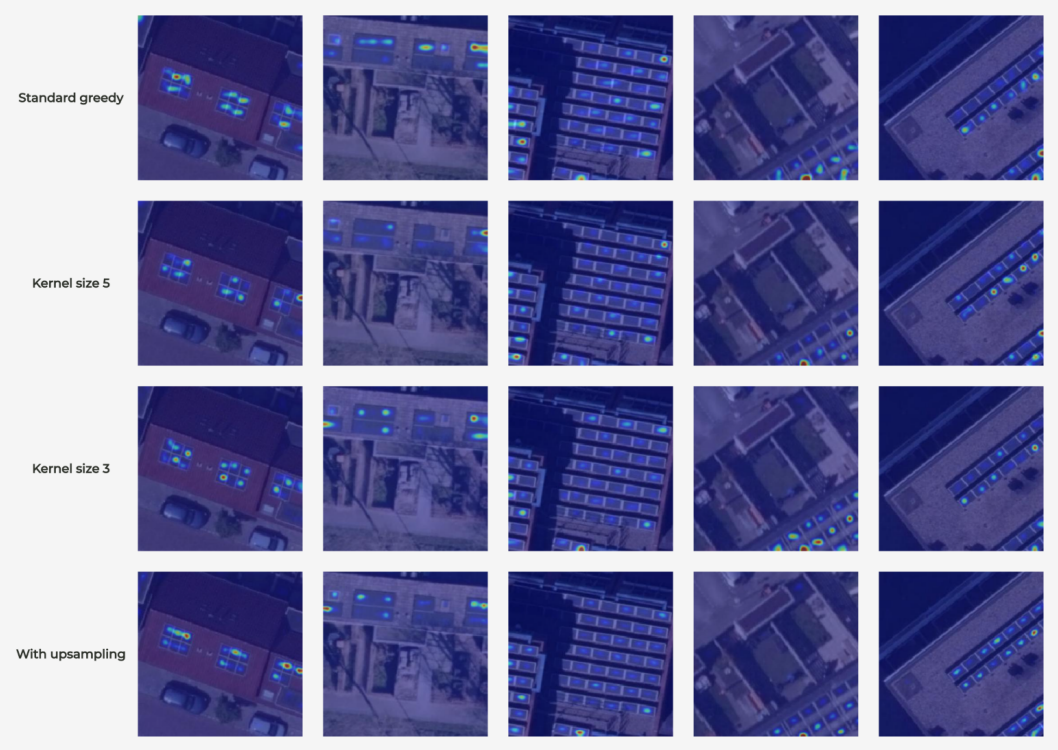

Figure 7.4: Class activation maps generated from networks with different modifications to the greedy layer. Small differences in activation can be observed. For example, in the fourth column there are panels where the standard greedy approach shows no activation, but the network with the kernel size of the first convolutional layer reduced to 3 shows activation. Another example of differing performance can be found in the last column, where the network with an upsampling layer before the greedy layer shows activation on almost all panels, whereas the other approaches still miss a few panels of the array.

7.4 Segmentation

The final step in the pipeline was the segmentation task. In this section results for fully-, semi, and weakly-supervised segmentation will be presented. For all networks, IoU, precision, and recall will be presented per class. Precision and recall are computed on a pixel basis, meaning precision is the amount of correct positive pixel predictions compared to the total amount of positive predictions, and recall is the amount of correct positive predictions compared to the total amount of actual positive pixels.

Table 7.3: Results of the binary segmentation task training. Best performance is underlined. "Ratio" refers to the partitioning ratio of labelled:unlabelled samples used for semi-supervised learning.

Labels	Training strategy	Precision	Recall	IoU
Weak SAM + CAM	*None* [3]	0.661	0.667	0.497
Weak SAM + Upsampled CAM	*None* [3]	0.713	0.613	0.492
Weak SAM + CAM	Semi-supervised - ratio 2:1	0.779	0.722	0.599
	Semi-supervised - ratio 1:1	0.794	0.702	0.594
	Semi-supervised - ratio 1:2	0.819	0.655	0.572
	Semi-supervised - ratio 1:4	0.809	0.620	0.540
Weak SAM + Upsampled CAM	Semi-supervised - ratio 2:1	0.795	0.627	0.543
	Semi-supervised - ratio 1:1	0.831	0.637	0.564
	Semi-supervised - ratio 1:2	0.856	0.594	0.540
	Semi-supervised - ratio 1:4	<u>0.879</u>	0.535	0.498
Manually annotated	Fully-supervised	0.847	0.806	0.703
	Semi-supervised - ratio 3000:13865	0.878	<u>0.816</u>	<u>0.733</u>

7.4.1 Binary segmentation

Results of binary segmentation utilising different training strategies are presented in Table 7.3. It is immediately clear that training on manually annotated labels outperforms weakly supervised methods by a large margin, which can be expected given the obvious advantage of learning on near perfect labels compared to noisy pseudo labels. However, utilising the set of positive samples without pixel level training annotations in a semi-supervised setting is also shown to improve performance significantly with an increase in IoU of 3%. Training on the pseudo labels also shows a significant improvement compared to directly utilising the pseudo labels as output, indicating that utilising SAM in combination with refined CAMs alone is not yet a viable strategy for binary segmentation. Visualisation of the masks generated by the different networks can be found in Figure 7.5.

Precision is higher than recall for all networks, highlighting the fact that solar panels have a tendency to blend in with the environment, i.e. by having similar textures to the surrounding rooftops. Surprisingly, the highest precision is achieved by a weakly-semi-supervised network, trained on pseudo labels generated from CAMs of the network with an upsampling layer before the greedy layer. While the other metrics show that the network trained on pseudo-labels originating from the

[3] Results for this column are the scores obtained when performing the strategy used for pseudo label generation directly on the test set. It therefore measures the quality of the pseudo labels as if they were utilised directly as output.

unmodified greedily trained network still outperforms this network, it shows that there is potential for the upsampling layer to improve the quality of the pseudo labels. Precision also improves when increasing the partition ratio of labelled and unlabelled images utilised during training, which can be attributed to the higher quality of the pseudo labels in the labelled set. Recall decreases however as the ratio increases, which likely indicates that the network requires a larger number of labelled samples to learn to generalise over the different types of solar panels properly.

Finally, comparing different labelled to unlabelled ratios for partitioning the pseudo labels based on confidence scores, it can be noted that for the best performance, at least half of the pseudo labels should be utilised as labelled samples. For both pseudo label types, the performance starts to decrease as the partition ratio increases beyond 1:1. For the unmodified greedy CAM based pseudo labels, the best performance is found at the 2:1 ratio, with a slight improvement over the 1:1 ratio. In the case of pseudo labels generated from CAMs with upsampling, the best performance is obtained when splitting the pseudo label set with a ratio of 1:1, with decreasing performance for both increasing and decreasing the ratio. This observation highlights the importance of scoring pseudo labels and utilising semi-supervised learning in combination with weakly-supervised pseudo labels, as it is clear that learning only on pseudo labels undermines the potential performance. This is due to the fact that the network will spend time learning on the low quality labels, harming the training process.

