Milan Kacar, BSc

Robust Detection of Solar Panels Using Areal and Satellite Imagery

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Geodesy

submitted to

Graz University of Technology

Supervisor

Saukh, Olga, bak. Assoc.Prof. Dr.rer.nat. MSc

Institute of Technical Informatics

Graz, October 2024

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date, Signature

Abstract

Accurate detection and analysis of rooftop solar Photovoltaic (PV) installations are crucial for supporting the transition to sustainable energy. As the number of solar panel installations grows, understanding their distribution is essential for grid management, energy production forecasting, and effective policymaking. However, tracking solar panels remains challenging due to their decentralized deployment across various locations and the diversity of image data sources. This thesis aims to develop a robust, scalable methodology for detecting solar panels using both aerial and satellite imagery, leveraging advanced computer vision techniques. This study employs models such as You Only Look Once version 5 (YOLOv5) to detect solar arrays from high-resolution aerial and satellite images. The use of diverse datasets from sources including the Full Collection: Distributed Solar Photovoltaic Array Location and Extent Data Set for Remote Sensing Object Identification (FCDPA), Deep Solar, Solar Finder, and Vienna GIS data provides a broad foundation for developing a detection method capable of handling varied image resolutions and geographic locations. Data augmentation and adversarial training are implemented to enhance model robustness against natural distribution shifts and environmental variations, ensuring high accuracy under different conditions like seasonal changes and lighting variations. An important contribution of this work is the use of model soups, where multiple trained models are combined to improve accuracy and robustness. This approach achieves an accuracy of 95% across different image types, presenting a cost-effective and scalable solution for estimating solar panel installations on a global scale. Additionally, this research uses affinity and diversity metrics to assess the models' resilience, providing critical insights into their performance under various data conditions. Furthermore, the study also provides an analysis of solar panel distribution and growth in Vienna, Austria, from 2015 to 2020. However, it is important to note a disclaimer regarding these findings: due to the lack of ground truth data for direct validation, the presented estimates should be interpreted with caution. The results serve as preliminary insights rather than definitive metrics, highlighting the need for further data collection and validation efforts. Overall, this thesis offers a comprehensive methodology for detecting and monitoring solar PV installations, significantly contributing to the field of renewable energy. Future research directions could explore the integration of additional data sources, such as thermal imagery, and the development of more advanced machine learning models to further enhance detection capabilities under diverse environmental conditions.

Kurzfassung

Eine genaue Erkennung und Analyse von Solaranlagen auf Dächern ist entscheidend für die Unterstützung des Übergangs zu nachhaltiger Energie. Da die Anzahl der installierten Solarmodule wächst, ist das Verständnis ihrer Verteilung für das Netzmanagement, die Vorhersage der Energieproduktion und eine effektive Politikgestaltung unerlässlich. Allerdings bleibt das Tracking von Solarmodulen aufgrund ihrer dezentralen Installation an verschiedenen Standorten und der Vielfalt der Bilddatenquellen eine Herausforderung. Diese Arbeit zielt darauf ab, eine robuste, skalierbare Methodik zur Erkennung von Solarmodulen mithilfe von Luft- und Satellitenbildern zu entwickeln und dabei fortschrittliche Techniken der Computer Vision zu nutzen. Diese Studie verwendet Modelle wie You Only Look Once version 5 (YOLOv5), um Solararrays in hochauflösenden Luft- und Satellitenbildern zu erkennen. Die Verwendung verschiedener Datensätze aus Quellen wie dem Full Collection: Distributed Solar Photovoltaic Array Location and Extent Data Set for Remote Sensing Object Identification (FCDPA), Deep Solar, Solar Finder und den GIS-Daten von Wien bietet eine breite Grundlage für die Entwicklung einer Erkennungsmethode, die mit unterschiedlichen Bildauflösungen und geografischen Standorten umgehen kann. Um die Robustheit der Modelle gegenüber natürlichen Verteilungsschwankungen und Umweltveränderungen zu erhöhen, werden Datenaugmentation und adversariales Training implementiert, um eine hohe Genauigkeit unter verschiedenen Bedingungen wie saisonalen Veränderungen und Lichtverhältnissen zu gewährleisten. Ein wichtiger Beitrag dieser Arbeit ist die Verwendung von Model Soups, bei denen mehrere trainierte Modelle kombiniert werden, um Genauigkeit und Robustheit zu verbessern. Dieser Ansatz erreicht eine Genauigkeit von 95% über verschiedene Bildtypen hinweg und bietet eine kosteneffiziente und skalierbare Lösung für die Schätzung von Solar-Photovoltaic-Installationen weltweit. Darüber hinaus führt diese Forschung Affinitäts- und Diversitätsmetriken ein, um die Widerstandsfähigkeit der Modelle zu bewerten und wichtige Einblicke in ihre Leistung unter verschiedenen Datenbedingungen zu geben. Die Studie bietet auch eine Analyse der Verteilung und des Wachstums von Solar-Photovoltaic-Anlagen in Wien, Österreich, von 2015 bis 2020. Es ist jedoch wichtig, einen Disclaimer für diese Ergebnisse zu beachten: Aufgrund des Fehlens von Ground-Truth-Daten für eine direkte Validierung sollten die präsentierten Schätzungen mit Vorsicht interpretiert werden. Die Ergebnisse dienen als vorläufige Einblicke und nicht als definitive Metriken und unterstreichen die Notwendigkeit weiterer Datenerhebungs- und Validierungsbemühungen. Insgesamt bietet diese Arbeit eine umfassende Methodik zur Erkennung und Überwachung von Solar-Photovoltaic-Installationen und leistet einen bedeutenden Beitrag zum Bereich der erneuerbaren Energien. Zukünftige Forschungsrichtungen könnten die Integration zusätzlicher Datenquellen, wie z. B. thermischer Bilder, und die Entwick-

lung fortschrittlicherer Modelle des maschinellen Lernens zur weiteren Verbesserung der Erkennungsfähigkeiten unter verschiedenen Umweltbedingungen erkunden.

Acknowledgements

First and foremost, I want to sincerely thank the Republic of Austria, which enabled me to study and complete my studies in this beautiful country. Being placed outside one's comfort zone in a new academic environment, with diverse perspectives, personally and professionally growing, has been an opportunity that comes truly as a great matter of privilege. This opportunity has further enriched not only my horizons but also the understanding of the world as a person from Bosnia. I am deeply in debt to Austria for having made this journey possible and enabling me to realize my educational goals.

I would like to take this opportunity and express my deep gratitude to my parents, too, for being the support in this long journey. My family's belief in me and my potential to follow my dreams, even when that meant leaving Bosnia, has been one of the major factors in my success. Of course, it is not easy to decide to move to another country, but the fact that I had their complete support gave me strength and confidence to embark on a new journey in my life. I am thankful beyond words for their love, patience, and sacrifices, due to which, I have got the opportunity to chase my dreams.

Special thanks are also due to Prof. Dr. Saukh, whose constant support and patience extended throughout this process were very valuable. Her readiness to reflect directions, constructive feedback, and encouragement made all the difference during the most testing moments of my research. Most importantly, I am greatly indebted to her for her understanding and flexibility in allowing me to maneuver through such obstacles that were sure to come up in the course of an effort such as this. Dr. Saukh has been much more than an academic advisor to me; she was a mentor who cared sincerely about my progress and well-being, making sure that I was focused and motivated throughout.

Above all, I want to extend my deepest gratitude to Prof. Dr. Saukh. She cannot be overrated in my academic trajectory. Prof. Dr. Saukh has been an exceptional source of guidance, wisdom, and support. She indeed went out of her way to make sure that I had all the resources, feedback, and encouragement I needed toward success. Her ability to provide clarity and direction into the most complex phases of my research was nothing less than remarkable. Her support did not stop at times related to regular working hours, but she would be there during holidays and other times when most of us would not expect a person to answer. Her dedication to my success was really quite beyond the ordinary, with her interventions often making that critical difference between lost and found. I feel really grateful for her kindness, insights, and unwavering belief in my work. I am definitely going to exaggerate by saying that had it not been for the incessant efforts and dedication of Prof. Dr. Saukh, I would have faced many more

hurdles on my way.

I am also deeply indebted to all those people who helped me reach this point. From the good environment provided in Austria to the support of my parents, and ending with the outstanding support of Prof. Dr. Saukh, this success is as much theirs as mine. I want to thank all of you for being a part of this journey and for your assistance in reaching this milestone.

Contents

1 Introduction

In this section, we discuss the motivation and the main contributions of this thesis followed by a detailed road map.

1.1 Background and Motivation

In recent years, the global transition towards solar power for residential and commercial use has gained significant momentum. This shift is driven by the urgent need to address the environmental impact of fossil fuels, which continue to dominate the world's energy production. Fossil fuels are a major contributor to climate change, as can be seen in Figure 1.1. Data from the Gravity Recovery and Climate Experiment (GRACE) satellite mission between 2004 and 2010 shows marked variations in polar ice mass, particularly during the winter months in the northern hemisphere.

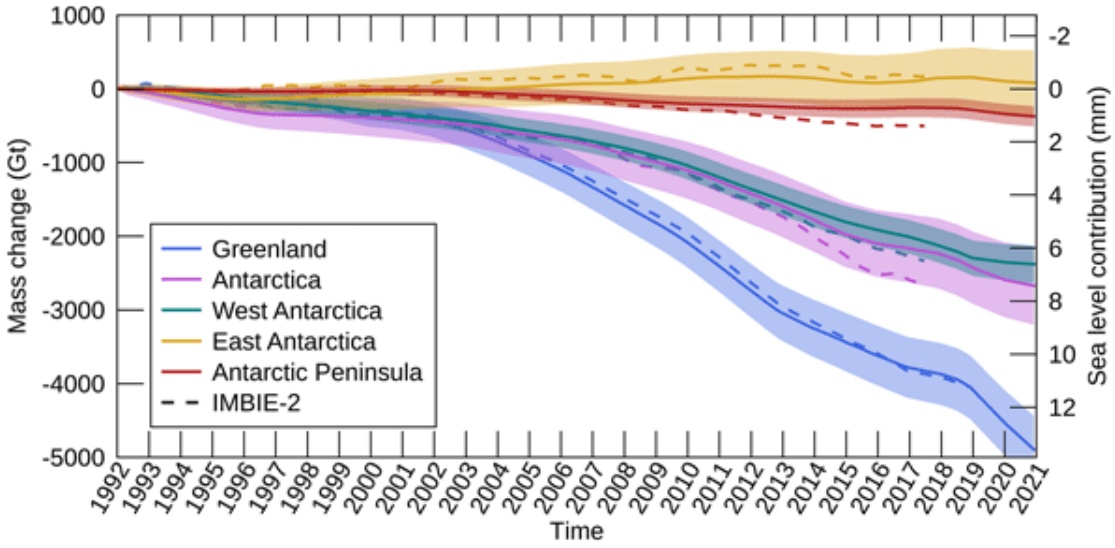

Figure 1.1: Polar ice mass variation over the years. Source: [82]

An extended analysis from 1990 to 2020 offers a significant negative trend in ice mass balance across Greenland and the Antarctic regions, with a particularly noticeable loss in the Greenland ice sheet. While East Antarctica and the Antarctic Peninsula have experienced slight gains in mass over the past three decades, East Antarctica's mass balance has remained relatively constant with minor fluctuations. Average trends from the Ice sheet Mass Balance Inter-comparison Exercise (IMBIE-2) data confirm a

widespread reduction in polar ice, which is linked to rising sea levels. The exponential growth of sea levels, particularly in the latter half of the study period, mirrors the compound effects of ice melt on ocean volumes.

These findings highlight the urgent need to mitigate climate change by reducing our reliance and strong dependency on fossil fuels. One possible solution would be the adoption of renewable energy sources, particularly solar energy. Solar power harnesses sunlight to generate electricity through PV cells, helping us reduce the need for fossil fuels. The deployment of solar panels significantly decreases the emission of harmful air pollutants such as sulfur dioxide, nitrogen oxides, and particulate matter, which are prevalent in coal and natural gas plants. This transition not only enhances air quality but also conserves water resources, as solar power generation requires minimal water compared to the cooling processes of conventional thermal power plants. The widespread adoption of solar panels alongside wind turbines represents a crucial step towards releasing and hindering the environmental pressures on air and water resources while promoting sustainable energy practices.

The urgency of transitioning to sustainable energy is additionally underlined by Earth Overshoot Day. In 2023, humanity reached Earth Overshoot Day on August 2, the date when we exhausted the planet's capacity to regenerate its renewable resources for the year. This implies that we are consuming resources at a rate that would require 1.7 Earths to generate annually [77]. This alarming rate of resource consumption underscores the critical need for spreading the adoption of renewable energy sources to reduce the environmental impact and promote sustainability.

Solar energy has became more affordable and the technology behind it improved dramatically in recent years. Swanson's Law, which describes the economic trend of photovoltaic modules, indicates that the price of solar modules tends to drop by 20% for every doubling of cumulative shipped volume. This trend is illustrated by the substantial decrease in prices from $76.67 per watt in 1977 to $0.36 per watt in 2014, making solar energy increasingly accessible and affordable.

However, a significant challenge remains in accurately tracking and analyzing the adoption of solar panels. Traditional methods, such as public surveys, are often costly, time-consuming, and inefficient. Satellite and aerial imagery present a more effective and efficient alternative for data collection. Despite this, detailed statistics on solar panel adoption, efficiency, and output are often aggregated with other renewable energy sources, lacking the information and granularity needed for specific analysis.

The primary motivation for this research is to develop robust methods for detecting solar panels using satellite and aerial imagery. Accurate detection is essential for understanding the distribution, capacity, and other details about solar panels, which is crucial for informing policy-making and supporting the broader adoption of renewable energy. By the use of advanced computer vision algorithms, this research aims to identify solar arrays from diverse image sources with varying resolutions from different areas and parts of the world. This approach not only facilitates the estimation of the number of solar panels but also provides a scalable and cost-effective solution for finding the exact positions of solar panel installations globally.

Furthermore, this research seeks to enhance the robustness of detection algorithms un-

der various environmental conditions and architectural variations. Given the diversity of environmental factors, such as seasonal changes and weather conditions, the algorithms must be resilient to natural distribution shifts. The research explores the use of model soups, where the weights of multiple models are combined in different ways to improve accuracy and robustness. This approach aims to increase the reliability of solar panel detection, even under adverse conditions.

Ultimately, this work aims to provide a comprehensive and scalable method for obtaining reliable estimates of solar panel installations, contributing to the transition towards sustainable energy sources and mitigating the adverse effects of climate change. By advancing the accuracy and robustness of solar panel detection, this research supports the broader goals of environmental sustainability and energy efficiency.

1.2 Research Questions and Objectives

With the continuing expansion of solar Photovoltaic PV arrays all over the world, the problem of forecasting the loads on the electricity grid becomes difficult. The traditional predictive models often struggle to generalize across different regions, especially in areas where labeled data is unavailable and where environmental and architectural factors differ significantly. The central issue this investigation is trying to solve is how to develop a reliable model that will work anywhere, including regions with varying landscapes, cities with unique architectural peculiarities, and during all seasonal changes.

The primary objective of this work is to create a predictive model that is not only accurate but also resilient to natural distribution shifts—these are changes in the data distribution that occur due to differences in environmental factors such as landscape variations, architectural styles, and seasonal effects. Current methodologies [122], [63] and [75] are usually based on data augmentation, which attempts to enhance the variety of the training set through synthetic means, such as rotation, scaling, and noise addition. Although data augmentation can make the model more robust to a certain degree, it is not enough to deal with the more complicated and natural distribution shift in the real world.

Moreover, the project aims to explore the potential of the "model soups" contribution to increase robustness and accuracy in solar panel prediction. According to [119], model soups exhibit a measurable impact on performance across different data distribution scenarios. The research highlights their efficacy in improving outcomes over a broad spectrum of datasets. Model soups have proven that they boost model performance substantially on heterogeneous datasets especially when traditional methods fail. Using both data augmentation and model soups the goal of this research is to create a model that can accurately count the number of existing solar panel installations in any part of the world with minimal labeled data and with large environmental variations. This study has demonstrated that model performance is significantly affected by various distribution shifts, such as changes in local architecture, seasonal variations, or demographic differences in the data. These shifts can introduce inconsistencies between training and test data, resulting in reduced model robustness. By addressing these distribution shifts,

models can better generalize across diverse datasets and perform consistently in real-world applications.

In essence, this project seeks to bridge the gap between the current capabilities of predictive models and the practical need for a solution that can adapt to a wide range of conditions. This work will help in making more precise and reliable solar energy forecasts, a crucial step in the world's transition to renewable energy sources, by concentrating on the model's robustness and generalizability.

1.3 Contributions

This work provides a comprehensive analysis of solar panel detection, focusing on estimating the number and growth of solar panels using a diverse dataset that includes images from various regions and environments. The employed model, based on model soups [119], facilitates the calculation of pixel-level coordinates and generates bounding boxes. These coordinates are then translated into geographical locations using a pre-trained YOLOv5 model. This approach allows for precise localization of solar panels with exact geographical precision.

This model provides a very high degree of accuracy, up to an overall detection accuracy of 91% within the mixed dataset of different landscapes, architectural styles, and seasonal conditions. Furthermore, the integration of the model soup technique proved to be quite effective. Model soups add another 4% accuracy beyond what was achievable using single models alone, enhancing robustness and effectiveness of the model in handling natural distribution shifts. This further underlines the potential of model soups in establishing the detection model as reliable and applicable over wide areas with diverse environmental and architectural conditions.

1.4 Road map

This thesis continues by presenting a comprehensive review of the existing literature and the broader context of the research in Section 2. Literature section provides an exploration and challenges of prior work on solar panel detection. It also identifies gaps in the current methodologies, setting the stage for the improvements of presented studies. Following the literature review, Section 3 illustrates the methodological framework of the research in this work. This section details the data sources, including the mixed dataset used, which incorporates images from a variety of geographical regions with differing architectural styles and environmental conditions. Besides that, some analytical techniques are also presented, like the implementation of the YOLOv5 model along with advanced techniques such as model soups. This therefore ensures that the model for detecting solar panels is robust and accurate, hence a good basis for the future coming analysis. Results of the experiments and their comparison against existing standards are presented in Section 4. The section describes how well the model performed in the mixed dataset, points out key findings and shows improvements achieved by this approach. Special consideration was given to the model's accuracy and robustness,

taking into account various conditions that may exist within the dataset. Another factor is how well the overall performance improved by enhancing the model with model soups technique. Finally, Section 5 interprets the findings of the research study and discusses their broader implications in the field of solar panel detection and renewable energy forecasting. The discussion provides strengths and limitations of this approach, keeping in mind how these could influence follow-up research and applications. An overview of potential further studies in this direction is presented, along with the implications of this work for improving both the accuracy and reliability of solar panel detection on a global scale in real-world scenarios.

2 Literature Review

The aim in this section is to analyze the literature and describe the main concepts used in this work. An overview of solar panel detection from satellite imagery is provided in the first part of the section. We also reviewed previous work on solar panel detection using deep learning and computer vision techniques, and the last section focuses on improving method robustness.

2.1 Overview of Solar Panel Detection from Satellite Images

Solar panel detection from satellite images has always been a challenging task, attracting considerable attention due to its importance for monitoring and understanding the renewable energy sector. This task is particularly complex due to factors such as varying geometries of solar panels, differences in lighting conditions, and the presence of shadows that can obscure the panels in images. These challenges have been addressed through a range of techniques related to computer vision, remote sensing, and machine learning.

One of the most effective strategies for solar panel detection involves the use of convolutional neural networks CNNs [81]. Deep learning models, such as CNNs, enable image recognition by learning to detect specific patterns from large amounts of labeled image data.The training datasets for CNNs in tasks related to solar panel detection include aerial and satellite images with manually labeled solar panels by human annotators [3]. These models perform really well because they can grasp the distinct visual appearance features of solar panels, such as shape, color, and reflectance, even under challenging conditions, like shadows or complex angles of sunlight [105]. However, the performance of CNNs can vary significantly depending on the quality and variability of the data. Their generalization capability might be poor if the models were trained on images taken from only one geographical region when applied to other regions with distinctive architectural features or environmental conditions. Besides CNN, other deep learning variants have also been attempted to improve the accuracy and robustness of solar panel detection. Advanced architectures, such as the U-Net [100], which have been designed for tasks related to image segmentation, proved to be quite effective in identifying and mapping solar panels over large areas. These models work in such a way that they segment the image into parts and classify each segment as either containing a solar panel or not. This is especially useful in complex environments where partial obscuration of solar panels could take place or when they are badly distinguishable from other objects. Beyond CNNs, traditional machine learning techniques like Random Forest [7], Support Vector Machines (SVM) [19], and Logistic Regression [18] have also been applied to this task. Random Forests are highly effective on high-dimensional data [7], with detection accuracy benefiting from the combination of multiple decision trees. This

approach minimizes overfitting and enhances the model's ability to generalize to unseen data. SVMs [89] are useful for distinguishing between image regions containing solar panels by finding a hyperplane that separates the classes in a high-dimensional feature space, which is especially effective when the data is not linearly separable. Logistic regression [51] provides a simpler model, and by doing some effective feature engineering, by selecting well-chosen panel characteristics, this model can work as a reliable baseline for the detection of solar panels in both satellite and aerial imagery.

In a nutshell, the research area of solar panel detection from satellite imagery has significantly benefited from the use of deep learning and computer vision. Employing large and diverse datasets and complex models, such as CNNs and U-Net, researchers have been successful in achieving ever-higher levels of accuracy and reliability in the detection of solar panels, which in turn helps toward better monitoring and assessment of solar energy installations worldwide.

2.2 Previous Work on Solar Panel Detection using Deep Learning and Computer Vision Techniques

The field of research on solar panel detection using deep learning and computer vision has considerably evolved since the automation and enhancement of the detection process have created increasing awareness across many applications [87]. This section therefore reviews previous works in the area with a focus on methodologies, key findings, and contributions that these works have made to the broader context of analysis in PV systems. This section offers a critical review of key works done in this area while highlighting their methodologies, key findings, and their contributions.

One notable study [112] of the applicability of machine learning techniques in fault detection for PV systems investigates different fault conditions at the PV module, component, and system level. These may include electrical faults as well as visible defects of solar panels. The paper emphasizes the applications of AI techniques, such as Fully Convolutional Networks (FCNs) [69], Convolutional Neural Networks (CNNs) [81], and Generative Adversarial Networks (GANs) [38] for intelligent fault diagnosis. Each one of these models will be fed by a large dataset of images and data from sensors to be obtained on PV systems, enabling the models to learn and understand the patterns associated with different kinds of faults. The findings support fault detection for improvement and reliability in PV systems with the understanding that early detection of faults may prevent huge energy losses and reduce maintenance costs.

Another useful contribution [61] is a study introducing an algorithm using deep learning for defect detection at solar panels. This approach reveals how defects in solar panels are visible and can be identified through image analysis. Defects such as hidden cracks, scratches, broken grids, black spots, and short circuits manifest visually on the surface of the panels, making them detectable through advanced computer vision techniques. This paper presents a revised version of the YOLOv5 algorithm [121], which has a Convolutional Block Attention Module (CBAM) [118] embedded to raise the detection of common defects like hidden cracks, scratches, broken grids, black spots on the pan-

els, and short circuits. The improved YOLOv5 algorithm [116] makes use of the CBAM which allows it to focus on the most relevant parts of the image, thereby improving detection accuracy. It was trained and validated with public datasets and real-world datasets from photovoltaic production lines. The algorithm shows strong potential for detecting a wide range of defect types, even under challenging conditions. Results demonstrate that this algorithm can detect small changes and defects, thus ensuring higher product quality in automated production lines. Other projects [93], [35] focus on the detection of photovoltaic panels and their mapping using ArcGIS [95] and deep learning. This work involves the integration of deep learning with Geographic Information Systems (GIS) for detecting and mapping PV panels across North Rhine-Westphalia using remote sensing imagery.

The main goal of [24] was to train a neural network to automatically identify and map PV panels. First, high-resolution satellite images had to be collected and annotated in order to create a training dataset. Then, a neural network was trained to recognize PV panels according to their shape, size, color properties, and brightening conditions in different aspects. This system allows for adjustability to new locations and the addition of other features, like wind turbines or streetlights. This addresses challenges related to changes in color properties and optical deformation due to panel orientation, hence making this a very robust solution for large-scale PV panel mapping.

DeepSolar [122] and SolarFinder [63] projects made the most notable contributions to the area of solar panel detection. DeepSolar is a deep learning framework for identifying and mapping solar panels across the United States from high-resolution satellite images. The convolutional neural network is trained on a large labeled dataset of satellite images, enabling accurate detection of solar panels across diverse geographical and environmental conditions. Contrary to this, SolarFinder does the detection and mapping for solar panels using machine learning algorithms along with image processing techniques. Both have shown a high accuracy of model predictions for the different parts of the US, but the datasets come from the same provider, which puts questions on how robust these solutions are to other datasets from other providers. In contrast, the studies [88], [49] have demonstrated effectiveness by targeting specialized applications with highly customized datasets. These datasets are often tailored to specific needs, such as airborne images captured around selected areas of interest. This approach enables more precise detection and analysis within specific contexts, ensuring that the methodologies are tailored to the unique characteristics of the data. This often leads to poor detection performance on other datasets and a range of different environments. Another approach [91] utilizes thermal images captured by Unmanned Aerial Vehicle (UAVs) in the identification of photovoltaic panel anomalies. In this methodology used by [112], UAVs equipped with thermal cameras that capture images of PV panels are analyzed with machine learning algorithms in recognizing hotspots and other anomalies that may indicate faults. While this combination of image types can be helpful, thermal data are expensive and not available for most regions of interest.

The most accurate contributions in this area originate from DeepSolar [122] and SolarFinder [63]. Although both projects achieved major improvements in model accuracy for the US, their datasets originate from the same source, which poses questions about

the robustness of different datasets. Other research [91] has been successful within narrow domains using specific datasets—like airborne images focused on particular areas, but these often perform poorly on different datasets and environments. Other projects [91], [88], [49] make use of thermal images to detect photovoltaic panel anomalies. While there is value in combining image types, thermal data is both expensive and not publicly available.

Related Work	**Contributions**	**Data Source**	**Algorithms**	**Focus**
DeepSolar	Detection framework for residential solar panels, using 7000 aerial images and Inception-v3.	Aerial images	FCN, Inception-v3	Residential areas
SolarFinder	Utilizes satellite imagery and Open Street Map pre-screening, combining SVM, RBF, and CNN.	Satellite imagery	K-means, SVM, RBF, CNNs	Buildings
SolarNet	Maps solar plants in China with U-Net and EMANet for various environments.	Solar plant (satellite) images	U-Net, EMANet	Large plants, complex environments

Table 2.1: Condensed Summary of Frameworks

2.2.1 DeepSolar

DeepSolar [122] is one of the most widely referenced frameworks in the field of solar panel detection, particularly in residential areas. This framework leverages state-of-the-art deep learning techniques toward the analysis of satellite imagery to create a comprehensive and accurate database of solar photovoltaic PV installations. This serves as a critical framework, as the ability to generate an almost comprehensive map of solar installations across the contiguous United States carries significant implications for research, policy-making, and the solar industry.

The ground truth data for the DeepSolar project consists of 366,467 satellite images, selected to represent a diverse range of regions across the U.S., both rural and urban residential areas. The dataset provides the backbone for training and testing the model. Solar panel detection from satellite imagery is challenging, considering the relatively small size of panels in high-altitude imagery; it can only be tackled by sophisticated algorithms. DeepSolar uses an FCN architecture that relies fundamentally on the pre-trained Inception-v3 model [108], trained on 1.28 million images over 1,000 classes. Inception-v3 was deliberately chosen, allowing the framework to capitalize on this pre-trained model's capability to recognize intricate image patterns. This is done by retraining only the final affine layer of the Inception-v3 model while fine-tuning the other layers to adapt it to the specific task of solar panel detection. This fine-tuning process gives way to the optimization of model parameters concerning the detection of solar panels in aerial imagery, thereby increasing the precision and recall of the model. Each image in the dataset is assigned a binary label denoting the presence or absence

of solar panels, which forms the foundation of the classification task. To address the challenge of segmenting solar panels from the background, the project employs a semi-supervised approach. The model generates class activation maps (CAMs) based on image-level labels, highlighting the regions most likely to contain solar panels. This allows the model to accurately estimate the location and size of solar panels without requiring extensive pixel-level annotations, significantly reducing the annotation workload while maintaining high accuracy in detection and segmentation. DeepSolar starts with a crucial preprocessing step that systematically enhances raw satellite images to improve model performance. This includes applying noise reduction techniques and increasing contrast to better distinguish objects within the images. Due to the nature of satellite imagery, variations in perspective caused by angles and tilts can introduce distortion, complicating the detection process. To mitigate these effects, image registration is performed, aligning the images to ensure geometric consistency for further analysis. After preprocessing, the pipeline proceeds to image segmentation using fully convolutional networks (FCNs). These networks excel at producing object maps that identify the precise locations of segmented objects within an image, in this case, solar panels. This enables more accurate identification of regions where solar panels are likely to exist. DeepSolar further refines these segmentation results through post-processing, enhancing the overall accuracy of object detection. This refers to the removal of false positives, where regions were mistakenly identified to contain solar panels, and reduction of true negatives, which are areas incorrectly marked as not containing solar panels. The steps described above are visually represented in Figure 2.1 which illustrates the complete processing pipeline, visually breaking down each critical step. From noise reduction and contrast enhancement to image segmentation and post-processing, every stage is captured to show how raw satellite data is transformed into accurate predictions of solar panel locations.

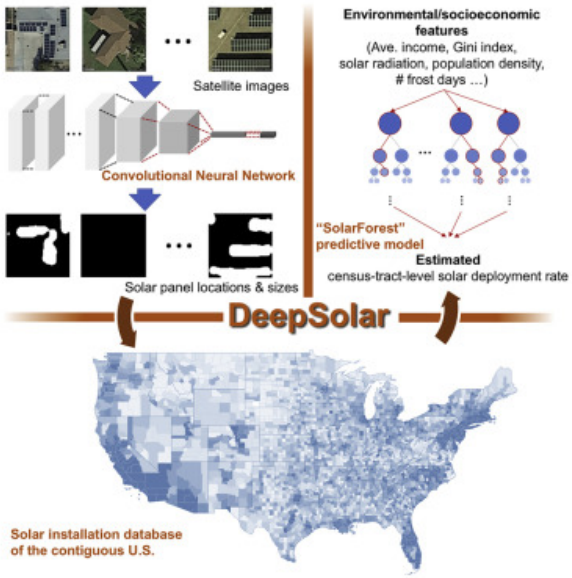

Figure 2.1: DeepSolar processing pipeline. Source: DeepSolar [122]

The detected solar panels will then be grouped into separate buckets depending on their shapes, sizes, and orientations. This categorization helps not only in refining the detection process but also provides the model with enrichment by facilitating generalization capabilities across a wide range of solar installations. The final product of the DeepSolar framework is a rigorously compiled database[1] of installations of solar panels, predominantly covering the contiguous United States. It provides very important information on the gps location, size, and orientation of the detected solar panels, making it a useful database for a wide variety of stakeholders. This dataset can therefore be used by researchers to quantify trends in the adoption and deployment of solar photovoltaic systems, while the policymakers can leverage such insights in designing and implementing better energy policies. On the other hand, the solar industry benefits from the capability for "informed identification of regions for new installations" and the optimization of existing infrastructures.

Besides, the DeepSolar framework is made scalable and adaptive to be continuously updated whenever new data becomes available. This ensures the database remains current with any recent trends in solar deployment. The fact that this database would be publicly accessible means that DeepSolar not only furthers the state of research in solar energy but, therefore, also serves as one of the most important means for accelerated evolutions toward more sustainable energy systems. The success of this project highlights the effectiveness of combining deep learning with satellite imagery for analyzing existing solar panel infrastructures.

2.2.2 SolarFinder

The SolarFinder [63] project is the state of the art in automatic solar PV array detection, considering that it uses a large dataset and the most current and complex machine learning techniques for improving the accuracy and scalability of the detection of solar panels. The dataset that has been used in SolarFinder is composed of 269,632 satellite images collected from 13 different geospatial regions of the United States, which are representative of all kinds of environments and building types. This extensive dataset is openly available and represents a very important source for further research and development in the field.

The first step in the SolarFinder processing pipeline for such large satellite images identifies just the building rooftops. This is achieved using data from OpenStreetMap [80], which provides detailed building footprint information. By utilizing OSM data, SolarFinder can effectively narrow its focus, within the images, to places where the chances that the solar panels may be located are while significantly reducing the computational resources for further processing steps. The segmented images containing only rooftop areas are then processed with the K-means clustering algorithm [40]. This unsupervised machine learning technique has been used in automatically segmenting rooftop images at distinct contours by color, texture, and other visual features. The K-means algorithm works by grouping pixels with similar characteristics into clusters, each cluster repre-

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

senting a different object or feature on the rooftop that may be solar panels, chimneys, or other units. The result of this clustering process is a set of contours, where each one of them is a candidate to include a solar photo-voltaic array. With this aim, SolarFinder sorts these contours for the presence of solar panels by adopting a hybrid machine learning model combining Support Vector Machine (SVM) using the Radial Basis Function (RBF) kernel with Convolutional Neural Networks [52]. The features that are extracted from these contours include the mean and standard deviation of grayscale values inside the contour, the mean and standard deviation of the blue color channel, the ratio between the width and height of the contour and the number of edges and corners detected inside the contour. Afterward, this set is fed to the SVM-RBF model for analysis. In this regard, Principal Component Analysis (PCA) [54] has been employed to reduce dimensionality, ensuring that only the most relevant features are utilized for the classification task. PCA selects directions in data space that correspond to the features contributing to the most variance therefore providing the most information to separate solar panels from other objects. The CNN part of this hybrid model further refines the classification process. In convolutional neural networks, there are multiple layers of convolutional followed by rectified linear unit (ReLU) activation functions and pooling layers which down-samples the input images. Initially, the input size fed to the CNN is 150x150 pixels. This size is sufficient to maintain processing efficiency while providing the necessary resolution to accurately detect solar panels. The last set of features—shapes of lines, textures, and patterns—are being extracted from these input images. The last two are becoming more complex and characteristic of solar photo-voltaic arrays. The last set of features is then used by the fully connected layers of the CNN to classify the contours as containing solar panels or not. The last part of the classification process in SolarFinder involves the combination of the outcomes from the SVM-RBF and the CNN models. It first uses a linear regression model that assigns weights to the outputs of both models in relation to their contribution toward the accuracy of classification. The linear regression model greatly combines the complementary strengths of both the SVM-RBF and CNN approaches, hence robustly classifying the contours more accurately.

Figure 2.2: SolarFinder processing pipeline. Source: SolarFinder [63]

The whole processing chain for SolarFinder, represented in Figure 2.2, has been cautiously designed to be super-efficient and highly scalable; therefore, it enables the processing of huge volumes of satellite imagery while detecting solar PV arrays over extensive geographical areas. This hybrid model then decides whether the contours are solar panels based on an analysis of the contours from the previous steps, while the final classification is done using the linear regression model. The entire pipeline is tuned to

work with publicly available satellite imagery so that SolarFinder can be deployed in a wide range of settings without having access to costly high-resolution data.

Apart from the solar PV array detection, SolarFinder does detailed profiling for every detected array, estimating such attributes as panel size, orientation, and degree of shading. Size is estimated by counting the number of pixels within the identified contour, multiplied by the area represented by each pixel, which in turn depends on the zoom level and the location on Earth. An estimate of the orientation of the solar array would be determined by the angle between the Minimum Bounding Rectangle (MBR) [55] of the contour and the real orientation of the rooftop, providing crucial information, that energy generation from the solar array would strongly depend on the orientation and shading of the panels.

In general, SolarFinder is a huge stride in the domain of detection of solar PV arrays. This is indeed a scalable and accurate solution to the problem of identifying and profiling solar installations using freely available satellite imagery. In this regard, the proposed hybrid approach for SolarFinder combines the strengths of both SVM and CNN models. Thus, it can effectively discriminate between solar panels and other rooftop objects in challenging environments with either low or regular-resolution satellite images.

2.2.3 SolarNet

SolarNet [47] is a deep learning-based framework that charts solar power plants across China from space using large-scale satellite imagery. The primary function of SolarNet is to detect and map solar farms automatically, which holds great importance in the management of solar energy resources across China. This framework successfully identified 439 solar farms, which were built on nearly 2,000 square kilometers-an expanse as wide as the city of Shenzhen. This study represents one of the first large-scale applications of deep learning to map solar farms in China and informs solar power companies, market analysts, and government agencies.

At the heart of SolarNet's architecture is the Expectation-Maximization Attention Networks (EMANet) [65], a deep learning model for semantic segmentation. The EMANet strives in challenging environments for most Chinese solar farms, which are located in deserts, mountains and even lakes. The EMANet module of SolarNet captures both local and global information of satellite images and hence can segment the solar farms from the background with high accuracy, even when the backgrounds are difficult to process. The architecture of SolarNet also partly consists of the U-Net [101] model, which is considered a well-known CNN model used for image segmentation. U-Net features a symmetric architecture, consisting of a contracting path that captures contextual information and an expansive path that facilitates precise localization, making it highly suitable for pixel-level predictions. In contrast, U-Net only acts here as the baseline, ensuring that the algorithm guarantees strong segmentation, while SolarNet enhances it with the multitasking capability of EMANet, improving the accuracy by leveraging both techniques. It was trained with different satellite images of sizes starting from 512 x 512 pixels to 10,000 x 10,000 pixels. Several data augmentation techniques used for improving generalization of the model include random cropping, scaling and rotation.

Performance is measured in terms of mean Intersection over Union (mIoU), and SolarNet outperforms all the baseline models with a mIoU of 94.21%. Trained, SolarNet was used to map solar farms across China, successfully detecting close to 500 farms. Most of these were located in the northwestern regions with ideal conditions of the sun. Challenges for SolarNet lie in cases of farms that are very similar to their surroundings, pointing at future improvements like including more diverse data in training and employing hyperspectral imagery for better detection.

In general, SolarNet marks a significant advancement in the automated mapping of solar energy resources, showcasing the potential of deep learning in large-scale renewable energy management. With further refinement, it could become an even more powerful and indispensable tool for monitoring and optimizing solar power production globally.

2.2.4 Robustness in Solar Panel detection

Robustness is a key challenge for satellite and airborne imagery analysis, as these methods need to perform reliably in a wide range of conditions that can greatly affect their accuracy. Some of the inconsistencies include changes in resolution, illumination, and atmospheric conditions, such as clouds or haze, in the images taken from satellites that could mask or distort the features that the given method tries to catch. Solar panels can look quite different depending on the time of day, weather conditions, and how they are installed. In desert areas, they must handle extreme heat and dust, while in cities, they might be placed on rooftops where nearby buildings cause shading. On mountain tops, they may be positioned at specific angles to catch more sunlight. These varying environments and mounting methods not only change how solar panels look but also affect their performance. This diversity highlights the importance of flexible detection and monitoring systems to ensure efficiency across different settings. This can make it hard for models to generalize since they may have been trained on data that does not fully capture the range of possible conditions they will encounter in practice. Moreover, due to the difference in orientation and tilt, the physical characteristics of solar panels can be so different that it further complicates the detection and segmentation because different materials are used for the panel surface [71]. In a complex environment with a high level of vegetation, reflective surfaces, or an overlapping background of artificial structures, there is more likelihood of misclassification. Both are challenges of importance that show up in the working of both the SolarFinder [63] and SolarNet [47] systems for consistent detection through geospatial regions of different environmental complexities. Consider the challenge faced by SolarNet when analyzing solar farms in China, where the terrain varies significantly. This diverse landscape creates a complex and often confusing backdrop, making it more difficult for the models to differentiate between solar panels and their surroundings. Moreover, the limited availability of training data intensifies the robustness challenges. The model tends to become biased toward specific environments or conditions where image data is readily available, leading to overfitting on those scenarios in which it has been predominantly trained. Performance often suffers when models are deployed into new contexts that are underrepresented in the training set. The lack of variation in the training [56], [103] data often leads to overly special-

ized models incapable of sustaining any reasonable degree of accuracy when confronted with the entire range of scenarios that occur in reality. The need for robust methods to cope with this variability is crucial for the reliability of an individual prediction and the reliability and scalability of such technologies in application over large, varied regions.

2.3 Research on Improving Robustness of Deep Models

In deep learning, robustness refers to the ability of models to maintain consistent performance across varying conditions and resist adversarial attacks. The robustness of the model exhibits an ability that would be resilient against those uncertainties, errors, and deviations encountered quite frequently in real-world applications. It becomes especially critical in dynamic environments, where shifts in natural data distributions can cause models to encounter out-of-distribution (OOD) scenarios, leading to significant drops in performance if not handled properly [60], [26].

Out-of-distribution (OOD) refers to those scenarios where models are asked to make predictions on data that differs significantly from the instances they were trained on. Most of them work poorly in this respect. That is very important in tasks such as satellite image analysis, where changes in the environment or seasonal conditions may pose unexpected challenges. The model should be invariant to changes in weather or geographic location when identifying solar panels, as many of these variations may not be well-represented in the training set. This variability necessitates robust mechanisms to ensure the model's reliability. The best way to understand the complexity of OOD scenarios is by comparing in-distribution (ID) and OOD generalizations. ID generalization refers to the performance of the model on data that is sampled from the same distribution in which it was trained. On the other hand, OOD generalization is required when the model is exposed to data that deviates from the training distribution.