7.4.2 Multi-label segmentation

In the case of multi-label segmentation, of which results are shown in Table 7.4, similar patterns emerge to those found in the binary segmentation task. The networks trained on manually annotated labels show a large increase in performance compared to the weakly supervised networks. Again, semi-supervised learning utilising the positive samples that were not manually labelled as unlabelled data besides the manually annotated samples shows a significant improvement in performance. This is especially true for the IoU of the PV class, which is increased by 6.1% compared to the fully-supervised network. It is interesting to note however, that while the mean IoU for the semi-supervised network is higher than that of the fully-supervised network, IoU for the ST class is slightly lower. Recall of ST panels increases drastically in the semi-supervised case, but at the cost of a significant

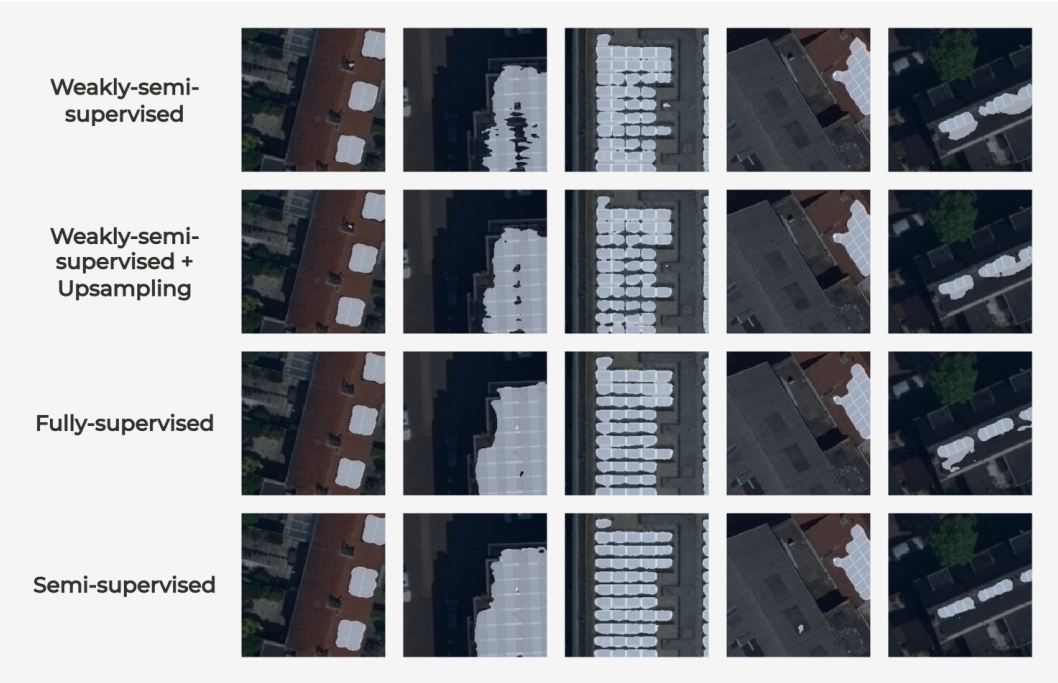

Figure 7.5: Output masks of the different networks. For the weakly-semi-supervised networks, the network corresponding to the partition ratio with the best performance is shown. It is clear that the weakly-semi-supervised networks produce output masks with a lower quality than the fully-supervised network, mainly due to the fact that they are unable to precisely segment the border of the panels. Between fully-, and semi-supervised, the semi-supervised case is slightly more precise.

Table 7.4: Results of the multi-label segmentation task training. Best performance is underlined.

Labels	Training strategy	Class	Precision	Recall	IoU
Weak SAM + CAM	Semi-supervised - ratio 2:1	PV	0.783	0.677	0.570
		ST	0.576	0.117	0.106
	Semi-supervised - ratio 1:1	PV	0.778	0.712	0.591
		ST	0.408	0.135	0.113
	Semi-supervised - ratio 1:2	PV	0.820	0.636	0.558
		ST	0.523	0.111	0.100
	Semi-supervised - ratio 1:4	PV	0.841	0.588	0.529
		ST	0.617	0.079	0.075
Manually annotated	Fully-supervised	PV	0.843	0.816	0.709
		ST	0.756	0.344	0.387
	Semi-supervised - ratio 3000:13865	PV	<u>0.890</u>	<u>0.851</u>	<u>0.770</u>
		ST	0.658	0.467	0.376

drop in precision.

In the weakly-semi-supervised networks similar patterns to the binary case can also be observed. Utilising a 1:1 split on the pseudo labels yields the highest performance, showing once more that increasing the amount of labelled data can actually

harm performance if the labels themselves are of a lower quality. For the ST class, performance in weakly-semi-supervised learning is especially low, with an IoU of only 11.3% in the best case, and 7.5% on the worst scoring partition. Output visualisations of the different networks can be found in Figure 7.6.

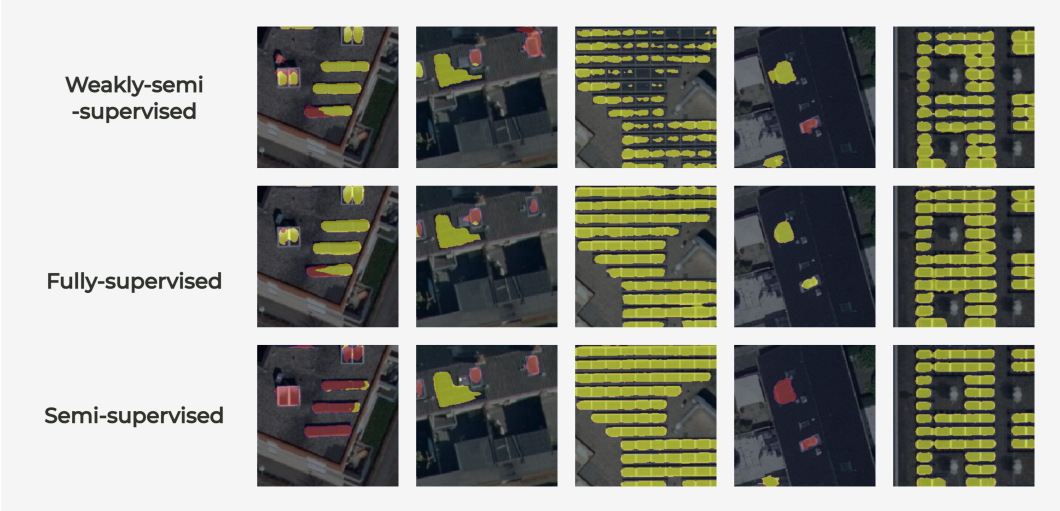

Figure 7.6: Output masks of the different networks for the multi-label task. PV predictions are marked in yellow, ST predictions are marked in red. On images with both PV and ST panels (columns 1 and 2), all networks struggle to make confident predictions, although the semi-supervised model clearly performs best. In cases with a single class (columns 3 and 5), the weakly-semi-supervised model shows a clear degradation in performance, whereas the fully- and semi-supervised models still perform well. In some ambiguous cases such as column 4, all models might have varying predictions that are not in line with each other.

8. Discussion

In this final chapter, the findings of this thesis project will be discussed and reflected upon. Based on these finding, the research questions proposed at the beginning of this thesis will be answered as well as possible. Afterwards, an analysis of the limitations of the project will be given, along with suggestions for further research based on these limitations as well as other observations. Finally, the chapter will be concluded with a summary of the main findings and conclusions of this thesis project.

8.1 Findings

The findings of this thesis project will now be summarised and discussed in the order in which they are presented in the results section. When applicable, the findings will be compared to related findings in literature. Additionally, an analysis of the findings will be given regarding possible explanations for the results, and the conclusions that can be drawn from them.

8.1.1 Pretraining

The first step in the design of the model was the pretraining stage, utilising masked auto-encoding modified for ConvNeXt V2 as the self-supervised training method. Reconstructions of masked images produced by the trained masked auto-encoder showed the network had learned to extrapolate common patterns in the aerial images from the unmasked patches into the masked patches. This result indicates a high likelihood that the pretraining stage has allowed the encoder side of the model to learn relevant low level features, which could then be extrapolated by the decoder to predict the masked patches.

The authors of ConvNeXt V2 unfortunately do not provide examples of their pretrained MAE reconstructing images, nor the decoder weights required to reproduce the trained MAE. However, the results of the MAE trained in this project can be compared to that of the MAE trained in the original paper on Masked Auto-Encoders as a pretraining strategy for ViTs [22] (referred to from here on as ViT-

MAE, as in their work). Figure 8.1 illustrates a comparison of reconstructions in the ViTMAE paper, and the reconstructions of the ConvNeXt V2 MAE in this project.

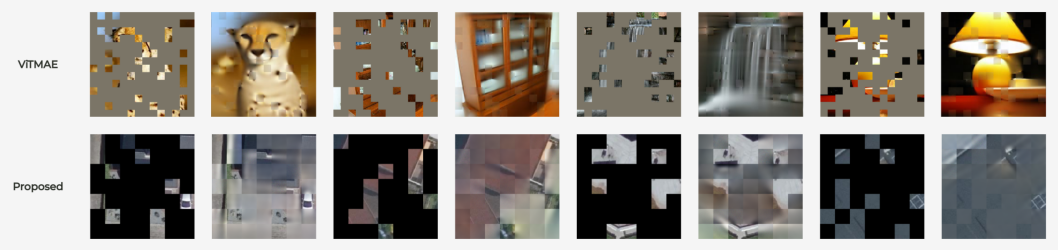

Figure 8.1: Comparison of reconstructions of images from ImageNet with a vision transformer masked auto encoder [22] (top) and aerial images with the ConvNeXt V2 masked auto encoder trained in this project (bottom).

A noteworthy difference is the slightly more precise prediction of object shapes in the ViTMAE reconstructions. A possible explanation for this might be the difference in the training set of the models, where images from ImageNet used in ViTMAE contain a large variety of well-separable objects in the images, whereas the aerial images used in this project contain mostly buildings, roads, and vegetation. Especially buildings can have extremely varying shapes and textures on their rooftops, making precise prediction difficult. A second possible explanation is the difference in training time, as the pretraining stage in this project was limited to 50 epochs due to time constraints, comparing to the 800 epochs for which the ViTMAE was trained.

Nevertheless, the results of pretraining in this project showed that pretraining ConvNeXt V2 on images in the target domain can be beneficial for performance of classification and segmentation, and especially for the generation of pseudo labels based on CAMs in a weakly supervised setting.

8.1.2 Classification

Classification of PV and ST panels in the binary case proved to be a feasible task, with an F1 score of 95.4% in the highest case. This is in line with previous solar panel classification studies utilising deep learning [37], [39], [42], [44]. Precision was found to be higher than recall in all networks, highlighting the difficulty of detecting solar panels in specific background, such as was analysed as well by Lindahl et al. [42].

The difference between finetuning on an ImageNet pretrained network and an aerial image pretrained network was found to be minimal, with a slight advantage

for the ImageNet pretrained model. This might be due to the low level features learned by the ImageNet network being more general and therefore more useful for the classification task. Alternatively, this may also again be due to the difference in pretraining time, as the ImageNet pretrained model was pretrained for 800 epochs on more than twice as many images than the aerial image pretrained model, which was only pretrained for 50 epochs.

Utilisation of BAG polygons as a 4th input channel was found not to improve classification performance, but instead slightly deteriorate scores instead. While one would assume building location to be a useful input for the model, the consideration should be made if the computational power of the network assigned to processing of the channel does not outweigh the benefit of the information contained in the channel. The dimensions of feature maps throughout the network is equal after the stem in both 3-channel and 4-channel network, implying that the network must distribute its learning capacity over more features in the 4-channel network. If the information in the BAG channel is then not sufficiently useful, or could potentially also be retrieved from the RGB channels, it is not surprising to observe a decrease in performance when adding the channel.

Finetuning of BAG polygons did not help to remedy the decrease in performance of the 4-channel network, which indicates that the cause of the performance decrease lies not in the slight misalignment of polygons in the BAG masks. Performance instead is near identical when using refined BAG polygons compared to the regular BAG polygons, which also seems to suggest the BAG channel is not utilised much overall by the networks.

Finally, classification of PV and ST panels in a multi-label fashion was also found to be a feasible task, although performance on the ST class was significantly lower than for the PV class. Besides the obvious disadvantage of significantly less positive samples in the training set for ST panels than for PV panels which seems to hinder recall, difficulty may also lie in the distinction of ST panels from PV panels and vice versa, as they have similar appearances. Contrarily, F1 scores for the PV class individually were higher in the best performing model than the F1 score found on the binary class. These findings are once again in line with those of Lindahl et al. [42], who also found ST panels to be more difficult to classify than PV panels.

8.1.3 CAM Refinement

Expanding on the work of Yu et al. [37] on DeepSolar, a CAM refinement strategy was proposed based on adding and retraining a greedy branch from one of the lower level stages of the finetuned classification model. During the design process of the pipeline however, it was found that simply adding 2 simple convolutional layers identical to the DeepSolar approach led to unsatisfying results. Instead, a full ConvNeXt V2 block was added as the greedy branch. In order to further explore the potential of greedy retraining for CAM refinement, a handful of other adjustments were also experimented with.

In contrast to DeepSolar, where greedy retraining led to an improvement in recall of PV pixels when using the CAMs after refinement for segmentation, greedy retraining with a ConvNeXt V2 block was found to increase the precision of the CAMs and decrease the overall activation. A detailed explanation of this phenomenon is not immediately clear, although it might be due to the GELU unit before the final pointwise convolution, which propagates a negative value for slightly negative logits, leading to a sparser activation map.

If the CAMs produced would have been utilised directly for segmentation, this decrease in recall would have been detrimental. However, since the motivation behind refining the CAMs was that they could be used to select masks from a SAM segmentation of each image, the increase in precision was found to be beneficial.