A crucial distinction exists between a model's accuracy on ID data and its robustness to OOD data. While high ID accuracy signals the learning of training set-specific patterns, the better OOD robustness would reflect the model performance in real-world scenarios where conditions are far from being controlled. Therefore, modern research is more geared toward bettering OOD robustness in order to make models resilient against conditions that may not have been seen. Techniques like data augmentation and domain adaptation are at the forefront of efforts to improve this resilience by introducing variability during training and encouraging the model to learn more generalized features [60].

In the case of ID shifts, both the training and testing data share the same underlying statistical distribution. Mathematically, ID shifts are represented as:

$$P_{\text{train}}(X, Y) = P_{\text{test}}(X, Y)$$

where $P_{\text{train}}(X, Y)$ and $P_{\text{test}}(X, Y)$ denote the joint probability distributions of the training and testing data respectively. Concretely, suppose we have a model trained on some subset of the DeepSolar dataset [22], but then it is tested on some other subset of that very same dataset. This model has been tested under an ID shift. The model is expected

to generalize effectively when the test data closely aligns with the characteristics of the training data.

Various types of OOD shifts exist, each posing unique challenges. Covariate shift occurs when there is a change in the marginal distribution of the inputs $P(X)$, without any disturbance in the relationship between inputs and labels $P(Y|X)$. This can lead to performance issues in machine learning models, as they may not adapt well to the altered input distribution. For instance, a model which has been trained to recognize solar panels when the sky is clear can be tested against images with cloudy conditions. In this scenario, the shift in weather conditions altered the input distribution; however, the underlying relationship between input features, such as the geometric structure of the solar panels, and the corresponding labels remained unchanged. Consequently, label shift is said to occur when the label distribution $P(Y)$ changes, but the conditional distribution of the inputs given labels $P(X|Y)$ remains fixed. This can occur when a model is trained in regions with a high density of solar panel installations but is tested in areas where installations are less prevalent. Domain shift involves changes of both the input distribution, $P(X)$, and the conditional distribution, $P(Y|X)$. Consider as a good example a model that has seen only high-resolution satellite images for training; at test time, lower-resolution images taken by another satellite are fed to the model. In such a case, both the panel patterns and the image quality will differ.

Recent studie [92] have established that multimodality in the learning architecture also benefits the models, combining various input data sources to improve OOD generalization. As such, the CLIP model, trained using both image and text data, has shown amazing robustness in handling distribution shifts. Such models learn to associate visual information with textual information, hence generalizing better across diverse datasets and domains.

In the following subsections, we will examine several key methodologies that improve robustness in deep learning models, such as adversarial training, data augmentation strategies, and the implementation of multimodal architectures.

2.3.1 Data Augmentation

Data augmentation is a powerful technique that enriches the training dataset by applying a variety of transformations—such as rotation, translation, scaling, and noise addition—enhancing the model's ability to generalize effectively. Generally, data augmentation tries to increase the diversity of training data so as to make the model robust against various variations in the real world.

Several data augmentation techniques have been considered and implemented to increase model robustness. Random-angle rotation makes the model invariant to the orientations that objects in the image may take. This is particularly useful when scenarios place the subject in different orientations. Horizontal, vertical, or both translations shift the object in the frame. This technique enables the model to recognize objects that are not exactly in the center of an image. Scaling involves resizing the images through zooming in or out. This variation makes the model handle different sizes of objects in the images and hence makes it resilient in terms of scales. Adding random noise into

the images makes the model strong against noisy data, which is quite a common case in real-world scenarios. Addition of noise can particularly help models that have to work in conditions where the environment is less controlled.

Empirical studies have shown that data augmentation techniques significantly upgrade the performance and robustness of machine learning models. For example, Kim [57] shows that rotations, translations, and addition of noise increased the accuracy of the model in image classification problems by making it robust to real-world scenarios. Li [64] also investigated the efficiency of data augmentation in generating high-dimensional image data. Their work demonstrated that amplifications, such as scaling and rotation, are useful in image creation, which increases diversity, hence improving the performance of models involved in image-to-image translation.

Another recent paper by Rebuffi [94] aims at mitigating robust overfitting in adversarial training with standard data augmentation schemes. A variety of methods, such as CutMix [123] and MixUp [124], improve robust accuracy significantly when combined with model weight averaging. CutMix is a data augmentation technique that differs from Cutout [25] by not merely removing pixels from an image. Instead, it replaces the removed regions with patches from another image, effectively blending parts of two different images. This approach not only helps in maintaining the overall structure of the image but also introduces additional variability in the training data, encouraging the model to learn more robust features and improving its ability to generalize across different scenarios. The ground truth labels are also mixed proportionally to the number of pixels of combined images. Mixup is another data augmentation approach creating a weighted combination of random pairs of images in the training data. AugMix [43], proposed recently by Hendrycks [44], has become an efficient data processing method using a combination of augmentations like rotation, translation, and channel mixing for improving the robust performance of deep neural networks.

The referenced articles demonstrate that augmenting datasets through various transformations introduces greater diversity in the data distribution, enabling models to be trained on a broader spectrum of variations, which subsequently enhances their capacity for generalization and improves robustness against overfitting.

2.3.2 Adversarial Training

One specially designed approach is the adversarial training method, which can help enhance the robustness of machine learning models by training them on adversarial examples. The adversarial examples in this case are inputs that have been intentionally perturbed to provoke incorrect predictions from the model. At training time, this exposition of the model to such adversarial attacks positions it to better withstand such attacks at deployment.

The basic principle of adversarial training is a process for building adversarial examples and including them in the train set. As soon as the model has learned from such challenging examples in addition to the original data, it will then be more robust. Among the first studies on this, Goodfellow [39] introduces adversarial examples and demonstrates that adversarial training helps make a model more robust. In this work,

they introduced the Fast Gradient Sign Method (FGSM), which is applied to generate adversarial examples by adding a small perturbation in the direction of the gradient of the loss function concerning the input.

Recent improvements have been proposed to further optimize the adversarial training methods. For example, Madry [74] proposed a method using Projected Gradient Descent (PGD) for generating stronger adversarial examples. By definition, this approach iteratively perturbs the input, taking small steps in a gradient direction of the loss function and projecting the perturbed input back into an allowed input space at each step. This has gained wide adoption since it is pretty good at creating robust models.

To that end, Ma and Liang [72] designed an end-to-end adaptive-margin adversarial training (AMAT) method. In contrast to SAT, where a fixed level of noise is injected, in AMAT, adaptive adversarial noises are generated and are adaptive to individual training samples. This dynamic adjustment helps keep the accuracy of the model unchanged on clean data but improves robustness against adversarial attacks. Experiments on medical image segmentation, landmark detection and object detection tasks showed that AMAT outperformed SAT in both robustness and accuracy. The key idea of AMAT is to adapt the magnitude of perturbation based on current model performance for each training sample. In such a way, it keeps the amount of perturbation under very rigorous control; hence, adversarial training becomes more effective.

It is possible to apply common data augmentation schemes to adversarial training to mitigate robust overfitting, as proposed by Rebuffi [94]. Techniques such as CutMix and MixUp, discussed in the previous section, significantly improved robust accuracy when combined with model weight averaging. CutMix, as described earlier, augments image data by replacing removed regions with patches from another image, instead of simply removing pixels as in Cutout. Ground truth labels are proportionally mixed based on the number of pixels from the combined images. MixUp, also previously covered, creates a weighted combination of random pairs of training images to enhance data diversity. Additionally, Hendrycks [44] introduced AugMix, a data processing strategy that leverages diverse augmentations to further strengthen the robustness of deep neural networks.

There is empirical evidence from many studies that shows the efficacy of adversarial training as a means to improve the model's robustness. For example, Tramer [111] studied how the models trained with adversarial examples work against adaptive attacks. Their results showed that adversarial training noticeably boosts robustness against most attacking approaches. They also highlighted the ongoing advancements in adversarial training methodologies to keep pace with emerging attack techniques.

To conclude, techniques for the adversarial training of machine learning models are one of the cornerstones in increasing robustness. In the process, adversarial examples help make models resistant to malicious attacks. Improvements, among them adaptive-margin adversarial training and further innovations that have taken place with the incorporation of data augmentation methods, make this strategy much more effective. Future research will be in a position to further investigate new strategies for the optimization of adversarial training and its applications in various domains. Continued improvement of the methods of adversarial training is key to ensuring and enhancing security and

reliability in real-world applications of machine learning models.

2.3.3 Ensamble Methods

Ensemble methods are techniques through which models could be combined to reduce variance and enhance robustness. In such techniques, one would take advantage of the fact that different models may be able to capture different aspects of the data distribution. This is potentially very powerful, increasing model accuracy and generalizability. Basic strategies in the use of ensembles include bagging, boosting, and stacking [78]; each strategy has its own way of fusing outputs from different models into a more robust model.

Bagging, or Bootstrap Aggregating [29], means training several instances of a model on different subsets of the training data. Basically, these subsets are created through bootstrapping. At the end, predictions from these individual models are combined—usually by averaging in regression tasks or majority voting in classification tasks. It helps in reducing variance and can be applied to avoid overfitting. Breiman [6] introduced bagging and showed how it minimized model variance and increased the robustness of models. Bagging is especially useful for high-variance algorithms like decision trees; with random forest, several trees are combined to create one much more accurate and stable model.

Boosting [8] is another very effective way of ensembling. It trains models sequentially, and each new model corrects the errors of previous models. The final predictions are a weighted sum of all the predictions. Probably the most well-known boosting algorithm is that proposed by Freund and Schapire [30], who significantly improved model performance by iteratively readjusting weights on the training samples based on the errors of prior models. Gradient Boosting generalized it into the optimization of a loss function in an iterative way, introduced by Friedman [31], which made it a robust approach for both classification and regression. The boosting methods have been applied to a number of fields, from finance and healthcare to marketing, due to the fact that the combination of weak learners gives very accurate models.

Stacking, also known as Stacked Generalization [117], involves training multiple models whose predictions are combined as inputs to a second-level model, commonly referred to as a meta-learner. The second-level model will learn how to combine base models' predictions in the best possible manner. Wolpert [117] introduced stacking, which has had great success in most cases, improving model performance by capturing the strengths of different algorithms. Stacking is a flexible method, and it can be applied to any combination of models; for this reason, it has gained significant popularity in competitions and practical applications where predictive performance is maximized.

Versatile studies [76], [120], [13] have demonstrated the advantages of ensemble methods in varied applications. For example, Wyatt [120] applied ensembling methods with the purpose of improving the robustness of deep learning models on ecological data. In this respect, the authors were able to demonstrate an increase in robustness to noise, outliers, and better prediction accuracy due to the ensembling of several models. Casado-García and Heras [13] developed an independent ensemble algorithm for object detection that is applicable regardless of the underlying model applied for detection. Their ap-

proach reached performance improvements of up to 10% when predictions of different models were aggregated and test-time augmentation applied.

Certified robustness [62] refers to formal guarantees about a machine learning model's resilience against adversarial attacks. These guarantees ensure that the model behaves reliably within a specified range of input perturbations, providing a mathematical assurance that small changes in input won't lead to incorrect predictions or classifications. Jia [50] investigated methods for improving a model's robustness against an adversarial attack by ensembles. They found out that ensembles give much better defense mechanisms and improve the bounds for certified robustness compared to individual models. This work points out the potential of ensemble methods in making machine learning systems much more safe and reliable.

Ensemble methods have been applied with success to improve the robustness of models in practical applications. For example, Casado-García and Heras [13] implemented an ensemble method that combines the output of different object detectors, improving the accuracy of object detection and reducing false positives in object detection. In the approach, there is included a test-time augmentation procedure intended to improve the performance of models by applying a variety of transformations to the test images and appropriately combining all the predictions.

Another notable application is in ecological data analysis, where Wyatt [120] used ensemble methods to improve the robustness of deep learning models against noisy and incomplete data. By combining multiple models, they were able to achieve more reliable and accurate predictions, which are crucial for ecological monitoring and conservation efforts.

This flexibility and efficiency of ensemble methods make them very useful in modern machine learning. They not only enhance the predictive performance but also improve model robustness and reliability. Future work on ensemble methods can focus on investigating more sophisticated techniques such as Dynamic Ensemble Selection (DES) [58] and hybrid ensembles [14]. DES selects the most competent models with respect to each input instance from an ensemble, by which it improves prediction accuracy through emphasis on the models that perform best in the local region of the feature space. In contrast, hybrid ensembles combine several ensemble methods or different types of models to improve performance by harnessing all their advantages in constructing a more robust and adaptive predictive model. Both techniques are trying to enhance ensemble learning either by fitting model use to the right conditions or applying complementary methods.

Ensemble methods can improve the robustness and performance of any machine learning model. It can combine the strengths of different models and apply bagging and boosting to ensembles to reduce variance, avoiding overfitting and improving generalizability. Ensemble methods have seen continuous improvements in pushing the edge on what might be realized in Machine Learning, therefore becoming an important component for developing any robust Machine Learning strategy. Future work can be devoted to more sophisticated ensemble strategies, with their applications over a wide range of divergent domains, thus strengthening the role of ensemble methods in advanced machine-learning practice.

2.3.4 Robust Loss Functions

Robust loss functions are designed to be less sensitive to outliers, making models more resilient to noisy data. Traditional loss functions like Mean Squared Error (MSE) can be heavily influenced by outliers, leading to suboptimal performance. Robust loss functions address this issue by modifying the loss calculation to reduce the impact of these anomalies. Prominent examples include the Huber loss and quantile loss, among others.

The Huber loss, introduced by Huber [48], comes in handy in regression tasks where data contains outliers. It is a combination of the good properties from Mean Squared Error (MSE) and Mean Absolute Error (MAE); it is quadratic when the error is small, and linear when the error is large. This duality enables the Huber loss to handle outliers more gracefully than MSE. Formulation for the Huber loss goes as follows:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta) & \text{for } |a| > \delta, \end{cases}$$

where δ is a hyperparameter that determines the point where the loss function transitions from quadratic to linear.

Quantile loss, another robust loss function, is particularly useful for tasks where the goal is to predict a specific quantile of the target distribution, such as in quantile regression. This loss function is defined as:

$$L_\tau(a) = \begin{cases} \tau a & \text{if } a \geq 0, \\ (\tau - 1)a & \text{if } a < 0, \end{cases}$$

where τ is the quantile being estimated. This approach is effective for dealing with skewed distributions and outliers, providing a more nuanced understanding of the underlying data distribution.

Barron [2] provided a general and adaptive robust loss function that subsumes many of the existing robust loss functions within one common framework. Through its parameters, this can change to MAE, MSE, Cauchy loss, etc., into other types of loss functions. Added with this flexibility, it hence comes very handy in quite a range of scenarios replete with large outliers or noisy data.

Recent work by Gonzalez-Jimenez [37] introduced T-Loss, a robust loss derived from the negative log-likelihood of the Student-t distribution. It is a loss function for medical image segmentation and does handling of outliers quite elegantly because of sensitivity control through one parameter. This parameter itself gets updated during the process of backpropagation without any extra computation or prior information on the level and spread of noisy labels. Their experiments proved that the T-Loss outperformed traditional loss functions in dice scores for skin lesion and lung segmentation tasks in public medical datasets, thus proving it to be very effective in noisy environments.

Ghosh [34] benchmarked various robust loss functions against label noise for deep neural networks. Their paper showed that robust loss functions such as Generalized Cross Entropy (GCE) and Normalized Generalized Cross Entropy (NGCE) obtained large improvements in model performance against noisy labels. These results make the loss

functions particularly well-suited for quite diverse applications, like image classification and medical diagnostics.

In the context of label noise, Ma [73] introduced normalized loss functions for deep learning, which normalize gradient contributions to mitigate overfitting caused by noisy labels. Their empirical studies demonstrated improved performance over traditional loss functions, particularly in scenarios with high levels of noise.

In other words, strong loss functions are of prime importance in enhancing the ability of machine learning models to become increasingly resilient to noisy and outlier-prone data. The functions reduce sensitivity in the calculation of the loss due to extreme values and keep model performance and robustness. Particular further research in the area of robust loss functions is able to boost the reliability of machine learning applications in diverse fields, such as medical imaging, finance, or autonomous systems.

2.3.5 Defensive Distillation

Defensive distillation is a technique intended to make neural networks robust against adversarial attacks since it forces the model to output soft probabilities instead of hard classifications. Initially developed for model compression, this method has proven useful in enhancing both the safety and security of machine learning models.

Originally introduced by Papernot [85], the concept of defensive distillation requires training a teacher model to create soft labels using a higher temperature parameter. These labels are smoother probability distributions over classes and are utilized in training the student model. This helps in making the model less confident in its predictions, which allows it to distribute probabilities more smoothly and makes it less sensitive against adversarial perturbations. By smoothing the output probabilities, the model becomes more difficult to manipulate through adversarial attacks, which typically rely on exploiting sharp decision boundaries.

However, Carlini and Wagner [10] have pointed out weaknesses in defensive distillation, showing that it is still possible to bypass adversarial attacks. They developed more sophisticated attack algorithms that adjusted the input in such a way as to make even defensively distilled models misclassify. For example, they showed that the targeted attack on MNIST achieved a very high success rate of misclassification with only minimal changes to the input.

Although the defensive distillation approach does have some limitations, the technique remains an important milestone in the quest for developing robust machine learning models. The approach has inspired further research into combining distillation with other kinds of defense mechanisms to be able to develop more secure systems that can be realized from later works by Papernot [86], [84].

In the context of domain adaptation, multiple techniques have been developed to adapt models trained on a source domain to perform effectively on a target domain, improving robustness and generalization. This section explores additional methods to complement previously discussed techniques like ensemble methods, adversarial training, and robust loss functions. These methods build on the core principles of feature alignment, domain mapping, and minimizing domain shift.

2.3.6 Model Adaption Methods

Different methods have been developed for domain adaptation, each of which adapts models trained on a source domain to perform well on the target domain to improve robustness and generalization. This section discusses more supplementary methods to complement the previously discussed ensemble methods, adversarial training, and robust loss functions. These methods are based on principles of feature alignment, domain mapping, and minimizing domain shift.

Domain-Invariant Feature Learning (DIFL)

The key concept of Domain-Invariant Feature Learning (DIFL) [125] is to align the feature distributions of the source and target domains in a manner that the model treats the features of both domains similarly. The core assumption in this approach is that there is a common feature space where both domains overlap. If the model learns to represent data from both domains similarly, then the performance of a classifier trained on the source domain will be effective on the target domain.

The Domain-Adversarial Neural Network (DANN) was proposed by Ganin [33] for this challenge. The idea of DANN relies on adversarial training between a domain classifier that can distinguish between source and target domain features and a feature extractor trained to confuse this domain classifier to learn domain-invariant features. It is set up in a way that there is competition between the feature extractor and the domain classifier so that learned features are generalized across domains. The adversarial nature of this method keeps the network in reduction of domains-specific information, enhancing feature generalization.

Saito [102] subsequently proposed the Maximum Classifier Discrepancy method, which applies two task-specific classifiers. These are allowed to maximize the discrepancy of their predictions between source and target domains. Later, this discrepancy is minimized for the feature extractor to be allowed to learn features suitable across domains. It will induce the feature extractor to generate domain-invariant representations while maintaining classifier attention to domain differences.

Multi-Level Adaptation

Multilevel adaptation addresses domain adaptation at different levels of abstraction on the neural network. Its intuition is that domain differences might exist at multiple levels; for instance, pixel-level differences regarding color or brightness, and feature-level differences on texture or shape. Having these observations at many steps in the network can help to reduce the domain shift.

Hoffman [45] suggests Cycle-Consistent Adversarial Domain Adaptation (CyCADA) with a dual approach to adaptation. First, at the pixel level, CyCADA performs image-to-image translation to map source images to match target images. For example, synthetic images can be transformed to look in style like real-world images, reducing low-level differences such as lighting or style. Then, at the feature level, it aligns the high-level semantic features learned by the model. These range from low-level to high-level

discrepancies, and by leveraging them in a multi-level fashion, CyCADA ensures domain discrepancies are at a minimum. This is reflected in its strong performance across various tasks, such as semantic segmentation and object detection.

This is essentially based on the idea that high-level feature alignment may not be enough in that it does not rule out the case where input images from different domains could still be very dissimilar under this very same setting. Early-stage alignment at the pixel level helps bridge domain gaps so that feature alignment can refine adaptation.

Batch Normalization Adaptation

Batch normalization (BN) normalizes the inputs of every layer to have zero mean and unit variance, thereby stabilizing and speeding up the training of neural networks. However, in domain adaptation, these BN statistics (mean and variance) may be different across source and target domains, thereby leading to degraded performance when the model is applied to the target domain.