CAM refinement was tested both on a network finetuned from the ImageNet pretrained model, as well as on a network finetuned from the aerial image pretrained model. An interesting observation here was that CAMs produced by the greedily retrained network based on aerial image pretraining showed increased activation at more precise locations than for the alternative based on ImageNet. The most likely cause is the fact that the encoder trained on ImageNet has learned to extract very general features, which are useful for classification of a wide variety of objects, but not necessarily for the specific task of detecting solar panels. Instead, the encoder trained on aerial images has learned to extract features often present in aerial images, of which solar panels are a part. The ImageNet based network showed activation mostly on the edges of solar panels, where a light frame characterises many panels. The aerial image based network displayed higher activation in the centre of the panels, which is more suitable for selecting masks from SAM segmentations. See Figure 8.2 for a comparison of CAMs produced by the two

networks.

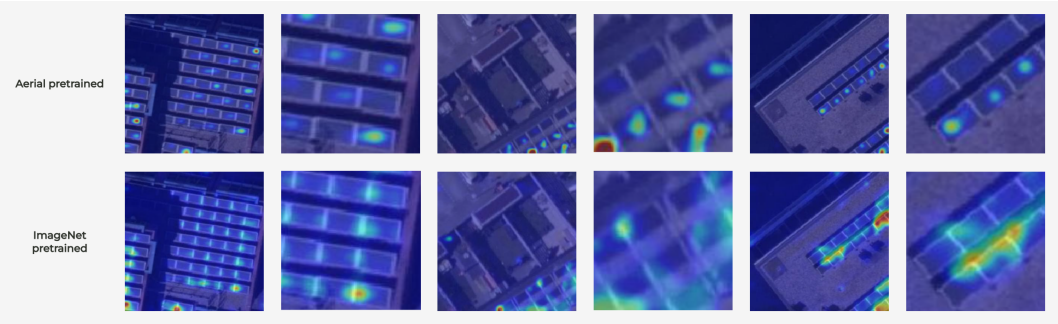

Figure 8.2: Comparison of activation location in CAMs generated from greedily retrained networks based on aerial image pretrained weights (top) and ImageNet pretrained weights (bottom). 3 Samples are illustrated, with a zoomed portion of the image following every sample.

Experimentation with variations in the greedy block found only minimal differences in the activation maps produced. Changing the kernel size of the depth wise convolution from 7 to 3 or 5 made activation slightly more precise, but adding an upsampling layer before adding the greedy block improved precision even more. The upsampling layer might help the greedy block to operate on a higher resolution feature map, although without a skip layer from the earlier stage there may be difficulties in properly locating the desired features in the upsampled feature map. Nevertheless, the greedy network with upsampling was utilised in the final pipeline for CAM refinement along with the network greedily retrained with only a single ConvNeXt block.

The experimentation with different greedy CAM refinement techniques add to works such as DeepSolar [37] and the work of Yang et al. [75] in showing that CAMs are suitable for weakly supervised segmentation, although a refinement step or specialised CAM generation technique are often key to achieving good results.

8.1.4 Segmentation

The final components of the pipeline were the segmentation models trained in a fully-, semi-, and weakly-supervised manner. The findings will be discussed in the aforementioned order.

Fully-supervised segmentation

Fully supervised segmentation for PV systems is a task which has known success in recent literature. IoU scores on this task have been presented by Malof et al. (66%-

69%) [66], Zech and Ranalli (69%) [71], Parhar et al. (82%) [72], and Zhuang et al. (75%) [74], among others. Comparing to the reported scores, the fully supervised network for the binary task presented in this work performs similarly to that of the earlier works with an IoU score of 70.3%. The model is still outperformed by some other recent works, although a direct comparison is not possible since the models are all trained and evaluated on differing datasets with different image qualities, label distributions, and source locations.

The multi-label task displays a worse performance when taking the mean IoU over all classes, although PV IoU is higher than results for the binary task. This pattern is found for the classification task as well, showing once more that the ST class is more difficult to detect than the PV class. Once again, precision outweighs recall in segmentation for both the binary and the multi-label case.

Semi-supervised segmentation

Utilising the remaining unlabelled positive samples in the training set for semi-supervised segmentation resulted in a significant boost in segmentation performance for both the binary and the multi-label task. The IoU score for the binary task increased by 3% to 73.3%, while the IoU for the PV class in the multi-label class increase by 6.1% to 77.0%. The ST class saw a slight decrease in IoU when training with semi-supervised learning, although recall increased drastically.

It can therefore be concluded that CorrMatch is a suitable method for semi-supervised solar panel segmentation. The effectiveness of the method can most likely be attributed to the homogeneous texture of the target class. Pixels representing solar panels can be expected to have high correlation between weakly and strongly perturbed versions, as the texture of the surface of the panel is mostly uniform. The refinement of pseudo labels with correlation regions is therefore expected to be very effective, improving the information that can be learned from each unlabelled sample.

Weakly-supervised segmentation

Weakly supervised segmentation of solar panels is a task which has only been explored by a handful of studies in recent literature. The first of these was the work by Yu et al. on DeepSolar [37], who refined CAMs via greedy retraining of the network and used these refined CAMs directly as segmentation output. Yang et

al. [75] recently showed that there is a benefit in using a specialised CAM, namely EigenCAM, and training a dedicated segmentation network on pseudo labels generated from a combination of SAM segmentations and these CAMs. Additionally, they showed including a semi-supervised training regiments that only uses high quality pseudo labels as supervised samples, and the remaining samples as unsupervised samples can further improve performance.

In this work, the ideas of these studies were combined, by using a similar greedy refinement method to DeepSolar to refine standard CAM generation, and using these refined CAMs to select masks from SAM segmentations for training a segmentation network. A simple yet effective metric was then proposed to determine the confidence in the generated pseudo-labels, which was used to then create partitions of labelled and unlabelled samples for semi-supervised training.

Results of the weakly-supervised semantic segmentations showed worse performance than that of the fully-, and semi-supervised models, though this is to be expected. The IoU scores that were reached still showed the potential of the method, with the best model reaching an IoU of 59.9% for the binary task. Compared to the work of Yang et al. [75], who reported an IoU of 73.5% on the best performing model, this result is significantly lower. A fair comparison is however once again hard to make, as the datasets and training duration differ here too.

Utilisation of the confidence score for partitioning is however shown to be beneficial, as on average the models with a labelled:unlabelled partition of 1:1 outperform the 2:1 partition. This shows that it is better to discard low confidence pseudo labels and use them as unlabelled samples than to include as many pseudo labels as possible in the labelled set.

Finally, it is found that using an upsampling layer before adding a greedy layer in greedy retraining for CAM refinement does not necessarily produce better segmentation results when training on these pseudo-labels. However, it can be noted that precision of both the pseudo labels itself and the segmentation model trained on them is significantly higher for the model with upsampling layers compared to that of the model trained on the pseudo-labels generated from the network greedily retrained without upsampling. When considering the precision-recall trade-off in the pseudo labels generated using the network with an upsampling layer, it seems there is still some room for improvement in the total IoU if the balance was more equal, such as is the case for the pseudo labels generated from the standard greed-

ily retrained network. This might be due to the fact that the thresholding factors used in mask generation strategy were optimised manually for the standard greedy retrained network, and not for the network with the upsampling layer. Thus, shifting values such as the mean thresholding factor or the SAM mask intersection ratio threshold might lead to an increase in performance.

8.2 Revisiting the research questions

The research questions proposed at the beginning of this thesis project will now be revisited and answered based on the findings of this project. Before answering the main research questions, the sub-questions will be addressed in order to then be able to provide a more complete answer to the main questions.

8.2.1 SQ1: To what extent can building location data be utilised in addition to RGB channels to improve the performance of a PV and ST detection model?

As is apparent from the classification training results, adding building location data in the form of BAG polygons as an added input channel does not immediately benefit the performance of the model. In fact, models trained with the BAG masks as fourth input channel perform worse overall compared to the 3-channel networks operating solely on RGB imagery. In initial answer to this questions might therefore be that there is no evidence that building location data can be utilised to improve the performance of a PV and ST detection model. However, in the pseudo label generation pipeline proposed in this thesis, it is shown how BAG polygons can be utilised to filter out activation in CAMs outside of building, increasing the precision of the pseudo labels. While no direct comparison is made between pseudo labels generated with and without this filtering, visual inspection during the design process of the pipeline showed an improvement in the masks when utilising the filtering. Thus, it can be concluded that there is merit in utilising building location data, although not necessarily directly in the model architecture.

8.2.2 SQ2: What is the effect of self-supervised pretraining on a large domain-specific dataset on the performance of a PV and ST detection model?

While classification results comparing models finetuned from an ImageNet pretrained model and an aerial image pretrained model showed a slight performance advantage for the former, later inspection of the features learned via CAM inspection showed that the aerial image pretrained model learned more relevant features for the task at hand. Combining this observation with the fact that the ImageNet pretrained model was trained for a much larger number of epochs and on a much larger dataset, there is reason to believe the aerial image pretrained model has potential to outperform the ImageNet pretrained model with further training. Besides the performance on the classification task, qualitative analysis showed the CAMs after greedy retraining on the aerial image pretrained model were more suitable for pseudo label generation than those based on the ImageNet pretrained model. The effect of aerial pretraining on an PV and ST detection model can therefore be summarised as follows: while the performance on the classification task might not immediately benefit from aerial pretraining with the amount of training that was performed, the features learned by the model are already more relevant for the task at hand with aerial image pretraining, and are more suitable to be used for the weakly supervised version of the problem.

8.2.3 SQ3: To what extent can photovoltaic and solar thermal systems be distinguished by a machine learning model?

On both the classification and the segmentation task, models trained to perform the task in a binary fashion exhibited dominant performance over the models trained for the multi-label task. While performance for PV classification and segmentation showed a similar if not increased performance compared to the binary task, performance of classifying and segmenting the ST class was significantly lower than the joined case. Because of this, the mean performance over both PV and ST classes was lower than for the binary tasks. Therefore, while a model is capable of at least distinguishing between PV and ST systems, it is not able to do so with the same performance as it would be able to classify and segment the joined class of PV and ST systems.

8.2.4 SQ4: What is the performance impact of choosing a semi-supervised or weakly- supervised segmentation approach over a fully-supervised approach for PV and ST segmentation?

The segmentation results for both the binary as the multi-label segmentation tasks show that while a weakly-supervised strategy can be devised to train a model to segment PV and ST panels, it does come at the cost of a rather significant performance penalty. Weakly supervised binary segmentation can reach an IoU of approximately 60% in the best case, while the fully supervised model improves upon that IoU by 10%. Similar differences are found for the multi-label case. However, if unlabelled samples are available besides a manually annotated set of training masks, then a semi-supervised approach can be leveraged to improve the performance by as much as 3% IoU. Besides improving performance of training on manually labelled data, it was also shown that semi-supervised learning techniques can be utilised to boost performance of weakly supervised learning if the pseudo labels can be ordered by a measure of quality.

8.2.5 RQ: To what extent can photovoltaic and solar thermal systems be segmented from Dutch aerial imagery?

From the results, it is clear that a model can be trained to both classify, and segment PV and ST systems from Dutch aerial imagery. The classification task can be performed with a very high accuracy, while segmentation is still a more challenging task, although feasible. With a manually labelled set of 3,000 images and an unlabelled set of approximately 13,000 images, a classification-pretrained model can be trained to segment PV and ST systems as a binary class with an IoU of approximately 73%, and an approximately 77% and 38% IoU for the PV and ST classes respectively. This task could even be performed in a weakly supervised manner utilising only image-level labels, although this comes at a performance hit of approximately 10% IoU, which might decrease confidence in the utility of the resulting model.

8.3 Limitations and further research

8.3.1 Training duration

Due to budget and time constraints, network training of multiple components and their variations was limited to a set amount of epochs that was below what could be expected to yield optimal results. In the case of pretraining for example, the MAE trained on aerial images was only trained for 50 epochs, compared to the 800 epoch for which the authors of the ConvNeXt V2 paper pretrained their MAE [25]. Similarly, training of the segmentation networks with CorrMatch was limited to 21 up to 33 epochs, which is much lower than the work of Yang et al. [75] who trained their networks for 1,000 epochs with UniMatch on a similarly sized dataset. In further research, the first adjustment made to hyperparameters should therefore clearly be an increase in training duration, as it is expected that almost all networks presented in this paper could be trained for longer than was done to improve results.

8.3.2 SAM segmentation quality

During the process of generating pseudo-labels, it was noted that a limiting factor to the quality of the produced pseudo-labels using SAM was the recall of solar panels segmented from the background in the set of SAM output masks. In many cases, SAM was unable to recognise the solar panel arrays as separate objects from the background, especially in difficult cases such as for dark frameless panels on dark rooftops. In large arrays, SAM also struggled with recalling every individual panel, often leaving out gaps in the array of panels. Figure 8.3 illustrates these issues. This forms a problem for the pseudo-label generation as panels that are not present on any of the masks produced by SAM can impossibly be selected by intersection with the corresponding CAM, decreasing recall in the pseudo-labels.

Attempts to solve this issue in further research might focus on investigating options to improve the image qualities that allow SAM to make proper segmentations, such as sharpness or contrast. Another potential solution would be to fine-tune SAM on the target domain by manually segmenting a small portion of aerial images and retraining SAM on this task, which was out of scope for this project. Finally, SAM also allows for prompted segmentation, by inputting bounding boxes or points. Extracting these prompts from the refined CAMs and building pseudo

masks via prompted segmentation by SAM might also be a feasible approach to weak pseudo-label generation.

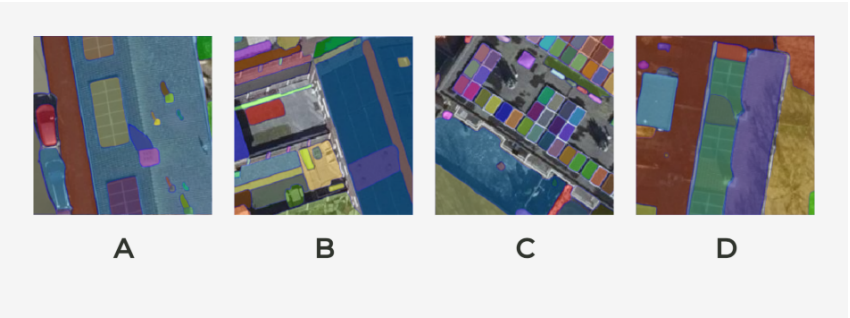

Figure 8.3: Examples of SAM segmentations. For solar panels with a contrasting colour to the background rooftop (A) segmentation works well. However, SAM produces unsatisfying results for dark panels on dark rooftops (B), large arrays of panels (C), and images with environmental disturbances such as shadows (D).