Li [66] addressed this by proposing Adaptive Batch Normalization, AdaBN, which adapts batch normalization statistics w.r.t the target domain. Instead of using source domain statistics for inference, the normalization layers of AdaBN update with target domain statistics, hence effectively adapting the model without modifying the weights of the training. Perhaps the most important advantage of AdaBN, however, is simplicity: it requires no retraining of the model and yet greatly improves performance on the target domain because it accounts for domain-specific statistical differences.

Carlucci [11] extended this concept into AutoDIAL: Automatic Domain Alignment Layers. AutoDIAL introduced domain-specific alignment layers operating along with the batch normalization. The layers will automatically adapt how much alignment is performed at every step through the network so that the model can continuously handle varying levels of domain shift. The underlying concept is to enable the model to learn dynamically how much adaptation a given layer requires in order to enhance flexibility and robustness.

Divergence Minimization

One of the major techniques for reducing domain shift is to minimize the difference between the target and source domains. The idea here is to quantify the difference in the two domains' distributions and reduce this difference to a minimum, thereby pulling the feature representations closer together.

One study [70] includes Maximum Mean Discrepancy (MMD) to calculate the distance between the mean embeddings of source and target domain data in a reproducing kernel Hilbert space. Optimizing this distance will allow aligning the source and target distributions in feature space. Being simple and achieving effective distribution alignment without adversarial training, MMD remains one of the most applied techniques in domain adaptation.

Similarly, Shen [104] extended the DANN model with a modified domain classifier network to learn the Wasserstein distance [83] between source and target distributions. The

Wasserstein distance, also known as the Earth Mover's Distance, calculates the amount of "work" that must be done to transform one distribution into another. Minimizing loss in the transport problem of Wasserstein distance causes smoothness and stability of the alignment of source and target domains, hence providing better adaptation performance.

Self-Ensembling Techniques

In certain domain adaptation scenarios, data from multiple source domains may be available, and combining insights from these different domains can enhance the model's performance on the target domain. Co-training and multi-source adaptation aim to leverage information from multiple domains to learn robust, domain-invariant representations.

Co-Training for Domain Adaptation (Co-DA), introduced by Kumar [59], does this by training two different feature networks where each learns different aspects of the input data. These complementary representations then co-regularize each other by the nature of training, such that both networks must focus on different yet relevant features of the data. This results in reducing overfitting to a single source domain while increasing robustness across domains. Further on, these are combined together in order to generalize well on the target domain.

This was further extended by Zhao [126] using multi-source domain adaptation methods. When the target domain is formed by a combination or mixture of several source domains-as may happen in speech recognition across different environments or speaker variations-multisource domain adaptation looks at aligning all source domains to the target. Adversarial methods that basically minimize the differences between the target domain and each of the source domains individually are implemented. That makes this model more robust by aligning the target domain to a range of source domains, hence handling the variation in the unseen target data.

Multi-source domain adaptation demonstrates great performance in case the domain shifts are not common for all data. In such cases, when the target domain can be considered a mixture of several factors related to environmental or contextual conditions, this can be taken into account. This approach robusts the models by learning from multi-aspects to ensure that the target domain is well-represented across the learned feature spaces.

2.3.7 Model Soups - increase robustness of deep models

Improving model robustness against both in-distribution (ID) and out-of-distribution (OOD) shifts has seen significant progress through the application of a technique called Model Soups, which effectively enhances performance across diverse data conditions [119]. Taking several fine-tuned models and averaging their weights together into a single model can result in a more robust model. Model Soup is based on an observation that starting from the same initialization, deep learning models converge to the same region in an optimization landscape even when fine-tuning with different hyperparameters.

One particular approach within this framework is the $Unified\ Soup$ technique. This

method involves averaging the weights of multiple models that were fine-tuned with different hyperparameters, creating a single model that combines the knowledge from all the fine-tuned models. This approach has the added benefit of not introducing any additional computational cost during inference, as only one model is used. The following algorithm outlines how Unified Soup is constructed:

Algorithm 1 Unified soup

Require: A collection of N models, each with different weights $w_1, w_2, ..., w_M$
Ensure: A new model with weights that are the mean value of all models

Initialize an empty array W of size M
for i in range 1 to N **do**
    Load the i-th model and read its weights $w_{i,1}, w_{i,2}, ..., w_{i,M}$
    for j in range 1 to M **do**
        $W_j \leftarrow W_j + w_{i,j}$
    end for
end for
for j in range 1 to M **do**
    $W_j \leftarrow W_j / N$ ▷ Calculate the mean value of weight j
end for
Create a new model with weights $W_1, W_2, ..., W_M$
return The new model

While Unified Soup offers an effective and simple solution to the problem of model combination, a more sophisticated solution can be obtained by resorting to the *Greedy Soup* technique. Unlike Unified Soup, Greedy Soup adds one model at a time to the mixture but only if its addition improves the overall validation accuracy. It only includes the models that contribute the most and, thus, create a stronger model that will not be negatively impacted by suboptimal models. The process stops in the case when adding further models reduces performance, meaning the final soup will be as strong as possible.

Algorithm 2 Recipe 1 GreedySoup

Require: Potential soup ingredients $\{\theta_1, ..., \theta_k\}$ (sorted in decreasing order of ValAcc(θ_i))
Ensure: the Average of selected ingredients that maximizes ValAcc

Initialize an empty set of selected ingredients *ingredients*
for $i = 1$ to k **do**
    if ValAcc(average($ingredients \cup \{\theta_i\}$)) $\geq$ ValAcc(average($ingredients$)) **then**
        $ingredients \leftarrow ingredients \cup \{\theta_i\}$
    end if
end for
return average($ingredients$)

There are a few very important reasons why Model Soup works so well for improving

robustness: Models trained from the same initialization usually converge to solutions that are often similar; hence, averaging their weights will also yield an optimal model performance. Model weight averaging is a kind of regularization that helps in smoothing out overfitting, especially in OOD cases where test data might be quite different from training data. This has a regularizing effect, whereby the model generalizes to data it has not seen yet. Finally, diversity induced by fine-tuning with various hyperparameters ensures that when weighing these models together, systematic errors by individual models are reduced. This combination of factors makes Model Soup a highly effective method for increasing robustness to both in-distribution and out-of-distribution shifts. The idea of Model Soups is related to ensemble methods, which combine predictions from multiple models to improve overall performance. In general, several models, each trained with a different setting, would independently make predictions, and the final output would be determined through averaging or a vote on the individual predictions. It is understood that in general, this enhances robustness due to variance reduction in model errors. However, ensembling comes with a significant computational cost, as each model must be evaluated separately during inference. Model Soup, on the other hand, ensembles at the weight level and provides one model leveraging the knowledge of multiple fine-tuned models. By averaging the weights of these models, a new model is generated that retains the robustness advantages of ensembling without incurring the additional inference costs typically associated with ensemble methods. This weight averaging approach enhances both in-distribution accuracy and out-of-distribution robustness, particularly in test sets with OOD shifts, as it combines the strengths of multiple models into a single, more generalizable solution.

2.3.8 Robustness measures

One of the most important metrics in machine learning, especially in applications like solar panel detection, is robustness. It enables model reliability under different conditions, including natural data shifts. Robustness is less concerned with achieving high accuracy on the training data itself and focuses more on maintaining reliable performance on out-of-distribution data. The primary objective is to ensure that models generalize well to unseen scenarios and data shifts rather than overfitting to the training set. This perspective on robustness is crucial for practical applications, as it better reflects a model's real-world applicability. It emphasizes how well a model can adapt to and maintain performance under distribution shifts, which is key for ensuring its effectiveness in dynamic environments.

The robustness of a model m is measured by applying it to both the test set of its training dataset D_1 and an unseen dataset D_2, yielding two accuracy measurements: one on the native dataset and one on the naturally shifted dataset. This means that, comparing two models m_a and m_b, we prioritize the model with a smaller gap between its native and shifted accuracies. For example, a model with $accuracy_{native}(m) = 0.78$ and $accuracy_{shifted}(m) = 0.75$ is considered more robust compared to another model where $accuracy_{native}(m) = 0.88$ but $accuracy_{shifted}(m) = 0.76$, due to the smaller gap between two datasets. Previous studies have evaluated robustness on synthetic distribution shifts

[12], [32], [42], but recent work by Taori [109] highlights that current interventions for robustness are less effective on natural distribution shifts.

Another study [20] introduces two measures for assessing robustness: Affinity and Diversity. Affinity shows how far the augmentation of data shifts the position of data around the decision boundary of the baseline model. It is calculated as a ratio of validation set accuracies for the model trained on clean data and tested on an augmented validation set and accuracy tested on the same data without augmentations. The data is separated into D_{train} and D_{val}, both drawn from the same distribution. Augmented validation set D'_{val} is derived from D_{val} with applied stochastic augmentation techniques, represented as $D'_{val} = \{(a(x), y) : \forall(x, y) \in D_{val}\}$. The model's performance is then evaluated on both D_{val} and D'_{val}, with the Affinity measure given by:

$$\mathcal{T}[a; m; D_{val}] = \frac{\mathcal{A}(m, D'_{val})}{\mathcal{A}(m, D_{val})}. \tag{2.1}$$

A small value of $\mathcal{T}$ suggests the model is operating in an out-of-distribution scenario. Ideally, the value of $\mathcal{T}$ should be close to 1, indicating optimal performance.

Diversity quantifies the complexity of augmentations, serving as a metric to empirically test the hypothesis that augmentations mitigate overfitting. This is achieved by augmenting the training set with additional samples, thereby increasing its size and diversity. The metric is computed as follows:

$$\mathcal{D}[a; m; D_{train}] = \frac{\mathcal{E}_{D'_{train}}[L_{train}]}{\mathcal{E}_{D_{train}}[L_{train}]} \tag{2.2}$$

where $\mathcal{E}_{D'_{train}}[L_{train}]$ represents the final training loss on a model trained with a given augmentation, and $\mathcal{E}_{D_{train}}[L_{train}]$ is the loss of the model trained on clean data.

Affinity and diversity are two essential elements in assessing model robustness. Affinity refers to how well a model has internalized the patterns and regularities of the data it was trained on. A highly affine model is closely fitted to these regularities, yielding highly accurate results when tested on in-distribution (ID) data or data that closely resembles the training set. The degree of affinity reflects the model's ability to capture the underlying structure of the data, making it effective for tasks involving similar data distributions. However, a model with excessive affinity might face challenges when encountering out-of-distribution data, where robustness becomes crucial.

High affinity is not sufficient to guarantee robustness. While a model may perform excellently when recognizing data it saw during training, it may fail when unfamiliar data is presented. This brings in the importance of diversity. Diversity refers to the amount and range of inputs that the model has been exposed to during training and testing. A model exposed to more diverse data during training becomes more adaptable to new, unexpected data points. This is because it learns to handle a broader range of input scenarios, allowing for greater flexibility in making predictions on unseen data. Increased exposure to variability helps the model generalize better, making it less reliant on specific patterns from the training set and more capable of handling out-of-distribution data.

This balanced affinity and diversity imply that the model performs well both in fitting the data it has seen during training and in generalizing to new, unseen data. As a result,

the model becomes robust and can handle a wide range of data variations without a significant drop in performance. In real-world applications, where data is rarely static or predictable, this form of robustness defines true reliability. It ensures that models can adapt and maintain their performance when faced with dynamic and unexpected data variations. Robustness, in this context, is crucial for the long-term success of models in real-world scenarios, where data distributions are often subject to shifts and fluctuations.

Problems arise if affinity and diversity are out of balance. A model, though having high affinity, might have low diversity, resulting in overfitting to the training data and only memorizing patterns from that set. This may cause poor generalization to data that is dissimilar to what it has seen before. This is particularly problematic in applications where input data varies significantly over time or across different environments, such as solar panel detection under varying lighting conditions, weather, or angles.

On the other hand, low affinity with high diversity may result in a model becoming overly sensitive to input variations. Such a model might exhibit inconsistent performance as it becomes too responsive to data variations. Without a strong affinity to specific patterns, the model may struggle to converge on a stable understanding or consistent perspective, leading to poor performance on augmented or transformed datasets.

The goal is to optimize both affinity and diversity to ensure accuracy not just for specific data but also for new data groups. For example, in solar panel detection, a robust model should identify panels under various environmental conditions or orientations. Ensuring high affinity for appropriate detection in known conditions and high diversity for adaptation to new conditions makes the model more reliable in real-world settings.

Ultimately, affinity and diversity together provide a more comprehensive view of model performance. A model that scores high in both affinity and diversity will be better prepared to handle complex, dynamic environments where data can change unpredictably. This makes it valuable for practical applications, where robustness is as important as accuracy.

3 Methodology

This chapter describes the methodology used to detect solar panels from satellite images, including the datasets used, the deep learning model used for detection, the data preprocessing and techniques used to improve the robustness of the model, and the evaluation metrics used to assess the performance of the models.

3.1 Pipeline

Next figure displays the pipeline used in this work.

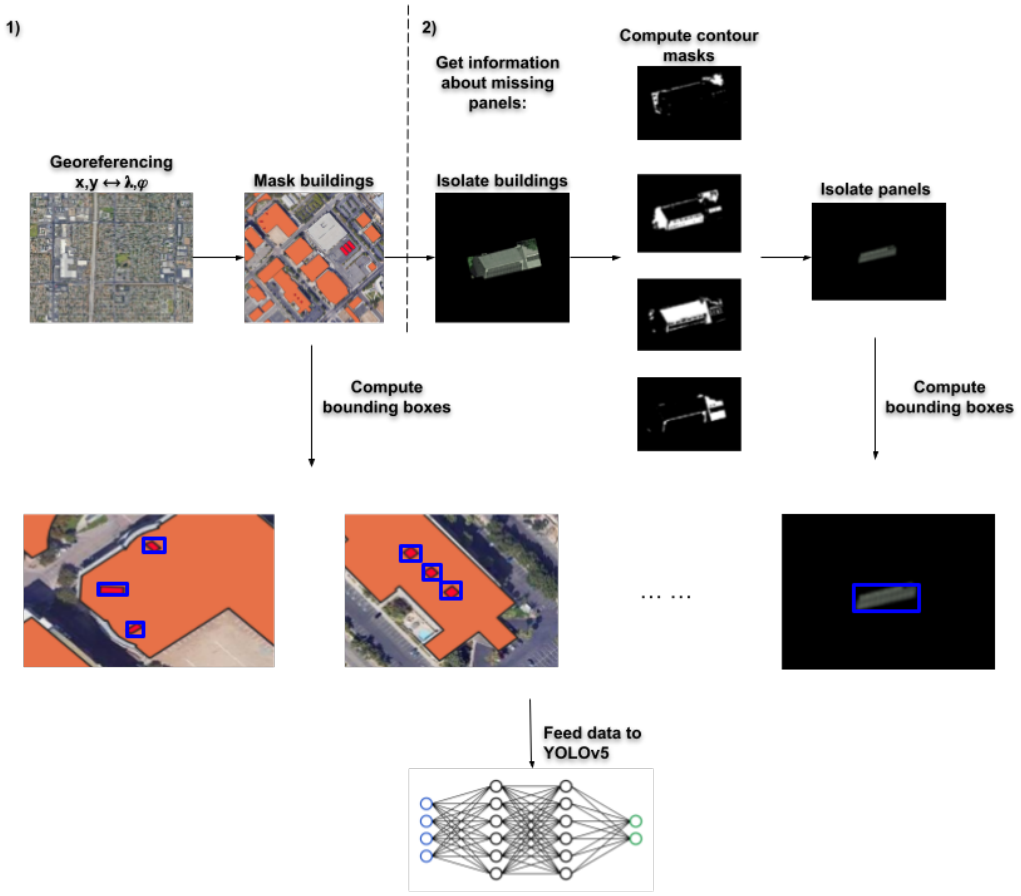

Figure 3.1: Procssing pipeline

This pipeline has a variety of datasets in its stream. Each is further differentiated by several different properties that make it particularly useful for certain use cases within the pipeline. There are two different streams of data within this pipeline, each tailored with specific functionality. First, there is a stream of satellite images with the location of solar panels represented as polygons with the geolocation encoded in the information of the panel. Due to the fact that polygon shapes do not always fit perfect rectangles because of the different panel orientations, bounding boxes are calculated instead. These boxes supply the input feeding into YOLOv5 neural network models. he second data stream, however, is powered by SolarFinder's algorithm and showed poor performance during initial testing when it relied on fixed class numbers for most images. There is no assurance, however, that changing these parameters would return accurate contour detection for the panels. Sometimes, even after the detection of changes, they are grouped with unrelated entities or not detected at all. Once the contour is located, then the calculation of bounding boxes and relevant data is fed to the YOLOv5 model.

In general, this work proposes a multistep pipeline for solving the challenge of detecting solar panels in satellite images. This pipeline uses several datasets with their very unique properties and combines manual and automated techniques for detection and mapping of solar panels. The resulting model enables insights into the effectiveness of various detection methods and offers significant potential for real-world applications in renewable energy.

3.2 Dataset Description

This section provides details on the images and coordinates of solar panels collected from various sources, including Google, Vienna GIS, and others, along with their main characteristics. The resolution of Very High Resolution (VHR) satellite images ranges from a minimum of 2 meters per pixel to as fine as 0.3 meters per pixel. Aerial images typically offer higher resolutions due to their lower altitude flights. Conversely, LEO (Low Earth Orbit) satellites operate at altitudes between 160 and 2,000 kilometers above the Earth. The resolution of aerial images varies from a few centimeters to several meters, depending on the flight altitude and camera specifications. This work utilizes several publicly available datasets, including the Full Coverage Data for Photovoltaic Arrays (FCDPA), GIS data for Vienna, SolarFinder (Google Satellite Images), and DeepSolar (Google Satellite Images). In Figure 3.2, a sample from each dataset is shown.

Figure 3.2: Image samples showcasing different solar panel detection methodologies: DeepSolar (top left), FCDPA (top right), GIS Vienna (bottom left), and SolarFinder (bottom right).

3.2.1 Solar Finder

The SolarFinder dataset [63] comprises images from eight U.S. states displayed in Table 3.1, each with distinct environmental characteristics. These differences stem from urban and rural regions, where rural regions feature objects like vegetation, and urban regions are characterized by buildings.

Location
Fresno, California Palo Alto, California San Jose, California
Adams, Colorado
Waukesha, Wisconsin
Springfield, Massachusetts
Minneapolis, Minnesota
Gilbert, Arizona
Baltimore, Maryland
Seattle, Washington

Table 3.1: Locations in Solar Finder data

Data Description			
Image Source	Image Size	Nr. of Images	Number of Contours
Google	800x800 pixels	269,632	1,143,636

Table 3.2: Statistics for Solar Finder Data

It is important to note that the number of contours from Table 3.2 does not directly correspond to the count of solar panels, as these contours may include other objects such as vehicles, vegetation, and various complex structures extracted from the SolarFinder images. Additionally, not every image listed in Table 3.2 necessarily depicts or contains solar panels.

In the pre-processing stage, the houses have already been masked and excluded in the images. This procedure adds value to the dataset, which contains large data on different roof shapes and various solar panel installations. The method makes the detection of the solar panels easier since it contains less clutter in the images, mainly the structure of roofs. This is, however, not perfect because not all structures can be masked with extreme accuracy, which makes this technique inappropriate for practical applications in real life.

3.2.2 DeepSolar

The DeepSolar dataset [122], originating from the United States, encompasses four geographically diverse states as detailed in Table 3.3, thereby enriching the dataset with a variety of image contexts. The data is sourced directly from Google, providing raw imagery available on Google Maps. In contrast to SolarFinder, which utilizes preprocessed images, DeepSolar employs raw data.

Location
Clark, Washington
Travis, Texas
Santa Clara, California Los Angeles, California
Union, New Jersey Hudson, New Jersey

Table 3.3: Locations in DeepSolar Data

Data Description			
Image Source	Image Size	Data Amount	Nr. of Panels
Google	320x320 pixels	45 GB	41,100

Table 3.4: Statistics for DeepSolar Data

The benefit of this dataset is in the wide coverage and distribution, providing a wealth of mapped data (see Table 3.4), containing up to 45GB of data. However, there are limitations. Firstly, it is exclusively based in the United States and that the great deal of the data comes from California, which is 38% desert area. This climatic condition probably does not offer the best setting for our project, but adds to the strength of the model. Another point is that the data are binary-mapped, where '1' corresponds to the presence of solar panels and '0' corresponds to their absence; there is no specific geo-location information given. Mapping and labeling of the panel positions were therefore done to improve the usability of the dataset, though this was only done on 2672 images.

3.2.3 FCDPA

The Full Collection: Distributed Solar Photovoltaic Array Location and Extent Data Set for Remote Sensing Object Identification (FCDPA) dataset [5] is also sourced from the U.S., specifically California (see Table 3.5). Although its geographical coverage is limited, the dataset includes precisely drawn polygons of solar panels by human annotators, pinpointing their exact positions in the images.

Location
Fresno, California
Stockton, California
Oxnard, California
Modesto, California

Table 3.5: Locations in FCDPA Data

Data Description			
Image Source	Image Size	Nr. of Images	Number of Panels
Unknown	5000x5000 pixels	1,093	19,863

Table 3.6: Statistics for FCDPA Data

Solar panel detection algorithms can be well-trained on the FCDPA dataset, which has extensive size and high resolution (see Table 3.6). Its extensive size and large resolution provide many photos and annotated panels, an extensive and varied set of instances. This goes very well in order to construct reliable machine learning models. But in this case, already the high resolution of the dataset showed a big disadvantage. Real-world imagery seldom reflects these images available in datasets, and very few images are captured at this resolution worldwide. Hence, models trained on this dataset might perform poorly on low-resolution photos or images that carry fewer distinctive features, and the generalization ability of the models is significantly reduced. This impedes its application value in a real-world setting.

3.2.4 GIS Vienna

Aerial images from Vienna comprise this dataset, which includes images from different years and seasons, representing a natural distribution shift beneficial for robustness testing. The GIS data, sourced from government records, includes high-resolution images similar to the FCDPA dataset, yet lacks pre-mapped classifications, presenting a plain dataset.

Data Description			
Image Source	Image Size	Nr. of Images	Number of Panels
GIS Vienna	416x416 pixels	113,403 per year	290

Table 3.7: Statistics for GIS Vienna Data

A manually-labeled subset of this dataset yielded 290 solar panels for analysis, representing a small portion of the total dataset (see Table 3.7). This manual mapping process was essential to create a usable dataset for training and testing solar panel detection algorithms.

3.2.5 Mixed Dataset - Data Preparation and Splitting

In this work, data preparation and splitting for the various models followed structured methodologies designed to optimize the training, validation, and testing processes for solar panel detection algorithms. Various models in this study were based on an 80:10:10 split, where 80% is used for training, 10% for validation, and 10% for testing. However, depending on a number of model types and their requirements, different methods have been applied.

With YOLO, which needs training on images with annotated polygons in order to accomplish object detection, the idea was to construct the training dataset from the available polygon-mapped data. This would involve images from both SolarFinder and FCDPA since both come with pre-annotated polygons of solar panel polygons. These datasets were quite ideal for the object detection tasks of YOLO, given the detailed annotations which are used to precisely detect bounding boxes. A portion of the DeepSolar dataset, which was manually self-mapped, was incorporated to increase the diversity of the training data. Since the original DeepSolar dataset lacked polygon annotations for solar panels, manual mapping was performed to create the necessary polygons, enabling its use for YOLO training. This step was crucial in expanding the dataset, allowing the model to learn to detect solar panels across a broader range of geographical and environmental conditions, thereby improving its robustness.

However, the GIS Vienna dataset was reserved exclusively for the validation phase. The dataset was manually annotated and was used to test the model's performance on completely unseen data. The GIS Vienna dataset introduced a more significant distribution shift, as it included images taken across multiple seasons and years, resulting in considerably greater variation compared to the original training data. This temporal and environmental diversity posed a substantial challenge for the model, requiring it to adapt to a broader range of conditions. This was used only as a validation set in this work, in order strictly to challenge the generalization capability of the model in a new environment and test its robustness under realistic scenarios where seasonal and environmental changes are encountered.

In the cases of all three VGG, Random Forest, and Logistic Regression models, which have been used for binary classification rather than object detection, the entire dataset was utilized. All these models do not require the polygon annotations since their task was to classify if the image contains solar panels or not, so they can use the entire dataset including those without annotations as well. In this case, the split was applied once again, incorporating all available images from the SolarFinder, DeepSolar, FCDPA, and GIS Vienna datasets for the models addressing the simpler binary classification task during the training phase. By using all the available data sources in this particular binary classification problem, it enables the binary classification models to use a larger, more varied training set. This was especially helpful to ensure better generalization across regions, resolutions and environmental conditions since the models were trained on such a diverse set of images.

The YOLO model was trained only on the polygon-annotated dataset, supplemented with hand-mapping of extra data from DeepSolar, and tested on previously unseen data from GIS Vienna to check for generalization. The models designed for binary classification—VGG, Random Forest, and Logistic Regression—were used consistently with an 80:10:10 split. This approach ensured comprehensive coverage of all data sources, enhancing the robustness of the results. It also allowed each model type to be trained and tested optimally, ensuring their effectiveness and ability to generalize well to new data.

Method \ Dataset	SolarFinder	DeepSolar	FCDPA	GIS Vienna
YOLO (OD)	80:10:10	80:10:10 (MM)	80:10:10	Validation (100%) (MM)
VGG (BC)	80:10:10	80:10:10	80:10:10	80:10:10
Random Forest (BC)	80:10:10	80:10:10	80:10:10	80:10:10
Logistic Reg. (BC)	80:10:10	80:10:10	80:10:10	80:10:10

Table 3.8: Overview of dataset usage across different methods. OD: Object Detection, BC: Binary Classification, MM: Manually Mapped for YOLO in DeepSolar and GIS Vienna datasets.

3.3 Description of the Models

This study employs a diverse set of machine learning/DNN models to achieve the desired research goals. While certain models, namely VGG, Random Forest, and Logistic Regression, have been previously implemented [63] within the context of the research area, novel models such as You Only Look Once version 3 (YOLOv3) and YOLOv5 are also incorporated and evaluated in this work.

3.3.1 Random Forest

A machine-learning algorithm, like the $Random\ Forest\ Classifier$, makes a number of decision trees to predict the outcome. This technique is more accurate and reliable than using a single tree, due to the fact that it averages the output from several individual trees. In this work, we have used the classifier with 100 estimators creating 100 decision trees. This choice of 100 trees provides a reasonable middle ground between computational efficiency and high accuracy. Although adding trees to the model may improve its performance, the marginal gains are generally observed to decrease, making 100 a realistic decision when empirically supported by data. The maximum depth of each tree is constrained to be at most two to avoid overfitting. Under such a constraint, each tree should be simple enough so that it is able to handle new information. It also uses class balancing in order to deal with issues such as imbalanced datasets, where particular classes might be overrepresented or underrepresented in the training set. The model makes sure that all classes are equally represented by dynamically changing weights inversely proportional to class occurrence, which has the potential to improve its predictive accuracy in several scenarios. This makes the $Random\ Forest\ Classifier$ a reliable and versatile machine-learning tool.

3.3.2 VGG

Among the most used CNN models, the VGG16 architecture [126] was first proposed for the classification task of large-scale images. Its efficiency comes from a simple deep structure that is comprised of 16 layers, of which 13 are convolutional and 3 fully connected. All the convolution filters in this model are of size 3x3, enabling the capture of both low-level fine details and higher-level patterns within the images. This makes the model

particularly effective for tasks requiring detailed feature extraction, such as solar panel detection. To identify solar panels, VGG16 is very helpful when applied through transfer learning with pre-trained weights on the ImageNet dataset [23]. In other words, it allows the model to initialize parameters that are optimized for general features of images and further tune such weights to learn solar panel identification in satellite images. This reduces the requirement of big datasets of a particular task, hence increasing efficiency along with accuracy. Keras deep learning framework [17] supports pre-trained model integration and their adaptation for task-specific applications in an efficient way. Where VGG16 is concerned, the architectural approach rests on small 3x3 convolution filters followed by ReLU (Rectified Linear Unit) for activation-introducing the 'non-linearity' effect in the model. This non-linearity may provide a better capability of learning complex patterns made of edges, shapes, and textures of solar panels. Max-pooling layers downsample the feature maps on the spatial dimensions so that higher-level features can be captured with reduced computational burden. It can very well portray the boundaries of solar panels if they are on a rooftop along with other things such as a chimney or window. VGG16 is ideal for segmenting and identifying solar arrays because it is very deep and able to make critical feature extraction from satellite images. It has a powerful geometrical pattern recognition and feature extraction that makes the accuracy in detection very high for solar panels on various rooftop configurations with different lighting conditions.

Aspect	**Details**
Model Depth	VGG16 consists of 16 layers (13 convolutional, 3 fully connected).
Filter Size	Uses uniform 3x3 filters for all convolutional layers.
Transfer Learning	Utilizes pre-trained ImageNet [23] weights to expedite training and improve accuracy.
Feature Extraction	Effective at capturing essential features like edges and textures crucial for identifying solar panels.
Computational Complexity	VGG16 has 138 million parameters, making it computationally expensive.
Training and Inference Time	Longer training and inference times due to the large number of parameters.
Memory Requirements	Requires significant memory resources, limiting its usage in resource-constrained environments.

Table 3.9: Summary of Advantages and Limitations of VGG16 in Solar Panel Detection

This table highlights both the advantages and challenges of using VGG16, particularly in solar panel detection. The depth of the architecture and its ability to capture fine-grained features make it highly effective for this task, but its computational demands and slower inference times are its downside. To mitigate these challenges, several optimization techniques are employed. Data augmentation is commonly used to enhance the dataset artificially by applying transformations such as flipping, rotation, and scal-

ing. These transformations expose the model to a wider variety of input conditions, improving its generalization ability and ensuring accurate detection of solar panels from different perspectives, orientations, and lighting conditions.

3.3.3 Logistic Regression

Logistic regression [18] is a generalized linear model that is used to solve problems related to binary classification. A given example is likely to belong to a particular class based on an estimation of its input features. The logistic function, also known as the sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

maps the result of a linear equation into a value between 0 and 1, which stands for the predicted probability [46]. The particular advantages of this model occur when the relationship between the independent variables and the dependent binary outcome, while not strictly linear, could be well approximated by a log-odds relationship. One of the strong sides of logistic regression is its interpretability: it is very easy to judge feature importance directly from the coefficients of the model, since they reflect how the log odds of the outcome change given a certain feature. This interpretation possibility, along with simplicity, makes logistic regression a preferred method in solar panel detection where a correct understanding of the influence of each separate feature may be critical. Also, in high-dimensional space, it does not overfit as much compared to any other complex model, whether a decision tree or a neural network. Regularization techniques, such as L1 (Lasso) and L2 (Ridge), can be applied to prevent overfitting by penalizing large values of the coefficients and adding to the model's robustness [107]. Logistic regression still finds a wide appeal in today's age of more complex models because of its capability to give reliable and decent predictions in most real-world, practical applications along with being computationally effective [107]. Further to binary classification, logistic regression can also be adapted for multiclass problems using one-vs-rest, or in a more general setting, the multinomial logistic regression, expanding the use possibilities to more complex classification tasks [46]. The aforementioned characteristics show it as a strong foundational tool within the machine learning toolkit.

3.3.4 YOLOv3

Three essential parts make up the state-of-the-art convolutional neural network (CNN) architecture of the YOLOv3 algorithm, which is executed by Ultralytics [97]. These parts are the backbone, neck, and head. For reliable multi-scale feature extraction, its backbone incorporates a Feature Pyramid Network (FPN) [67]. The head uses these features for accurate object detection and bounding box prediction after the neck processes them to improve representation [96]. With the help of this architecture, YOLOv3 demonstrates its resilience in object detection tasks by achieving a mean Average Precision (mAP) of 67.3% at an Intersection over Union (IoU) threshold of 0.5 [96]. Using the following formula, one can calculate the overlap between the ground truth and predicted

bounding boxes in terms of space:

$$IoU = \frac{\text{Area of Union}}{\text{Area of Intersection}}$$

Improved alignment between the predicted and actual object locations is indicated by higher IoU values, which are essential for precise object localization [27]. By averaging precision across various recall levels for every class in a dataset, Mean Average Precision (mAP) assesses the overall performance of a model [28]. It displays the model's accuracy and recall, which are critical for determining how well it performs in different situations. With 47.0 million parameters, the YOLOv3 model variant that is used, YOLOv3l, strikes a balance between efficiency and accuracy.

3.3.5 YOLOv5

YOLOv5 [121] makes significant improvements in the architecture of CNNs for the task of object detection. It is lightweight and efficient, building off the CSPDarknet to achieve the best of both worlds in speed and accuracy. This was done by fine-tuning a balance between several key architectural improvements that make the YOLOv5 architecture very suitable for real-time applications such as autonomous driving, surveillance, and robotics, where decisions are made in a split second.

One of the new novelties that YOLOv5 brings along with it are Cross-Stage Partial (CSP) networks. This brings partial connectivity to the model. Unlike in previous architecture where layers were all connected, YOLOv5 selectively connects them. It also increases during a training to optimize gradients and enhance information flow across the network [4]. This decreases computational overhead without sacrificing accuracy. Therefore, YOLOv5 can be more efficient for large-scale object detection tasks when compared to a fully connected network. The result is a model that is both computationally efficient and powerful for large-scale object detection tasks.

Another feature of the YOLOv5 architecture is the incorporation of the Feature Pyramid Network, FPN, to enhance this capability of object size variation detection through multi-scale feature extraction. This becomes important in real-world applications, where objects of interest can dramatically vary in size and detail. While the FPN itself is not new, having been part of the earlier YOLOv3 model, in this instance its implementation was fine-tuned to work in complete harmony with the CSP architecture. This allows the network to capture features both at a fine and coarse level with much better detection accuracy across complex scenes with multiple object types and sizes [67]. In that regard, it enhances the adaptability and effectiveness of YOLOv5 for various applications, ranging from small object detection to larger and more prominent targets. YOLOv5 has notable performance improvements compared to its predecessors. The performance for the model variation in use shows a mean Average Precision of 49.0% at an IoU threshold of 50-95, an indication of a strong detection. In addition, YOLOv5 has some very fast inference speeds-233.9 milliseconds on a CPU using ONNX, 1.86 milliseconds on an A100 TensorRT GPU [121]. The combination of speed and accuracy for YOLOv5 makes the detection algorithm really appealing for real-world applications because results can come out fast but with high accuracy.

Beyond the architectural changes, new training strategies advanced the performance of YOLOv5 even further. The following additional features were added: multiscale training, mixed precision training, and auto-anchor optimization. Multiscale training allows the model to change the size of input images dynamically during training, improving generalization across a wide range of object sizes and scales. Mixed precision training accelerates the learning process by allowing lower precision in certain operations while reducing memory usage, yet maintaining accuracy. Meanwhile, auto-anchor optimization fine-tunes the model's anchor boxes for the best fit against ground-truth objects, specifically when dealing with custom data. This is merely one of a few methods that allow YOLOv5 to maintain high performance across an extremely wide scope of applications and environments, therefore increasing its versatility and robustness. Other aspects where YOLOv5 does better, bounding box prediction adds up with more convolutional layers and uses a fixed-size, fixed-scale box estimation strategy to improve the accuracy of the position and size of the detected objects. Such accurate localization will make YOLOv5 handle diverse and challenging environments, starting from partially occluded and overlapped objects to difficult lighting conditions, while detecting objects accurately. The performance of YOLOv5 further illustrates its robustness and adaptiveness to various deployment conditions. Whether deployed on high-performance GPUs or on edge devices, which generally have low computational power, YOLOv5 gives the right balance of speed and accuracy. This scalable nature enables the algorithm to excel across a wide range of real-world scenarios, from high-demand applications such as autonomous driving to more resource-constrained environments where edge computing is necessary. Moreover, the robustness towards changes in environmental factors such as lighting, occlusions, and motion proves its feasibility in real-world applications. The conclusion would be that YOLOv5 is a great jump from the older versions of YOLO due to its increased efficiency, speed, and accuracy. Architectural changes such as adding CSP networks and, most importantly, the refined use of FPN contribute much to the extremely good performance of this model both in controlled and complicated environments. Rich training strategies and a strong design make YOLOv5 fit modern object detection tasks, providing reliability and precision where it is needed most.

Table 3.10 presents the architectural and methodological differences between YOLOv3 and YOLOv5. This comparison explains how each iteration of the YOLO model framework was changed to overcome the challenges in real-time object detection. This comparison of features brings out light on neural network design and improvements in performance metrics.

Table 3.10: Key Differences Between YOLOv3 and YOLOv5

YOLO version	**YOLOv3**	**YOLOv5**
Neck	No CSP structure	CSP-PAN (Path Aggregation Network)
Data Augmentation	Random flips, scaling, color space augmentations	Mosaic, Copy-Paste, Random Affine, MixUp, Albumentations
Bounding Box Loss	IoU Loss	Complete IoU (CIoU) Loss
Speed Optimization	Faster than RetinaNet and SSD but no specific speed optimizations	SPPF replaces SPP for faster processing
Multiscale Predictions	3 scales, using Feature Pyramid Networks (FPN)	3 scales, with CSP-PAN for improved multi-scale predictions
Grid Sensitivity Reduction	No grid sensitivity reduction	Bounding box prediction formula reduces grid sensitivity
Training Strategies	Traditional training	Hyperparameter evolution, mixed precision training, EMA, cosine LR scheduler
Parameter Count	47 million parameters	Varies by model size (small to large options for speed-accuracy trade-offs)
Inference Speed	51 ms on a Titan X (YOLOv3-608)	Faster than YOLOv3, notable speed gains with SPPF

3.3.6 Later versions of YOLO

From YOLOv6 to YOLOv10, the continuous development of the YOLO series has marked one milestone after another in real-time object detection. Each version came with a set of fresh changes in every respect: architecture, optimization and ability to make it appropriate for real-world-large-scale applications. Each of these YOLO versions has emerged from papers on advancement research.

YOLOv6 improved multi-scale feature propagation, enhancing the accuracy of object detection—specifically the detection of objects such as solar panels—across different resolutions [68]. This is ideal for applications in satellite and aerial imagery where there is a variation in resolution depending on distance and sensor quality. YOLOv6 also kept CSPNet for the improvement of the gradient flow. It introduces Mosaic data augmentation that enables the model to generalize much better through different images by combining randomly chosen images during training [4]. That would be useful in improving the performance of the model over a wide range of environmental conditions, such as light and terrain conditions common in imagery from solar panels.

YOLOv7 fine-tuned the detection accuracy and efficiency with the addition of Dynamic Label Assignment and Extended Efficient Layer Aggregation Networks, respectively [114]. This allowed the model to dynamically perform label assignments during training to improve accuracy in complicated situations, for example, in detecting sun panel installations in cluttered urban environments with multiple overlapping objects. This enhanced efficiency of performance enables YOLOv7 to allow real-time analysis for monitoring over a wide area using aerial or satellite imagery.

Indeed, another important achievement was brought about by the YOLOv8 [9] upon the incorporation of transformers in the respective architectures such that the model gained its capacity to capture long-range dependencies found in images. In other words, for the detection of solar panels, the model will easily manage larger-scale and wide-area imagery in a better manner to attain panel detection across vast solar farms or highly concentrated urban areas. YOLOv8 also introduced variants at different scales-small, medium, and large balance performance across a wide range of computational environments, from high-performance GPUs used for processing large satellite datasets to lower-power edge devices found in drones [53]. Sophisticated data augmentation techniques, such as CutMix and MixUp, were also introduced; this had the effect of allowing the model to generalize well to different conditions, including those due to time of day or weather patterns commonly present in the imagery of solar panels.

YOLOv9 [115] introduced self-supervised learning techniques that reduce the need for large labeled datasets; these are often hard and expensive to get in satellite and aerial imagery [53]. The fact that it is possible to develop decent models with less labeled data is relevant due to the challenge of creating such large labeled datasets to detect solar panels. YOLOv9 even further improved multi-scale feature fusion, thus becoming effective in spotting solar installations of small sizes in an urban or rural context. It is better suited for large-scale deployment into monitoring at a regional level with varying panel sizes and types. Increased accuracy and reduced reliance on labeled data are some of the key advantages for YOLOv9 in large-scale deployments.

The most recent version YOLOv10 [113] came up with an overall efficiency-accuracy-driven design strategy, which has holistically optimized both post-processing and model architecture to minimize computational redundancy and further improve accuracy. This will be useful in many end-to-end detection tasks, such as identifying solar panels from satellite images, for which both efficiency and accuracy are of the highest order. YOLOv10 replaces non-maximum suppression (NMS) with consistent dual assignments, which leads to a much faster inference without loss of accuracy. Its stronger architecture-large-kernel convolution combined with self-attention mechanisms-allows it to learn finer details, including distinguishing solar panels from similarly-shaped objects in complex scenes [113]. Due to the lightweight design, YOLOv10 inherently enables efficiency on inference for edge devices, including drones, and is super ideal for real-time monitoring applications.

3.4 Evaluation Metrics

This study assesses performance using multiple evaluation metrics grouped into categories: conventional metrics (Recall, Precision, F1 Score, Mean Average Precision (mAP) and Matthews Correlation Coefficient (MCC) and robustness metrics (Affinity and Diversity). These metrics together provide an analysis of the model's performance and robustness.