8.3.3 CAM refinement variations

While a handful of variations in the greedy block used for CAM refinement were tested, only two variations were used in the generation of pseudo-labels for weakly-supervised segmentation due to time constraints. The full effect of the proposed variations could be investigated further in future works. Additionally, other variations might be devised and tested. Examples might include using different block types inspired by other architectures for the greedy layer, or the introduction of a skip layer from the earlier stage of the network to the greedy block.

8.3.4 ST recall

While most models were able to classify and segment panels in the PV class to a high degree, ST panels proved to be much harder to detect. A main cause of this is likely to be the low representation of ST panels in both the nation covering aerial imagery, as well as the more balanced dataset created for this study. Future work might focus on building a dataset in which the distribution of PV and ST panels is even more balanced, or on the implementation of a loss function that mitigates the existing class imbalance.

8.4 Conclusion

This thesis project aimed to investigate the extent to which photovoltaic and solar thermal panels could be segmented from Dutch aerial imagery. To this end,

a pipeline was designed to first classify images, and then segment the positive samples. To train these networks, self-supervised pretraining with Masked Auto-Encoders was for the first time employed on Dutch aerial images, after which the classifier was finetuned on a novel manually annotated Dataset. Finally, a segmentation model was trained in a fully-, semi-, and weakly-supervised manner to segment the panels. For the weakly-supervised variant, a novel pipeline was proposed that utilised CAMs refined by greedy retraining to select masks from SAM segmentations. Each of these steps was performed on both a binary task, as well as a multi-label task.

Pretraining on Dutch aerial images with a Masked Auto-Encoder was found not to directly improve classification results compared to ImageNet pretraining with the amount of time for which was trained. However, with a fraction of the training time of the ImageNet pretrained model, the aerial image pretrained model was able to be finetuned to almost identical performance on the classification task. Additionally, later inspection showed the model had learned more relevant features in the target domain, which were more useful for the proposed pseudo-label generation pipeline. Multi-label classification was found to be more challenging than binary classification, with the ST class being especially difficult to detect. PV panels on the other hand were able to be classified with high accuracy.

Finally, segmentation in the fully-supervised case showed promising results, with an IoU of 70.3% for the binary task, but it was found that performance could be significantly improved by utilising an extra set of unlabelled images in a semi-supervised manner. The weakly-supervised network was trained on pseudo-labels generated with the proposed pipeline, and showed promising performance, even though it was naturally significantly lower than the models trained on manually annotated masks. The introduction of semi-supervised learning to the weakly-supervised model highlighted the benefit of scoring pseudo-labels based on confidence and only training on high-confidence labels. Results of this thesis can be used as a stepping stone for further research into both segmentation of PV and ST panels from aerial imagery, as well as the utilisation of CAM refinement in combination with SAM for weakly-supervised segmentation.

Bibliography

[1] U. Secretariat, *Technical dialogue of the first global stocktake. synthesis report by the co-facilitators on the technical dialogue*. [Online]. Available: https://unfccc.int/documents/631600.

[2] S. Netherlands, *Renewable energy share rose to 15 percent in 2022*, Jun. 2023. [Online]. Available: https://www.cbs.nl/en-gb/news/2023/22/renewable-energy-share-rose-to-15-percent-in-2022.

[3] Aug. 2018. [Online]. Available: https://www.stedin.net/over-stedin/pers-en-media/persberichten/kwart-van-de-zonnepanelen-niet-in-beeld.

[4] [Online]. Available: http://solarheateurope.eu/wp-content/uploads/2022/12/Solar_Heat_Market_Report-2021.pdf.

[5] B. B. Kausika, D. Nijmeijer, I. Reimerink, P. Brouwer, and V. Liem, "Geoai for detection of solar photovoltaic installations in the netherlands," *Energy and AI*, vol. 6, p. 100 111, 2021, ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2021.100111. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546821000604.

[6] [Online]. Available: https://we-boost.nl/.

[7] A. Kirillov, E. Mintun, N. Ravi, *et al.*, *Segment anything*, 2023. arXiv: 2304.02643 [cs.CV].

[8] H. Mao, X. Chen, Y. Luo, *et al.*, "Advances and prospects on estimating solar photovoltaic installation capacity and potential based on satellite and aerial images," *Renewable and Sustainable Energy Reviews*, vol. 179, p. 113 276, 2023, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2023.113276. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032123001326.

[9] C. Feng, Y. Liu, and J. Zhang, "A taxonomical review on recent artificial intelligence applications to pv integration into power grids," *International Journal of Electrical Power & Energy Systems*, vol. 132, p. 107 176, 2021, ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2021.107176. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061521004154.

[10] J. de Hoog, S. Maetschke, P. Ilfrich, and R. R. Kolluri, "Using satellite and aerial imagery for identification of solar pv: State of the art and research opportunities," in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, ser. e-Energy '20, Virtual Event, Australia: Association for Computing Machinery, 2020, pp. 308–313, ISBN: 9781450380096. DOI: 10.1145/3396851.3397681. [Online]. Available: https://doi.org/10.1145/3396851.3397681.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[12] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3,

pp. 211–252, Dec. 2015, ISSN: 1573-1405. DOI: 10.1007/s11263-015-0816-y. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[14] C. Szegedy, W. Liu, Y. Jia, *et al.*, *Going deeper with convolutions*, 2014. arXiv: 1409.4842 [cs.CV].

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, 2015. arXiv: 1512.00567 [cs.CV].

[16] [Online]. Available: https://paperswithcode.com/paper/rethinking-the-inception-architecture-for.

[17] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature pyramid networks for object detection*, 2017. arXiv: 1612.03144 [cs.CV].

[19] [Online]. Available: https://cocodataset.org/#home.

[20] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2020. arXiv: 2010.11929 [cs.CV].

[22] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, *Masked autoencoders are scalable vision learners*, 2021. arXiv: 2111.06377 [cs.CV].

[23] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, *A convnet for the 2020s*, 2022. arXiv: 2201.03545 [cs.CV].

[24] Z. Liu, Y. Lin, Y. Cao, *et al.*, *Swin transformer: Hierarchical vision transformer using shifted windows*, 2021. arXiv: 2103.14030 [cs.CV].

[25] S. Woo, S. Debnath, R. Hu, *et al.*, *Convnext v2: Co-designing and scaling convnets with masked autoencoders*, 2023. arXiv: 2301.00808 [cs.CV].

[26] J. Malof, R. Hou, L. Collins, K. Bradbury, and R. Newell, "Automatic solar photovoltaic panel detection in satellite imagery," Nov. 2015, pp. 1428–1431. DOI: 10.1109/ICRERA.2015.7418643.

[27] M. Vasku, *An exploration of automatic detection of large-scale solar plants: Application of machine learning-based image classification in google earth engine*, Jun. 2019. [Online]. Available: https://projekter.aau.dk/projekter/en/studentthesis/an-exploration-of-automatic-detection-of-largescale-solar-plants-application-of-machine-learningbased-image-classification-in-google-earth-engine(f978aace-a4ef-4f4e-8ca0-879f13cd9486).html.