3.4.1 Conventional Metrics

True Positives, True Negatives, False Positives, and False Negatives

The foundation of many evaluation metrics in classification tasks involves the following four quantities, derived from the confusion matrix. The confusion matrix is a type of table used in the evaluation of performance for a classification model. It allows you to view the predictions of classes by the model against the actual classes:

- **True Positives (TP):** The number of correctly identified positive cases.
- **True Negatives (TN:** The number of correctly identified negative cases.
- **False Positives (FP):** The number of negative instances incorrectly identified as cases.
- **False Negatives (FN):** The number of positive instances incorrectly identified as negative.

These quantities are essential for the construction of the confusion matrix, which provides a summary of prediction results for a classification problem [90].

Recall

Recall [110], also known as Sensitivity or True Positive Rate, measures the ability of a model to correctly identify all relevant instances from specific class. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

This metric is critical in scenarios where missing a positive instance has significant consequences [21]. High recall is essential in fields such as medical diagnostics and security, where failing to detect positive instances can lead to severe repercussions.

Interpretation of Values

- **High Recall (close to 1):** Indicates that the model identifies most positive instances, minimizing false negatives. This is desirable in applications requiring comprehensive detection of positive instances.

- **Low Recall (close to 0):** Indicates that the model misses many positive instances, which is detrimental in critical applications where capturing all relevant instances is crucial.

Precision

Precision, or Positive Predictive Value, measures the accuracy of the positive predictions made by the model. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

This metric is crucial for understanding the model's performance and represents the cost of false positives is high [106]. High precision is vital in applications such as spam detection and financial fraud detection, where false positives can lead to unnecessary costs or actions.

Interpretation of Values

- **High Precision (close to 1):** Indicates that the model's positive predictions are mostly correct, minimizing false positives. This is desirable in applications where false positives can lead to unessential actions.

- **Low Precision (close to 0):** Indicates that the model makes many incorrect positive predictions, suggesting a significant number of false positives.

F1 Score

The F1 Score is the harmonic mean of Precision and Recall, providing a single metric that balances both concerns. It is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 Score [16] ranges from 0 to 1, where a higher score indicates a better balance between precision and recall. This metric is particularly useful when dealing with imbalanced datasets, where a high F1 score indicates that the model maintains a good balance between capturing positive and avoiding false positive instances.

Interpretation of Values

- **High F1 Score (close to 1):** Indicates a good balance between precision and recall.
- **Low F1 Score (close to 0):** Indicates poor performance in either precision, recall, or both.

Mean Average Precision (mAP)

Mean Average Precision [41] is used to evaluate object detection systems. It combines the precision-recall curve into a single value by computing the average precision (AP) across all recall levels for each class, then averaging these values:

$$\text{AP} = \sum_{k=1}^{n} (R_k - R_{k-1}) P_k$$

where R_k and P_k are the recall and precision at the k-th threshold. The mAP for C classes is defined as:

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^{C} \text{AP}_c$$

mAP ranges from 0 to 1, where a higher value indicates better overall performance [28]. This metric is crucial for evaluating models in tasks where multiple classes and varying thresholds need to be considered, such as object detection in images.

Interpretation of Values

- **High mAP (close to 1):** Indicates robust performance across different classes and recall thresholds.
- **Low mAP (close to 0):** Indicates poor performance, whereby the model has difficulty maintaining high precision and recall across different classes and thresholds.

Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) [15] is a comprehensive metric that accounts for all elements of the confusion matrix, providing a balanced measure even when class distributions are uneven. The MCC is defined as:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

The MCC is particularly valuable in binary classification problems with imbalanced class distributions, providing a more balanced evaluation contrary to accuracy, which can be misleading in such scenarios.

Interpretation of Values

- **High MCC (close to +1):** Indicates excellent predictive performance.
- **Low MCC (close to 0):** Indicates performance no better than random chance.
- **Negative MCC (close to -1):** Indicates significant inverse relationship between predictions and actual values.

3.4.2 Robustness Metrics

Affinity

Affinity [36] measures how much the results predicted agree with the expected results after applying a stochastic augmentation strategy. It characterize the impact of data augmentation on model performance by evaluating the shift in data distribution and its effect on the model's accuracy. Affinity is defined as:

$$\text{Affinity} = A(m, D'_{\text{val}}) - A(m, D_{\text{val}})$$

where $A(m, D)$ represents the accuracy of the model m on dataset D, and D'_{val} is the validation set after applying the augmentation strategy. High affinity indicates that the model's performance remains stable despite changes in the data distribution [1].

Interpretation of Values

- **High Affinity (close to 0):** Indicates robustness to changes in data distribution, with similar performance before and after augmentation. Zero affinity suggests that the model can generalize well to augmented data, maintaining its predictive capabilities even when the input data is affected by significant transformations.
- **Low Affinity (far from 0):** Indicates significant performance degradation due to data augmentation, suggesting a lack of robustness. Low affinity implies that the model is ultra-sensitive to shifts in distribution introduced by various applied augmentations, potentially overfitting to the original data distribution and failing to adapt to new variations.

Diversity

Diversity [36], in this regard, quantifies how much variability and complexity exists in augmented data generated from a given dataset. The metric measures how far-reaching are the changes applied to a dataset such that a model is exposed to a wide variation during training. High diversity forces the model to generalize better by not fitting specific patterns in the training data. One can then define and quantify diversity using a variety of metrics, including entropy or variance-based measures that capture the extent of the spread of the augmented data in the feature space.

One way to quantify diversity is by measuring the entropy of the augmented data distribution. Entropy provides a statistical measure of randomness, reflecting the variability within the augmented dataset. It is defined as:

$$\text{Diversity} = -\sum_{i=1}^{n} p_i \log(p_i)$$

where p_i represents the probability of occurrence of the i-th augmentation transformation. Higher entropy indicates greater diversity, suggesting that the augmented data covers a wider range of variations.

Interpretation of Values

- **High Diversity: 0.7-1.0** Indicates that the augmented data includes a wide range of variations, leading to better generalization and robustness of the model. High diversity is achieved through different augmentations that significantly alter the input data. These transformations ensure the model is exposed to a broad spectrum of potential real-world scenarios.
- **Low Diversity: 0.0 - 0.4** Indicates that the augmented data is very similar to the original data, offering limited new information to the model. This can result from slight adjustments like small changes in brightness, which do not provide extensive and valuable variations. Low diversity may limit the model's ability to generalize to unseen data.

3.4.3 Summary

These combined metrics provide a full evaluation of the model's performance. Recall and F1 Score give information about the balance between precision and recall; MAP gives a measure for the general performance of the detection; and MCC provides a robust measure accounting for all dimensions of prediction accuracy. Affinity and Diversity guarantee that there is reliability and variation in the predictions realized, hence supporting the strength of this approach.

4 Results

In this chapter, we outline the outcomes of our study in solar panel detection using different methods. The major goal was to establish the efficacy of existing approaches and make suggestions for possible enhancements. Initially, the established methods are tested using SolarFinder as the baseline, as it applies several techniques to detect solar panels, such as VGG, Random Forest, and Logistic Regression. However, these globally applicable methods are performing suboptimally in tests with different image types.

To address these limitations, more advanced models, specifically YOLOv3 and YOLOv5, were implemented and evaluated. The results indicate that both YOLO models significantly outperform the baseline methods proposed in other well-established approaches, with YOLOv5 yielding better results compared to YOLOv3. Based on these findings, further improvements are explored by applying model soups, particularly uniform soup and greedy soup, to enhance the detection accuracy and robustness of YOLOv5.

The detailed results of our evaluations, comparisons, and enhancements are presented in the following sections, providing a comprehensive evaluation and analysis of the performance and effectiveness of various methods.

4.1 Evaluation of baseline methods

Random Forest

The Random Forest model was assessed using two datasets: SolarFinder and FCDPA. The model's performance was evaluated across various data augmentations, such as adjusting contrast and brightness, vertical flipping, rotation, translation, and horizontal flipping of the image.

F1 scores trained on the SolarFinder dataset were disappointingly low. Specifically, Contrast augmentation scored an F1 of 0.07, while the original dataset without augmentation returned the highest score of 0.72. The other augmentations—Vertical Flip, Rotation, Horizontal Flip, Translation, Brightness, Channel Shift—turned in similar scores of between 0.25 and 0.31. Performance on the FCDPA dataset was equally poor, with F1 scores ranging from 0.07 to 0.26, showing that the Random Forest model performed generally poorly in recognizing solar panels independent of any data augmentations. Therefore, there was no need to assess other evaluation metrics because the overall performance was already poor for this use case.

Additionally, while SolarFinder's approach may have benefited from masking buildings using OpenStreetMap [80] data to filter out non-relevant regions, such a preprocessing step was not implemented. This likely contributed to the lower accuracy and overall poor performance of the Random Forest model in the experiments. Without the additional

spatial information provided by OpenStreetMap [79], the model struggled to distinguish between solar panels and other rooftop features, leading to high false positive and false negative rates.

The major issue with the Random Forest model in this case was the failure to generalize across different datasets. The mixed dataset containing a diversity of images of solar panels from multiple sources posed a huge challenge to the model. Clearly, it is reflected that there is a great performance gap between the two datasets, SolarFinder versus FCDPA. This lack of generalizability may mean that the model had to be too biased on the dataset in use by SolarFinder and failed to adapt to the variability present in our mixed dataset.

VGG Model

In this section, the performance of the VGG model is tested for the SolarFinder and FCDPA datasets, which are designed for binary classification—whether an image contains a solar panel PV or not. The datasets are balanced in terms of the number of images from both categories to avoid overrepresentation of any class.

Various augmentations were made on the data for testing, including contrast, flipping both vertically and horizontally, rotation, translation, changing brightness and channel shift. Main Metrics: The F1 score and Recall were the main metrics used for the evaluation indicators of the performance of the model in classifying imbalanced and binary datasets.

Results are presented showing F1 scores ranging from 0.08 to 0.14 for the VGG model performance on the SolarFinder dataset. While the original dataset scored 0.14, all other augmentations such as contrasting, flipping, rotating, and changing brightness returned F1 scores of less than 0.13. The implication is that VGG performed very poorly on this dataset, irrespective of the augmentation applied.

On the other hand, the results of the FCDPA dataset were higher, between 0.22 and 0.42, its original version and most augmented one reached an F1 score of 0.42. Considering SolarFinder, this is a high increase, but still really underlines the limits of the general applicability of VGG for finding solar panels in real conditions across datasets.

These performance gaps therefore indicate that the performance of VGG, although quite good in specific contexts such as SolarFinder, generalizes poorly on other datasets. For instance, it can be seen that Matthew's correlation coefficient for SolarFinder is 0.18, comparable to our results, overfitting to some dataset-specific characteristics and generalizing on new data. Overfitting in this case could be avoided with some solid preprocessing, for example, masking of non-relevant regions using OpenStreetMap [80] data was not part of the preprocessing pipeline.

Although the literature presented VGG as very powerful for SolarFinder, results showed that it is not flexible enough to be used globally for detecting solar panels without huge modifications. The following table represents some hyperparameters that have been used to train our VGG model:

Hyperparameter	Value
Input size	150 x 150 x 3
Batch size	32
Optimizer	RMSprop (lr = 2e-5)
Loss function	Binary Crossentropy
Epochs	50
Augmentation techniques	Rotation, Translation, Brightness, Channel Shift
Early stopping patience	20
Checkpoint	Best model saved

Table 4.1: Hyperparameters for training the VGG model.

The hyperparameters, combined with binary labels and balanced datasets, were aimed at achieving the most reliable results possible within the constraints of the datasets and augmentations. However, further refinement is required for more generalized solar panel detection across diverse datasets.

Logistic Regression

The Logistic Regression model was also assessed using the SolarFinder and FCDPA datasets. Its performance was evaluated under various data augmentations, including contrast adjustment, vertical and horizontal flipping, rotation, translation, brightness adjustment, and channel shifting.

The F1 scores for the Logistic Regression model on the SolarFinder dataset were generally low. On the original dataset, the model achieved a very low F1 score of 0.07. However, the application of different data augmentations such as contrast and brightness adjustments, significantly improved performance, raising the F1 scores to 0.70 and 0.72, respectively. Other augmentations, including vertical and horizontal flips, rotation, translation, and channel shifting, yielded more moderate improvements, with F1 scores ranging between 0.27 and 0.33. These results suggest that while logistic regression benefits from specific augmentations, it struggles to perform effectively without them. In contrast, the model's performance on the FCDPA dataset was consistently lower, with F1 scores ranging from 0.07 to 0.30 across all augmentations. This indicates that the logistic regression model is highly sensitive to dataset-specific characteristics and may not generalize well across different data sources.

Comparing these results to the ones reported by SolarFinder for a logical regression model, we see our implementation is far from comparable in performance. SolarFinder reports the value of Matthew's correlation coefficient was 0.15. As can be visibly seen from the benchmarked numbers given for these datasets, our results, and particularly those of the FCDPA dataset, fell short, meaning Logical Regression struggles with generalization when applied to mixed datasets.

The primary challenge with the Logistic Regression model in this context is its reliance on dataset-specific features and the effectiveness of certain augmentations leading to inaccurate results. The mixed dataset, incorporating images from various vendors and sources, represents significant difficulties for the model. This can be evidenced by the

variable and unstable performance across different augmentations and datasets. This variability suggests that while Logistic Regression can perform adequately under certain conditions, it lacks the robustness needed for diverse, real-world applications.

Another issue, representing the poor performance of Logistic Regression, is the possible lack of feature diversity in the training dataset. Unlike deep learning models like VGG, which automatically extract intricate features, Logistic Regression relies very much on the quality and relevance of input features. This might be what probably underlay the inconsistencies exhibited by the model in the different augmentations and datasets.

4.2 Evaluation of YOLOv3 & YOLOv5

This section provides an in-depth evaluation of YOLOv3 and YOLOv5 models for solar panel detection using diverse datasets and augmentation techniques. The performance of these models is assessed based on key metrics such as F1 Score, Recall, and Matthews Correlation Coefficient (MCC), highlighting their effectiveness compared to traditional models like SolarFinder, VGG, and Random Forest.

YOLOv3

In this regard, the YOLOv3 model is tested on a mixed dataset that incorporates multiple data sources, providing a holistic assessment of its performance about a wide array of data augmentations, as mentioned in the previous sections about Logistic Regression, VGG, and Random Forest.

To summarize the model performance, the F1 Score is evaluated averaging 0.64 by the YOLOv3 model over different data augmentation instances. This metric reflects a good balance between precision and recall in the detection process of solar panels. To compare it with the SolarFinder-specific models that recorded as low as 0.07 F1 score, this YOLOv3 model represented huge improvements. It indicates that YOLOv3 generalizes well across augmentations, making it quite effective for improving these limitations noticed in previous methods. On the other hand, the Recall score achieved by YOLOv3 was 0.60, thus very good in terms of correctly classifying the solar panels regardless of augmentation techniques applied. In comparison, Recall scores obtained using the models from SolarFinder were far lower and spanned between 0.08 to 0.72. This shows that, unlike SolarFinder, YOLOv3 can hold up its sensitivity and perform much better on the solar panel detection task even on these data variations, but also not optimal for this specific use-case.

MCC of YOLOv3 delivered an average MCC of 0.43, which shows balanced performance with good predictive accuracy. This score indicates that the model was able to tell between the positive and negative instances of solar panels, considering true positives and negatives. In comparison, SolarFinder models were observed to have lower MCC values, thus showing the superiority of YOLOv3 models in making more accurate predictions across the mixed dataset.

These results clearly show that the model of YOLOv3 outperforms all models specific to SolarFinder on all relevant performance metrics. More importantly, the F1 score,

Recall, and MCC values combined indicate that besides being more accurate, YOLOv3 is also more consistent in performance. This gain is important in practical applications where reliable detection of solar panels is required with diverse and challenging datasets.

Table 4.2: YOLOv3 Hyperparameters

Hyperparameter	**Default Value**
epochs	100
batch-size	16
imgsz	640
optimizer	`SGD`
learning-rate (lr0)	0.01
weight-decay	0.0005
momentum	0.937
warmup-epochs	3
warmup-momentum	0.8
warmup-bias-lr	0.1

YOLOv5

The YOLOv5 model has demonstrated strong performance in detecting solar panels, as shown by its comprehensive evaluation metrics. The model's performance, particularly in terms of accuracy, precision, recall, and mean Average Precision (mAP), indicates its suitability for this task.

The evaluation of the YOLOv5 model incorporates a range of important metrics that collectively provide a comprehensive understanding of its performance and can be seen in Table 4.3.

Table 4.3: YOLOv5 Results

Metric	**Value**	**Interpretation**
Accuracy	0.9168	The YOLOv5 model achieved an accuracy of 91.68% on the mixed dataset, indicating high overall prediction correctness.
mAP@0.5	0.7822	The mean Average Precision (mAP) at an IoU threshold of 0.5 reflects effective detection performance.
mAP@0.5:0.95	0.4426	The mAP averaged over IoU thresholds from 0.5 to 0.95 shows good performance across varying overlap criteria.
Matthews Correlation Coefficient (MCC)	0.8190	An MCC of 0.8190 indicates strong correlation between predicted and actual classifications.
Precision	0.9763	A precision of 97.63% reflects a low false positive rate and effective positive prediction accuracy.

Metric	Value	Interpretation
Recall (Sensitivity)	0.9020	With a recall of 90.20%, the model effectively identified most of the actual positive cases.
F1 Score	0.9377	The F1 Score of 0.9377 indicates a strong balance between precision and recall.

The initial models tested on the SolarFinder dataset (VGG, Logistic Regression and Random Forest) showed significantly lower performance compared to the YOLOv5 model.

VGG, Logistic Regression, and Random Forest models had accuracies around 7% to 13% on the SolarFinder dataset, indicating poor performance. In contrast, the YOLOv5 model's accuracy of 0.9168 and its high values in precision, recall, and MCC highlight its superior detection capabilities. The mAP@0.5 and mAP@0.5:0.95 scores further confirm the model's effectiveness in accurately detecting solar panels under various conditions.

The YOLOv5 model demonstrates robust performance in solar panel detection tasks. The model's high accuracy, precision, recall, F1 score, and MCC, together with high mAP scores, make it a reliable and effective tool for solar panel detection. This performance significantly surpasses the results obtained from traditional models, underscoring the advantages of using advanced object detection frameworks like YOLOv5 for such applications.

In this study, a total of 36 models were trained using a wide range of hyperparameters to optimize the performance of the YOLOv5 architecture for solar panel detection. The hyperparameters explored during the experiments are displayed in Table 4.4. These range from learning rates, gains of the loss function, image augmentation parameters to many others, which try to reach the best possible detection accuracy across diverse environments. We trained the models using the CUDA03 cluster provided by TU Graz, which contains 8x Tesla P100 GPUs that allow efficient large-scale training. In total, training took around three months, a substantial amount of time due to the computation-intensive nature of the process and the large number of experiments conducted.

The following table presents the range of hyperparameters that were varied during the training of the models. These hyperparameters were carefully selected and adjusted to optimize the performance of the YOLOv5 model. Each hyperparameter was explored within the given range to balance the trade-offs between model accuracy, convergence speed, and generalization capability.

Hyperparameter	**Range**
lr0 (Initial learning rate)	0.001 - 0.0001
lrf (Final learning rate factor)	0.1 - 0.2
momentum (SGD momentum/Adam beta1)	0.85 - 0.95
weight_decay (Optimizer weight decay)	1e-4 - 5e-4
warmup_epochs (Warmup epochs)	2.0 - 5.0
warmup_momentum (Initial momentum during warmup)	0.7 - 0.9
warmup_bias_lr (Initial bias learning rate during warmup)	0.05 - 0.15
box (Box loss gain)	0.02 - 0.1
cls (Class loss gain)	0.2 - 0.5
cls_pw (Class positive weight for BCELoss)	0.9 - 1.1
obj (Object loss gain)	0.5 - 0.9
obj_pw (Object positive weight for BCELoss)	0.9 - 1.1
iou_t (IoU training threshold)	0.15 - 0.25
anchor_t (Anchor multiple threshold)	3.5 - 4.5
fl_gamma (Focal loss gamma)	0.0 - 1.0
hsv_h (HSV-Hue augmentation)	0.01 - 0.02
hsv_s (HSV-Saturation augmentation)	0.6 - 0.8
hsv_v (HSV-Value augmentation)	0.3 - 0.5
degrees (Image rotation)	0.0 - 5.0
translate (Image translation)	0.05 - 0.2
scale (Image scale)	0.8 - 1.1
shear (Image shear)	0.0 - 5.0
perspective (Image perspective)	0.0 - 0.001
flipud (Image flip up-down probability)	0.3 - 0.6
fliplr (Image flip left-right probability)	0.3 - 0.6
mosaic (Image mosaic probability)	0.8 - 1.0
mixup (Image mixup probability)	0.0 - 0.2
copy_paste (Segment copy-paste probability)	0.0 - 0.2

Table 4.4: YOLOv5 hyperparameters used for the training

4.3 Outcomes of YOLOv5 Enhancement with Model Soups

In the process of enhancing the performance of our object detection models, the model soup technique has been explored. It involves combining multiple trained models to capitalize on their collective strengths. The primary goal of this work is to apply both the uniform and greedy model soup approaches to determine which would lead to the best results.

The enhancements achieved through the model soups are visible in Figure 4.1. This figure provides a detailed comparison of the accuracies for each model, having the results from the uniform and greedy model soups. The visual representation highlights the significant improvements realized by the greedy model soup.

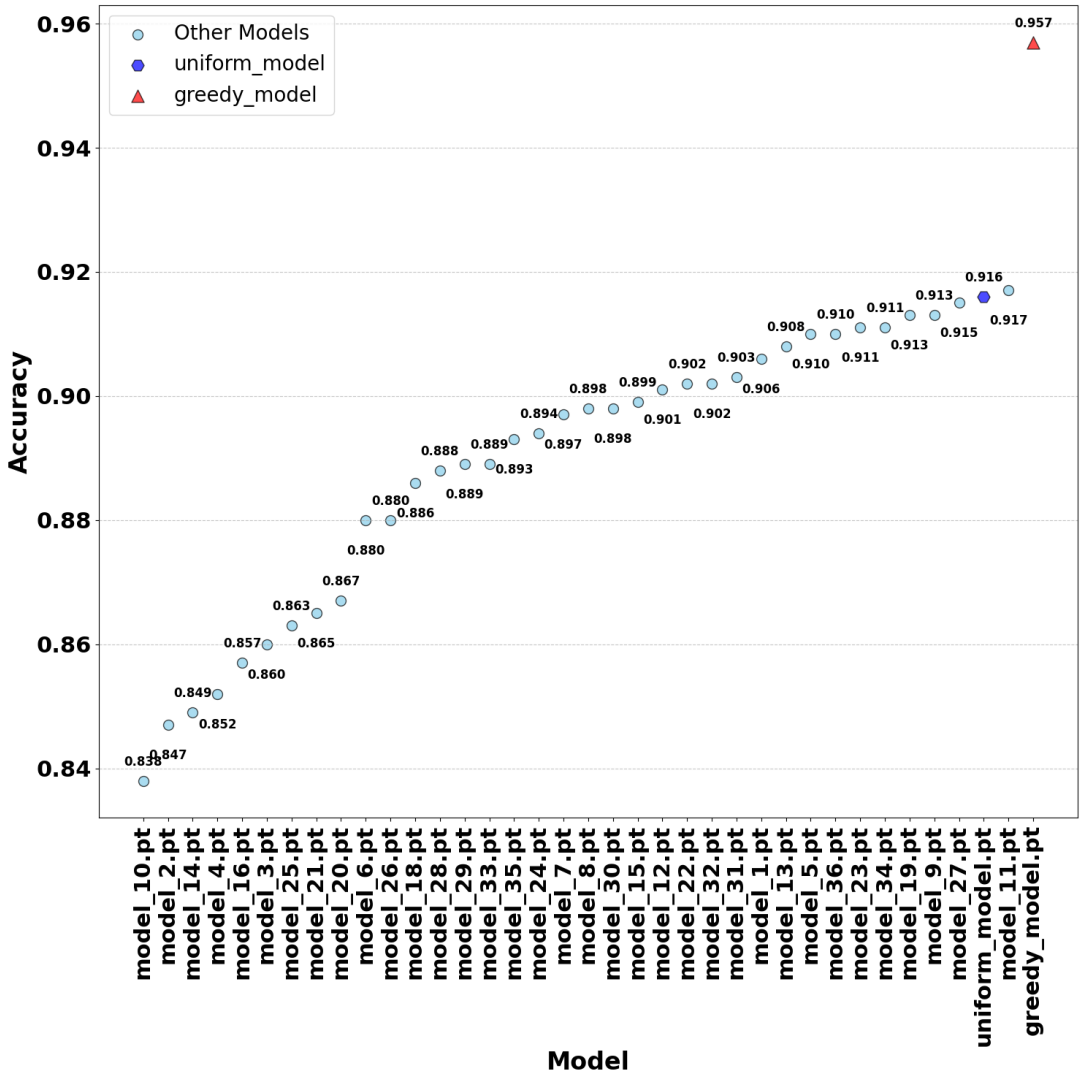

Figure 4.1: All models results

In the figure, it can be observed that the individual model accuracies vary, with model 11 achieving the highest accuracy of 0.9168. The uniform model soup, represented in Figure 4.1, shows a slight decrease in accuracy to 0.9158, suggesting that averaging the models did not result in a substantial improvement.

While the uniform model soup did provide some slight gain in stability, it still could not surpass the accuracy of the best of individual models. On the other hand, greedy model soup has rather impressive accuracy gains, establishing it as the superior solution for our purpose. Moving forward, based on these results, the greedy model soup will be used as a primary method. This solution also satisfies most of the performance objectives, hence providing a robust and reliable solution for our object detection tasks. The results in Figure 4.1 showcase the progress and advancements of the greedy model soup.

However, the greedy model soup stands out prominently in the figure, with an accuracy of 0.9569. This result underlines the effectiveness of selectively combining models based on their contributions, leading to a significant enhancement in overall performance. Detailed results for the greedy model, the best-performing model, are shown in Table 4.5.

Metric	**Value**
Accuracy	0.9569
Sensitivity	0.9580
Specificity	0.9544
Precision	0.9794
Recall	0.9580
F1 Score	0.9686
MCC	0.9005
mAP@0.5	0.8208
mAP@0.5:0.95	0.5139

Table 4.5: Performance metrics of the greedy model

These results confirm that a greedy model significantly outperforms the conventional methods, and more importantly, underline the important role of model selection and combination strategies in achieving superior performance for solar panel detection tasks. The high values across multiple metrics indicate the efficacy and reliability of the greedy model and hence make it a robust choice for practical applications.

The following figure illustrates the result of detection in a sample test image to showcase the efficiency of the greedy model in the detection task. In the figure, the bounding boxes of the model perfectly identify the solar panels, compromising their locations and classifying them as panels with high accuracy in diverse conditions. This visual representation underscores the high accuracy, precision, and robustness of the greedy model, as discussed in the evaluation metrics.

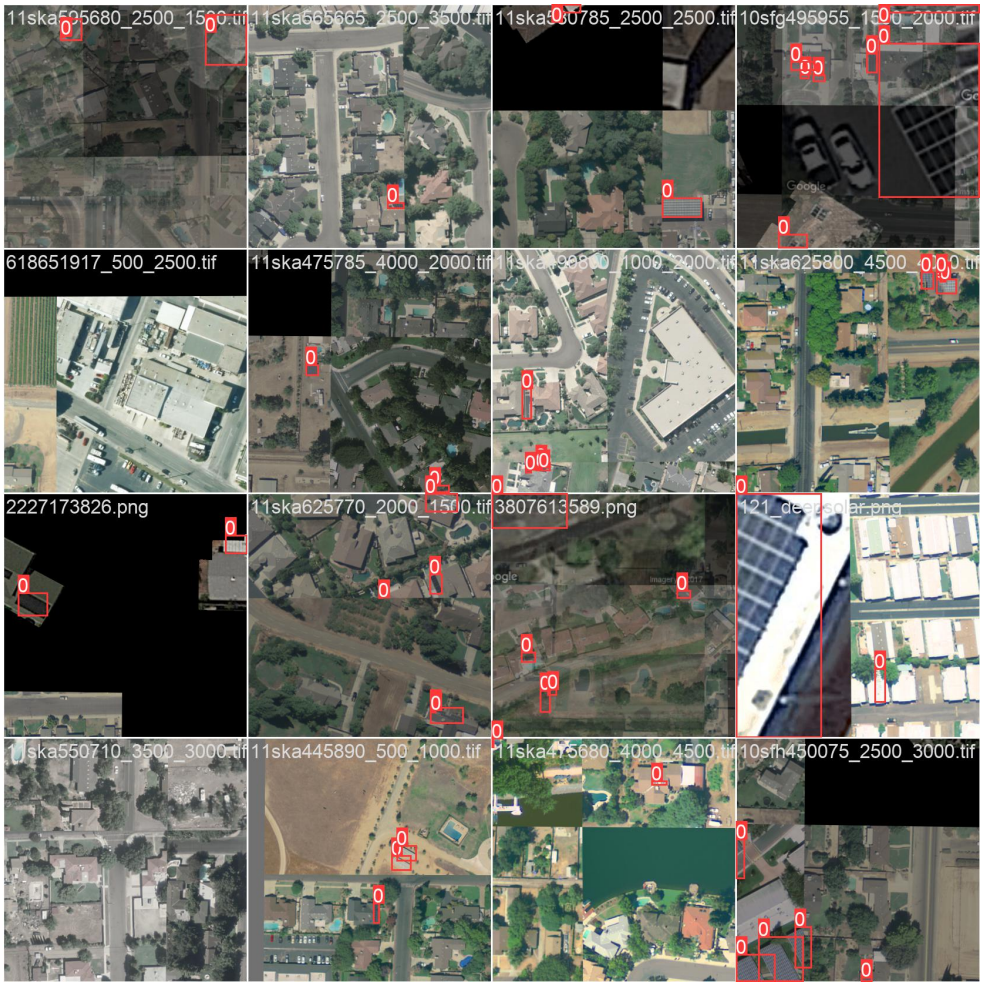

Figure 4.2: YOLOv5 results

4.4 Affinity and Diversity of the greedy model

The analysis demonstrates that the augmentation strategies achieve a harmonious balance between Diversity and Affinity within the greedy soup model.

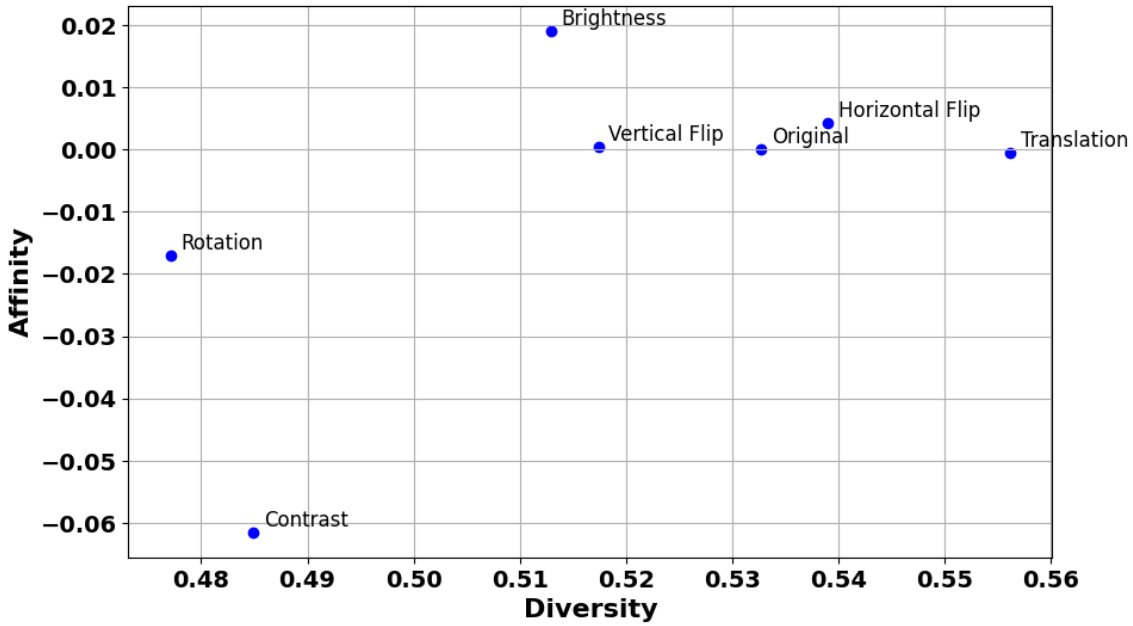

Figure 4.3: Affinity and diversity results

It can be seen that these augmentations are introducing a high level of variation into the dataset from the Diversity values. Translation has a Diversity value of 0.5562, thus most add variation to the dataset and are therefore likely to make the model more able in generalization by exposing it to more different ranges of data scenarios. In contrast, Rotation adds the least variation, with its Diversity value the lowest at 0.4771. Overall, the Diversity values suggest that these augmentations are effective at increasing the variety in the training data, which is generally beneficial for preventing overfitting.

The Affinity values at the start are close to zero and differ slightly among the various augmentations. The highest Affinity is that of brightness augmentation, with 0.0191, which means it is closest to the real data distribution. The lowest Affinity is Contrast with -0.0615, indicating that it adds the most out-of-distribution data. The overall Affinity across all datasets is very slightly negative (-0.0092), suggesting that the augmentations introduce a small amount of distribution shift. While this shift is generally minimal, it could potentially introduce noise if the Affinity is too low, particularly in the case of Contrast.

In summary, the augmentation strategies considered in this study appear to be well-balanced: there is sufficient diversity to improve model generalization without excessively compromising affinity. On the other hand, low-affinity augmentations like contrast may require some care since they can introduce harmful distribution shifts.

4.5 Number of panels in Vienna

Disclaimer: The number of panels computed in Vienna is based on the available data but lacks validation against ground-truth data. Therefore, the panel count should be interpreted with caution.

This subsection aims at analyzing the trend of the installation of solar panels in Vienna, Austria, from 2015 to 2021. We extract the areal images for this time frame and count the number of solar panels using the best-performing greedy model. By using this strategy, useful information about the trend and adoption of solar energy in Vienna is being gathered. This reflects the progress of the city toward sustainable energy solutions. The following analysis gives the count of solar panels detected in each year, hence clearly showing the growth trajectory and outlining significant changes or patterns in the installation of solar panels across Vienna over the period.

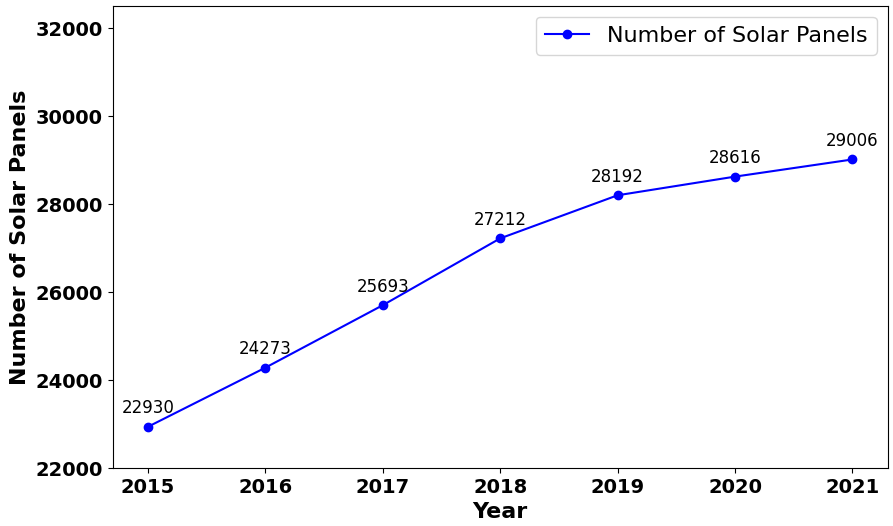

Figure 4.4: Number of solar panels in Vienna

The trend line, as indicated in Figure, shows a steady increase in the solar panels installed in Vienna from the year 2015 to 2021. This is an upwards trajectory that signifies the commitment of Vienna to sustainable energy solutions and the growing adoption of solar technology. The number of installed solar panels increases constantly from 2015 to 2019. This growth can be attributed to favorable government policies, incentives for renewable energy adoption, and increased public awareness of the benefits of solar energy. Installation numbers grew from 22,930 in the year 2015 to 28,192 in 2019, showing an average annual growth rate of about 5.6%. The COVID-19 pandemic brought huge challenges to the construction industry worldwide in the year 2020 [98]; Austria was no exception [99]. These have caused supply chain disruptions, labor shortages, and delays in projects connected with the installation of solar panels and, as such, impacted

construction activities. Despite all odds, Vienna increased from 28,192 to 29,416 panels from the year 2019 to 2020. This smaller growth rate, compared to the previous years, gives a reflection of the pandemic but shows the resilience from ongoing projects and commitment to renewable energy by the city.

Acknowledging that the figures presented are derived from an automated detection model rather than official statistics is essential. While the data obtained provides a very good indication of the trend in growth, it is not an indication of the exact number of installations, as inaccuracies in the model and limitations to detection may take place.

5 Conclusion and Outlook

5.1 Conclusion

This thesis presents the application of the deep learning architecture YOLOv5 for detecting solar panels from satellite images, motivated by the growing interest in monitoring renewable energy. Traditional detection methods, on the other hand, which rely on handcrafted features and classical machine learning, often lack generalization ability or accuracy when applied to complex and diverse geographical areas. The YOLOv5-based model in this study had an F1 score of 0.93 and an MCC of 0.81, proving its robustness and high precision under various conditions. These metrics are indicative of the actual performance that one might get from the model while detecting solar panels. Finally, the case study on the growth of solar panels in Vienna from 2015 to 2021 further demonstrated the model's effectiveness by revealing distinct trend lines in solar panel adoption. Lacking official data to confirm such findings constitutes a limitation and a need for more complete and accessible datasets in order to facilitate validation. Therefore, this study has demonstrated how deep learning models, particularly YOLOv5, have great potential to expand the scope of solar panel detection from satellite imagery. If optimized properly, such models can ensure the reliable data required for renewable energy infrastructure expansion. The contribution of this work, therefore, lies in demonstrating that deep learning holds significant potential and can substantially contribute to both geospatial analysis and environmental planning.

5.2 Outlook

While this work has achieved promising results with the YOLOv5 architecture for the detection of photovoltaic panels, there are several aspects that can be enhanced and developed further. Indeed, further research could be conducted from any of the following standpoints:

Robustness Across Ecologically Variable Settings: Among the primary weaknesses of the model, when implemented in the scope of this thesis, was that regarding performance where environmental conditions are not like those represented in the dataset from which the models were derived. Future works should be directed toward the extension of this dataset to a much wider range of environmental views, such as tropical, polar, and arid regions. The incorporation of images across seasons takes into consideration factors such as snow cover, foliage, or cloud cover, thus ensuring that this model performs well throughout the year. This would, in turn, make the model much more generalizable into new and unseen environments, reducing the risk of performance degradation when

used on global-scale detection.

Transfer Learning and Fine-Tuning: Given how promising the results with YOLOv5 were, leveraging this benefit, transfer learning became a good approach to better the performance for specific contexts. Transfer learning allows the fine-tuning of a pre-trained model using regionspecific data, which improves its accuracy for localized applications. It also minimizes the time and computation resources required for retraining. This will equally reduce overfitting by freezing early model layers and allowing later layers to be tuned in to capture local features that could improve both accuracy and generalization.

Multimodal Data Integration: Other than satellite images, there is a scope of integrating multispectral and LiDAR data among other sources, which can increase the accuracy in object detection. Multispectral images have been used to distinguish solar panel areas from other similar-looking objects due to the fact that different spectra highlight features not usually evident in common optical images. LiDAR data, on the other hand, incorporate height and spatial orientation information; hence, such data would be more appropriately used for recognizing rooftop solar panels, as it could identify objects by their level. The combination of these data modalities can bring significant reductions in false positives, especially for dense urban settings or regions with structurally similar looking objects.

Temporal Analysis: This study will logically be extended to conduct a temporal analysis in order to track the growth of solar panels. Timeseries satellite image analysis will allow researchers to quantify the rate at which solar panel installations occur in certain regions and give useful patterns in the adoption of renewable energies. A temporal perspective may also be used to help policymakers and city planners in assessing how well solar energy initiatives have fared with time.

In summary, this study has provided a high-level foundation in the domain of detecting solar panels using YOLOv5. Enhancing generalizability, transfer learning to enhance the model, incorporation of multimodal data, and understanding of temporal dynamics are all avenues where future work needs to be concentrated for wider applications of renewable energy monitoring and geospatial analytics.

5.3 Final Thoughts

This work is a significant contribution to the progress in the detection of solar panels from satellite images. This is a research that unlocks the power of deep learning, more precisely the YOLOv5 architecture and model soups for highly accurate and reliable detection in diverse settings. The findings will not only add to the present knowledge base on the distribution and growth of solar panels but also probably result in further research in this area. Subjected to further research and development, techniques and insights proposed in this work could very well become a part of transformative advancements in the monitoring of renewable energy resources and eco-conscious decision-making. The integration of advanced technologies into practical applications is in its evolution stage; however, this work signifies a step toward a sustainable data-driven future.

Bibliography

[1] Markus Augustin and S. W. U. Saeedi. "Adaptive Data Augmentation for Robust Detection Systems". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.11 (2020), pp. 4521–4534.