[28] J. M. Malof, K. Bradbury, L. M. Collins, and R. G. Newell, "Automatic detection of solar photovoltaic arrays in high resolution aerial imagery," *Applied Energy*, vol. 183, pp. 229–240, 2016, ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2016.08.191. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261916313009.

[29] K. Bradbury, R. Saboo, J. Malof, *et al.*, "Distributed Solar Photovoltaic Array Location and Extent Data Set for Remote Sensing Object Identification," May 2016. DOI: 10.6084/m9.figshare.3385780.v1. [Online]. Avail-

able: https://figshare.com/articles/dataset/Distributed_Solar_Photovoltaic_Array_Location_and_Extent_Data_Set_for_Remote_Sensing_Object_Identification/3385780.

[30] J. M. Malof, L. M. Collins, K. Bradbury, and R. G. Newell, "A deep convolutional neural network and a random forest classifier for solar photovoltaic array detection in aerial imagery," in *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, 2016, pp. 650–654. DOI: 10.1109/ICRERA.2016.7884415.

[31] Z. Xia, Y. Li, X. Guo, and R. Chen, "High-resolution mapping of water photovoltaic development in china through satellite imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 107, p. 102 707, 2022, ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2022.102707. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243422000332.

[32] X. Zhang, M. Zeraatpisheh, M. M. Rahman, S. Wang, and M. Xu, "Texture is important in improving the accuracy of mapping photovoltaic power plants: A case study of ningxia autonomous region, china," *Remote Sensing*, vol. 13, no. 19, 2021, ISSN: 2072-4292. DOI: 10.3390/rs13193909. [Online]. Available: https://www.mdpi.com/2072-4292/13/19/3909.

[33] J. R. Veerle Plakman and J. van Vliet, "Solar park detection from publicly available satellite imagery," *GIScience & Remote Sensing*, vol. 59, no. 1, pp. 462–481, 2022. DOI: 10.1080/15481603.2022.2036056. eprint: https://doi.org/10.1080/15481603.2022.2036056. [Online]. Available: https://doi.org/10.1080/15481603.2022.2036056.

[34] 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," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 874–877. DOI: 10.1109/IGARSS.2017.8127092.

[35] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: 1409.1556 [cs.CV].

[36] V. Golovko, S. Bezobrazov, A. Kroshchanka, A. Sachenko, M. Komar, and A. Karachka, "Convolutional neural network based solar photovoltaic panel detection in satellite photos," in *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, 2017, pp. 14–19. DOI: 10.1109/IDAACS.2017.8094501.

[37] 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*, vol. 2, no. 12, pp. 2605–2617, 2018, ISSN: 2542-4351. DOI: https://doi.org/10.1016/j.joule.2018.11.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542435118305701.

[38] K. Ioannou and D. Myronidis, "Automatic detection of photovoltaic farms using satellite imagery and convolutional neural networks," *Sustainability*, vol. 13, no. 9, 2021, ISSN: 2071-1050. DOI: 10.3390/su13095323. [Online]. Available: https://www.mdpi.com/2071-1050/13/9/5323.

[39] Z. Wang, M.-L. Arlt, C. Zanocco, A. Majumdar, and R. Rajagopal, "Deepsolar++: Understanding residential solar adoption trajectories with computer vision and technology diffusion models," *Joule*, vol. 6, no. 11, pp. 2611–

2625, 2022, ISSN: 2542-4351. DOI: https://doi.org/10.1016/j.joule.2022.09.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542435122004779.

[40] K. Mayer, Z. Wang, M.-L. Arlt, D. Neumann, and R. Rajagopal, "Deepsolar for germany: A deep learning framework for pv system mapping from aerial imagery," in *2020 International Conference on Smart Energy Systems and Technologies (SEST)*, 2020, pp. 1–6. DOI: 10.1109/SEST48500.2020.9203258.

[41] K. Mayer, B. Rausch, M.-L. Arlt, *et al.*, "3d-pv-locator: Large-scale detection of rooftop-mounted photovoltaic systems in 3d," *Applied Energy*, vol. 310, p. 118 469, 2022, ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2021.118469. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921016937.

[42] J. Lindahl, R. Johansson, and D. Lingfors, "Mapping of decentralised photovoltaic and solar thermal systems by remote sensing aerial imagery and deep machine learning for statistic generation," *Energy and AI*, vol. 14, p. 100 300, 2023, ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2023.100300. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546823000721.

[43] S. Netherlands, *Automatically detect solar panels with aerial photos*, Feb. 2023. [Online]. Available: https://www.cbs.nl/en-gb/about-us/innovation/project/automatically-detect-solar-panels-with-aerial-photos.

[44] H. van Leeuwen, *Detecting solar panels on aerial images of limburg with convolutional neural networks*, Feb. 2023. [Online]. Available: https://www.cbs.nl/en-gb/about-us/innovation/project/follow-up-study-on-detection-of-solar-panels-from-earth-observation.

[45] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-cnn*, 2018. arXiv: 1703.06870 [cs.CV].

[46] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, 2014. arXiv: 1311.2524 [cs.CV].

[47] R. Girshick, *Fast r-cnn*, 2015. arXiv: 1504.08083 [cs.CV].

[48] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2016. arXiv: 1506.01497 [cs.CV].

[49] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015. arXiv: 1505.04597 [cs.CV].

[50] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, *Expectation-maximization attention networks for semantic segmentation*, 2019. arXiv: 1907.13426 [cs.CV].

[51] 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*, 2017. arXiv: 1606.00915 [cs.CV].

[52] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, *Rethinking atrous convolution for semantic image segmentation*, 2017. arXiv: 1706.05587 [cs.CV].

[53] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, *Encoder-decoder with atrous separable convolution for semantic image segmentation*, 2018. arXiv: 1802.02611 [cs.CV].

[54] J. Xie, J. Xiang, J. Chen, X. Hou, X. Zhao, and L. Shen, *Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation*, 2022. arXiv: 2203.13505 [cs.CV].

[55] J. Ahn, S. Cho, and S. Kwak, *Weakly supervised learning of instance segmentation with inter-pixel relations*, 2019. arXiv: 1904.05044 [cs.CV].

[56] J. Lee, E. Kim, and S. Yoon, *Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation*, 2021. arXiv: 2103.08896 [cs.CV].

[57] R. Li, Z. Mai, Z. Zhang, J. Jang, and S. Sanner, "Transcam: Transformer attention-based cam refinement for weakly supervised semantic segmentation," *Journal of Visual Communication and Image Representation*, vol. 92, p. 103 800, Apr. 2023, ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2023.103800. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2023.103800.

[58] P.-T. Jiang and Y. Yang, *Segment anything is a good pseudo-label generator for weakly supervised semantic segmentation*, 2023. arXiv: 2305.01275 [cs.CV].

[59] T. Chen, Z. Mai, R. Li, and W.-l. Chao, *Segment anything model (sam) enhanced pseudo labels for weakly supervised semantic segmentation*, 2023. arXiv: 2305.05803 [cs.CV].

[60] K. Sohn, D. Berthelot, C.-L. Li, *et al.*, *Fixmatch: Simplifying semi-supervised learning with consistency and confidence*, 2020. arXiv: 2001.07685 [cs.LG].

[61] L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi, *Revisiting weak-to-strong consistency in semi-supervised semantic segmentation*, 2023. arXiv: 2208.09910 [cs.CV].

[62] B. Sun, Y. Yang, L. Zhang, M.-M. Cheng, and Q. Hou, *Corrmatch: Label propagation via correlation matching for semi-supervised semantic segmentation*, 2023. arXiv: 2306.04300 [cs.CV].

[63] H. K. Cheng, J. Chung, Y.-W. Tai, and C.-K. Tang, *Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement*, 2020. arXiv: 2005.02551 [cs.CV].

[64] J. Yuan, H.-H. L. Yang, O. A. Omitaomu, and B. L. Bhaduri, "Large-scale solar panel mapping from aerial images using deep convolutional networks," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 2703–2708. DOI: 10.1109/BigData.2016.7840915.

[65] J. Camilo, R. Wang, L. M. Collins, K. Bradbury, and J. M. Malof, *Application of a semantic segmentation convolutional neural network for accurate automatic detection and mapping of solar photovoltaic arrays in aerial imagery*, 2018. arXiv: 1801.04018 [cs.CV].

[66] J. Malof, B. Li, B. Huang, K. Bradbury, and A. Stretslov, "Mapping solar array location, size, and capacity using deep learning and overhead imagery," Feb. 2019.

[67] A. M. Moradi Sizkouhi, M. Aghaei, S. M. Esmailifar, M. R. Mohammadi, and F. Grimaccia, "Automatic boundary extraction of large-scale photovoltaic plants using a fully convolutional network on aerial imagery," *IEEE Journal of Photovoltaics*, vol. 10, no. 4, pp. 1061–1067, 2020. DOI: 10.1109/JPHOTOV.2020.2992339.

[68] Q. Li, S. Schott, and D. Chen, "Solardetector: Automatic solar pv array identification using big satellite imagery data," in *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*, ser. IoTDI '23, San Antonio, TX, USA: Association for Computing Machinery, 2023, pp. 117–129. DOI: 10.1145/3576842.3582384. [Online]. Available: https://doi.org/10.1145/3576842.3582384.

[69] S. Liang, F. Qi, Y. Ding, R. Cao, Q. Yang, and W. Yan, "Mask r-cnn based segmentation method for satellite imagery of photovoltaics generation systems," in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 5343–5348. DOI: 10.23919/CCC50068.2020.9189474.

[70] M. Schulz, B. Boughattas, and F. Wendel, "Deteektor: Mask r-cnn based neural network for energy plant identification on aerial photographs," *Energy and AI*, vol. 5, p. 100 069, 2021, ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2021.100069. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666546821000239.

[71] M. Zech and J. Ranalli, "Predicting pv areas in aerial images with deep learning," in *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*, 2020, pp. 0767–0774. DOI: 10.1109/PVSC45281.2020.9300636.

[72] P. Parhar, R. Sawasaki, A. Todeschini, *et al.*, "Hyperionsolarnet: Solar panel detection from aerial images," *CoRR*, vol. abs/2201.02107, 2022. arXiv: 2201.02107. [Online]. Available: https://arxiv.org/abs/2201.02107.

[73] L. Kruitwagen, K. Story, J. Friedrich, L. Byers, S. Skillman, and C. Hepburn, "A global inventory of photovoltaic solar energy generating units," *Nature*, vol. 598, pp. 604–610, Oct. 2021. DOI: 10.1038/s41586-021-03957-7.

[74] L. Zhuang, Z. Zhang, and L. Wang, "The automatic segmentation of residential solar panels based on satellite images: A cross learning driven u-net method," *Applied Soft Computing*, vol. 92, p. 106 283, 2020, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2020.106283. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620302234.

[75] R. Yang, G. He, R. Yin, *et al.*, "Weakly-semi supervised extraction of rooftop photovoltaics from high-resolution images based on segment anything model and class activation map," *Applied Energy*, vol. 361, p. 122 964, 2024, ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2024.122964. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261924003477.

[76] R. Wang, J. Camilo, L. M. Collins, K. Bradbury, and J. M. Malof, "The poor generalization of deep convolutional networks to aerial imagery from new geographic locations: An empirical study with solar array detection," in *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2017, pp. 1–8. DOI: 10.1109/AIPR.2017.8457965.

[77] 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*, 2019. arXiv: 1912.03685 [cs.CV].

[78] A. Greco, C. Pironti, A. Saggese, M. Vento, and V. Vigilante, "A deep learning based approach for detecting panels in photovoltaic plants," in *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, ser. APPIS 2020, Las Palmas de Gran Canaria, Spain: Association for Computing Machinery, 2020, ISBN: 9781450376303. DOI: 10.1145/3378184.3378185. [Online]. Available: https://doi.org/10.1145/3378184.3378185.

[79] M. S. Karoui, F. Z. Benhalouche, Y. Deville, *et al.*, "Partial linear nmf-based unmixing methods for detection and area estimation of photovoltaic panels in urban hyperspectral remote sensing data," *Remote Sensing*, vol. 11,

no. 18, 2019, ISSN: 2072-4292. DOI: 10.3390/rs11182164. [Online]. Available: https://www.mdpi.com/2072-4292/11/18/2164.

[80] D. Stowell, J. Kelly, D. Tanner, *et al.*, "A harmonised, high-coverage, open dataset of solar photovoltaic installations in the uk," *Scientific Data*, vol. 7, no. 1, p. 394, Nov. 2020, ISSN: 2052-4463. DOI: 10.1038/s41597-020-00739-0. [Online]. Available: https://doi.org/10.1038/s41597-020-00739-0.

[81] H. Jiang, L. Yao, N. Lu, *et al.*, "Multi-resolution dataset for photovoltaic panel segmentation from satellite and aerial imagery," *Earth System Science Data*, vol. 13, no. 11, pp. 5389–5401, 2021. DOI: 10.5194/essd-13-5389-2021. [Online]. Available: https://essd.copernicus.org/articles/13/5389/2021/.

[82] [Online]. Available: https://www.pdok.nl/.

[83] [Online]. Available: https://www.pdok.nl/introductie/-/article/pdok-luchtfoto-rgb-open-.

[84] [Online]. Available: https://www.pdok.nl/introductie/-/article/basisregistratie-adressen-en-gebouwen-ba-1.

[85] [Online]. Available: https://pypi.org/project/segmentation-refinement/.

[86] [Online]. Available: https://www.cvat.ai/.

[87] S. Chen, Y. Ogawa, and Y. Sekimoto, "Large-scale individual building extraction from open-source satellite imagery via super-resolution-based instance segmentation approach," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 129–152, Jan. 2023. DOI: 10.1016/j.isprsjprs.2022.11.006.

[88] Facebookresearch, *Facebookresearch/segment-anything: The repository provides code for running inference with the segmentanything model (sam), links for downloading the trained model checkpoints, and example notebooks that show how to use the model.* [Online]. Available: https://github.com/facebookresearch/segment-anything.