[2] Jonathan T. Barron. "A general and adaptive robust loss function". In: *CVPR* (2019), pp. 4331–4339.

[3] Roni Blushtein-Livnon, Tal Svoray, and Michael Dorman. "Performance of Human Annotators in Object Detection and Segmentation of Remotely Sensed Data". In: *Papers With Code* (2024). URL: https://paperswithcode.com/paper/performance-of-human-annotators-in-object.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal speed and accuracy of object detection". In: *arXiv preprint arXiv:2004.10934* (2020).

[5] Kyle Bradbury et al. *Distributed Solar Photovoltaic Array Location and Extent Data Set for Remote Sensing Object Identification*. 2018. DOI: 10.6084/m9.figshare.3385780. URL: https://figshare.com/articles/dataset/Distributed_Solar_Photovoltaic_Array_Location_and_Extent_Data_Set_for_Remote_Sensing_Object_Identification/3385780/4.

[6] Leo Breiman. "Bagging predictors". In: *Machine Learning* 24.2 (1996), pp. 123–140.

[7] Leo Breiman. "Random Forests". English. In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. URL: http://dx.doi.org/10.1023/A%3A1010933404324.

[8] Peter Bühlmann. "Bagging, Boosting and Ensemble Methods". In: *Handbook of Computational Statistics* (Jan. 2012). DOI: 10.1007/978-3-642-21551-3_33.

[9] Nicolas Carion et al. "End-to-End Object Detection with Transformers". In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2020), pp. 213–229.

[10] Nicholas Carlini and David Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *IEEE Symposium on Security and Privacy* (2017), pp. 39–57.

[11] Fabio Maria Carlucci et al. *AutoDIAL: Automatic DomaIn Alignment Layers*. 2017. arXiv: 1704.08082 [cs.CV]. URL: https://arxiv.org/abs/1704.08082.

[12] Yair Carmon et al. *Unlabeled Data Improves Adversarial Robustness*. 2019. arXiv: 1905.13736 [stat.ML].

[13] Ángela Casado-García and Jónathan Heras. "Ensemble methods for object detection". In: *arXiv preprint arXiv:2001.00910* (2020).

[14] Arjun Chandra and Xin Yao. "Evolving hybrid ensembles of learning machines for better generalisation". In: *Neurocomputing* 69.7 (2006). New Issues in Neurocomputing: 13th European Symposium on Artificial Neural Networks, pp. 686–700. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2005.12.014. URL: https://www.sciencedirect.com/science/article/pii/S0925231205003188.

[15] Davide Chicco, Matthijs J. Warrens, and Giuseppe Jurman. "The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment". In: *IEEE Access* 9 (2021), pp. 78368–78381. DOI: 10.1109/ACCESS.2021.3084050.

[16] Nancy Chinchor. *MUC-4 Evaluation Metrics*. Association for Computational Linguistics, 1992.

[17] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[18] David R Cox. "The regression analysis of binary sequences". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2 (1958), pp. 215–232.

[19] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. 1st ed. Cambridge University Press, 2000. ISBN: 0521780195. URL: http://www.amazon.com/Introduction-Support-Machines-Kernel-based-Learning/dp/0521780195/ref=sr_1_1?ie=UTF8&s=books&qid=1280243230&sr=8-1.

[20] Ekin Dogus Cubuk et al. "Tradeoffs in Data Augmentation: An Empirical Study". In: *ICLR*. 2021. URL: https://openreview.net/forum?id=ZcKPWuhG6wy.

[21] Jesse Davis and Mark Goadrich. "The Relationship Between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning (ICML)*. ACM. 2006, pp. 233–240.

[22] "DeepSolar Dataset". In: vol. 1. [Online; accessed 08-September-2021]. 2016. URL: https://academictorrents.com/collection/deepsolar-dataset.

[23] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[24] *Detection and Mapping of Photovoltaic Panels using ArcGIS and Deep Learning*. 2020. URL: /mnt/data/Detection-and-Mapping-of-Photovoltaic-Panels.pdf.

[25] Terrance DeVries and Graham W. Taylor. *Improved Regularization of Convolutional Neural Networks with Cutout*. 2017. arXiv: 1708.04552 [cs.CV]. URL: https://arxiv.org/abs/1708.04552.

[26] Rahim Entezari. "Optimization and Generalization of Neural Networks at the Edge". Doctoral Thesis. Graz, Austria: Graz University of Technology, Apr. 2023.

[27] Mark Everingham et al. “PASCAL Visual Object Classes Challenge 2007 (VOC2007)”. In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338.

[28] Mark Everingham et al. “The PASCAL Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (2015), pp. 303–338.

[29] Wei Fan and Kun Zhang. “Bagging”. In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 206–210. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_567.

[30] Yoav Freund and Robert E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[31] Jerome H. Friedman. “Greedy function approximation: A gradient boosting machine”. In: *Annals of Statistics* 29.5 (2001), pp. 1189–1232.

[32] Angus Galloway, Thomas Tanay, and Graham W. Taylor. *Adversarial Training Versus Weight Decay*. 2018. arXiv: 1804.03308 [cs.LG].

[33] Yaroslav Ganin et al. “Domain-Adversarial Training of Neural Networks”. In: *Journal of Machine Learning Research* 17.59 (2016), pp. 1–35. URL: http://jmlr.org/papers/v17/15-239.html.

[34] Aritra Ghosh, Himanshu Kumar, and P.S. Sastry. “Robust Loss Functions under Label Noise for Deep Neural Networks”. In: *AAAI* (2017), pp. 1919–1925.

[35] Supper & Supper GmbH. *Detection and Mapping of Photovoltaic Panels using ArcGIS and Deep Learning*. https://supperundsupper.com/wp-content/uploads/2019/10/Detection-and-Mapping-of-Photovoltaic-Panels.pdf. Project Report. 2024.

[36] Raphael Gontijo-Lopes et al. *Affinity and Diversity: Quantifying Mechanisms of Data Augmentation*. 2020. arXiv: 2002.08973 [cs.LG]. URL: https://arxiv.org/abs/2002.08973.

[37] Alvaro Gonzalez-Jimenez et al. “Robust T-Loss for Medical Image Segmentation”. In: *arXiv preprint arXiv:2103.00243* (2023).

[38] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[39] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2015).

[40] J. A. Hartigan and M. A. Wong. “A k-means clustering algorithm”. In: *JSTOR: Applied Statistics* 28.1 (1979), pp. 100–108.

[41] Paul Henderson and Vittorio Ferrari. *End-to-end training of object class detectors for mean average precision*. 2017. arXiv: 1607.03476 [cs.CV]. URL: https://arxiv.org/abs/1607.03476.

[42] Dan Hendrycks and Thomas Dietterich. *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*. 2019. arXiv: 1903.12261 [cs.LG].

[43] Dan Hendrycks et al. *AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty*. 2020. arXiv: 1912.02781 [stat.ML]. URL: https://arxiv.org/abs/1912.02781.

[44] Dan Hendrycks et al. "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty". In: *arXiv preprint arXiv:2305.14165* (2023).

[45] Judy Hoffman et al. "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". In: Proceedings of Machine Learning Research 80 (2018). Ed. by Jennifer Dy and Andreas Krause, pp. 1989–1998. URL: https://proceedings.mlr.press/v80/hoffman18a.html.

[46] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.

[47] Xin Hou et al. *SolarNet: A Deep Learning Framework to Map Solar Power Plants In China From Satellite Imagery*. 2019. arXiv: 1912.03685 [cs.CV].

[48] Peter J. Huber. "Robust Estimation of a Location Parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[49] Chaonan Ji et al. "Solar Photovoltaic Module Detection Using Laboratory and Airborne Imaging Spectroscopy Data". In: *Remote Sensing of Environment* 266 (2021), p. 112692. ISSN: 0034-4257. DOI: 10.1016/j.rse.2021.112692.

[50] Yan Jia, Tianyu Zhang, and James Zou. "On the Certified Robustness for Ensemble Models and Beyond". In: *arXiv preprint arXiv:2301.04564* (2023).

[51] H. Jiang et al. "Multi-resolution dataset for photovoltaic panel segmentation from satellite and aerial imagery". In: *Earth Syst. Sci. Data* 13 (2021), pp. 5389–5401. DOI: 10.5194/essd-13-5389-2021.

[52] Shihao Jiang, Richard Hartley, and Basura Fernando. "Kernel Support Vector Machines and Convolutional Neural Networks". In: *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE. 2018, pp. 1–8. DOI: 10.1109/DICTA.2018.8615840.

[53] Glenn Jocher et al. *YOLOv8: State-of-the-art object detection and image segmentation with YOLOv8*. 2023. URL: https://github.com/ultralytics/yolov8.

[54] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[55] Jules Raymond Kala, Serestina Viriri, and Jules Raymond Tapamo. "An approximation based algorithm for minimum bounding rectangle computation". In: *2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST)*. 2014, pp. 1–6. DOI: 10.1109/ICASTECH.2014.7068101.

[56] Nitish Shirish Keskar et al. "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836* (2017). URL: https://arxiv.org/abs/1609.04836.

[57] Chanyul Kim, Jaesun Park, and Yoonsik Kim. "Enhancing image classification using data augmentation: A practical approach". In: *Journal of Imaging Science and Technology* 62.6 (2018), pp. 060406–1–060406–9.

[58] Albert H.R. Ko, Robert Sabourin, and Alceu Souza Britto, Jr. “From dynamic classifier selection to dynamic ensemble selection”. In: *Pattern Recognition* 41.5 (2008), pp. 1718–1731. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2007.10.015. URL: https://www.sciencedirect.com/science/article/pii/S0031320307004499.

[59] Abhishek Kumar et al. *Co-regularized Alignment for Unsupervised Domain Adaptation*. 2018. arXiv: 1811.05443 [cs.LG]. URL: https://arxiv.org/abs/1811.05443.

[60] Hugo Larochelle et al., eds. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020. URL: https://proceedings.neurips.cc/paper/2020.

[61] Jiaqi Li et al. “Deep learning based defect detection algorithm for solar panels”. In: *2023 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*. IEEE. 2023, pp. 438–443. DOI: 10.1109/WRCSARA60131.2023.10261859. URL: https://discovery.dundee.ac.uk/en/publications/5122102d-d123-43b5-a5f9-dc18fb11f55c.

[62] Linyi Li, Tao Xie, and Bo Li. *SoK: Certified Robustness for Deep Neural Networks*. 2023. arXiv: 2009.04131 [cs.LG]. URL: https://arxiv.org/abs/2009.04131.

[63] Qi Li et al. *SolarFinder: Automatic Detection of Solar Photovoltaic Arrays*. 2020.

[64] Wendi Li et al. “Robust High Dimensional Image Data Generation via Disentangled Feature Representations”. In: *arXiv preprint arXiv:1912.05085* (2019).

[65] Xia Li et al. *Expectation-Maximization Attention Networks for Semantic Segmentation*. 2019. arXiv: 1907.13426 [cs.CV].

[66] Yanghao Li et al. “Adaptive Batch Normalization for practical domain adaptation”. In: *Pattern Recognition* 80 (2018), pp. 109–117. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2018.03.005. URL: https://www.sciencedirect.com/science/article/pii/S003132031830092X.

[67] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 936–944.

[68] Shu Liu et al. “Path aggregation network for instance segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[69] Jonathan Long, Evan Shelhamer, and Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV]. URL: https://arxiv.org/abs/1411.4038.

[70] Mingsheng Long et al. “Transfer Feature Learning with Joint Distribution Adaptation”. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2200–2207. DOI: 10.1109/ICCV.2013.274.

[71] José Antonio Luceño-Sánchez, Ana María Díez-Pascual, and Rafael Peña Capilla. "Materials for Photovoltaics: State of Art and Recent Developments". In: *Int. J. Mol. Sci.* 20.4 (2019), p. 976. DOI: 10.3390/ijms20040976.

[72] Linhai Ma and Liang Liang. "Adaptive Adversarial Training to Improve Adversarial Robustness of DNNs for Medical Image Segmentation and Detection". In: *arXiv preprint arXiv:2203.12709* (2022).

[73] Xingjun Ma, Bo Liu, and Dacheng Tao. "Normalized Loss Functions for Deep Learning with Noisy Labels". In: *ICML* (2020), pp. 6543–6553.

[74] Aleksander Madry et al. "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083* (2018).

[75] Jordan M. Malof, Leslie M. Collins, and Kyle 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.

[76] Ammar Mohammed and Rania Kora. "A comprehensive review on ensemble deep learning: Opportunities and challenges". In: *Journal of King Saud University - Computer and Information Sciences* 35.2 (2023), pp. 757–774. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2023.01.014. URL: https://www.sciencedirect.com/science/article/pii/S1319157823000228.

[77] Global Footprint Network. *Earth Overshoot Day 2022*. https://www.overshootday.org/content/uploads/2022/06/2022_Past_EOD_en.pdf. 2022.

[78] Rising Odegua. *An Empirical Study of Ensemble Techniques (Bagging, Boosting and Stacking)*. 2019.

[79] "OpenStreetMap API". In: vol. 1. [Online; accessed 17-November-2021]. 2021. URL: https://overpass-turbo.eu/.

[80] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. 2017.

[81] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: 1511.08458 [cs.NE]. URL: https://arxiv.org/abs/1511.08458.

[82] I. N. Otosaka et al. "Mass balance of the Greenland and Antarctic ice sheets from 1992 to 2020". In: *Earth System Science Data* 15.4 (2023), pp. 1597–1616. DOI: 10.5194/essd-15-1597-2023. URL: https://essd.copernicus.org/articles/15/1597/2023/.

[83] Victor M. Panaretos and Yoav Zemel. "Statistical Aspects of Wasserstein Distances". In: *Annual Review of Statistics and Its Application* 6.1 (Mar. 2019), 405–431. ISSN: 2326-831X. DOI: 10.1146/annurev-statistics-030718-104938. URL: http://dx.doi.org/10.1146/annurev-statistics-030718-104938.

[84] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. “Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples”. In: *arXiv preprint arXiv:1602.02697* (2016).

[85] Nicolas Papernot et al. “Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks”. In: *IEEE Symposium on Security and Privacy* (2016), pp. 582–597.

[86] Nicolas Papernot et al. “The limitations of deep learning in adversarial settings”. In: *IEEE European Symposium on Security and Privacy* (2016), pp. 372–387.

[87] Poonam Parhar et al. “HyperionSolarNet: Solar Panel Detection from Aerial Images”. In: *Proceedings of the 2022 AAAI Conference on Artificial Intelligence*. AAAI. 2022, pp. 1–7. URL: https://paperswithcode.com/paper/hyperionsolarnet-solar-panel-detection-from.

[88] Poonam Parhar et al. *HyperionSolarNet: Solar Panel Detection from Aerial Images*. Jan. 2022. DOI: 10.48550/arXiv.2201.02107.

[89] Diganta Kumar Pathak et al. “Differentially Private Image Classification Using Support Vector Machine and Differential Privacy”. In: *MDPI Sensors* 22 (2022), pp. 1–15. DOI: 10.3390/s22041534.

[90] David Martin Powers. “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation”. In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63.

[91] Steven Puttemans, Wiebe van Ranst, and Toon Goedeme. “Detection of photovoltaic installations in RGB aerial imaging: a comparative study”. In: 2016. URL: http://proceedings.utwente.nl/429/.

[92] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[93] Asif Raihan. “A Comprehensive Review of the Recent Advancement in Integrating Deep Learning with Geographic Information Systems”. In: *Research Briefs on Information and Communication Technology Evolution* 9 (Oct. 2023), p. 115. DOI: 10.56801/rebicte.v9i.160.

[94] Sylvestre-Alvise Rebuffi et al. “Data Augmentation Can Improve Robustness”. In: *arXiv preprint arXiv:2111.05328* (2021).

[95] CA: Environmental Systems Research Institute Redlands. *ArcGIS Desktop: Release 10*. 2011.

[96] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *arXiv preprint arXiv:1804.02767* (2018). URL: https://arxiv.org/abs/1804.02767.

[97] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767.

[98] RESEARCH and MARKETS. “Construction in Austria - Key Trends and Opportunities to 2025”. In: *RESEARCH AND MARKETS* (2021). URL: \url{https://www.researchandmarkets.com/reports/5275113/construction-in-austria-key-trends-and#}.

[99] RESEARCH and MARKETS. “COVID-19 Impact on Construction in Austria (Update 3)”. In: *RESEARCH AND MARKETS* (2020). URL: \url{https://www.researchandmarkets.com/reports/5125230/covid-19-impact-on-construction-in-austria}.

[100] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

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

[102] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. “Asymmetric Tri-training for Unsupervised Domain Adaptation”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 2988–2997. URL: https://proceedings.mlr.press/v70/saito17a.html.

[103] Anna C. Schapiro and Yael Niv. “The role of training variability for model-based and model-free learning of an arbitrary visuomotor mapping”. In: *PLOS Computational Biology* 19.4 (2023), pp. 1–19. DOI: 10.1371/journal.pcbi.1012471. URL: https://doi.org/10.1371/journal.pcbi.1012471.

[104] Jian Shen et al. “Wasserstein Distance Guided Representation Learning for Domain Adaptation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (2018). DOI: 10.1609/aaai.v32i1.11784. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11784.

[105] Lu Shi and Yue feng Zhao. “Urban feature shadow extraction based on high-resolution satellite remote sensing images”. In: *Alexandria Engineering Journal* 77 (2023), pp. 443–460. ISSN: 1110-0168. DOI: https://doi.org/10.1016/j.aej.2023.06.046. URL: https://www.sciencedirect.com/science/article/pii/S1110016823005112.

[106] Marina Sokolova and Guy Lapalme. “A Systematic Analysis of Performance Measures for Classification Tasks”. In: *Information Processing & Management* 45.4 (2009), pp. 427–437.

[107] Xudong Sun and Jia Huang. “A study on class imbalance problem in data mining”. In: *Data Mining and Knowledge Discovery* 34.6 (2020), pp. 1635–1659.

[108] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[109] Rohan Taori et al. *Measuring Robustness to Natural Distribution Shifts in Image Classification*. 2020. arXiv: 2007.00644 [cs.LG].

[110] Kai Ming Ting. "Precision and Recall". In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer, 2011, pp. 781–781. DOI: 10.1007/978-0-387-30164-8_652.

[111] Florian Tramer et al. "Ensemble Adversarial Training: Attacks and Defenses". In: *arXiv preprint arXiv:1705.07204* (2018).

[112] Jhon Jairo Vega Díaz et al. "Solar Panel Detection within Complex Backgrounds Using Thermal Images Acquired by UAVs". In: *Sensors* 20.21 (2020). ISSN: 1424-8220. DOI: 10.3390/s20216219. URL: https://www.mdpi.com/1424-8220/20/21/6219.

[113] Ao Wang et al. "YOLOv10: Real-Time End-to-End Object Detection". In: *arXiv preprint arXiv:2405.14458v1* (2024). URL: https://github.com/THU-MIG/yolov10.

[114] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors". In: *arXiv preprint arXiv:2207.02696*. 2022.

[115] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*. 2024. arXiv: 2402.13616 [cs.CV]. URL: https://arxiv.org/abs/2402.13616.

[116] Jiamin Wang, Kanglei Wang, and Lei Zhang. "Improved Small Object Detection Algorithm CRL-YOLOv5". In: *Sensors* 24.19 (2024), p. 6437. DOI: 10.3390/s24196437.

[117] David H. Wolpert. "Stacked generalization". In: *Neural Networks* 5.2 (1992), pp. 241–259.

[118] Sanghyun Woo et al. *CBAM: Convolutional Block Attention Module*. 2018. arXiv: 1807.06521 [cs.CV]. URL: https://arxiv.org/abs/1807.06521.

[119] Mitchell Wortsman et al. *Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time*. 2022. arXiv: 2203.05482 [cs.LG].

[120] Lewis Wyatt. "Using ensemble methods to improve the robustness of deep learning for image-based ecological data". In: *Methods in Ecology and Evolution* 13.3 (2022), pp. 465–478.

[121] "YOLOv5 Github Repository". In: vol. 1. [Online; accessed 08-November-2021]. 2020. URL: https://github.com/ultralytics/yolov5.

[122] Jiafan Yu et al. "DeepSolar: A Machine Learning Framework to Efficiently Construct a Solar Deployment Database in the United States". In: *Joule* 2 (2018), 2605–2617.

[123] Sangdoo Yun et al. *CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features*. 2019. arXiv: 1905.04899 [cs.CV]. URL: https://arxiv.org/abs/1905.04899.

[124] Hongyi Zhang et al. *mixup: Beyond Empirical Risk Minimization*. 2018. arXiv: 1710.09412 [cs.LG]. URL: https://arxiv.org/abs/1710.09412.

[125] Jian Zhang and Jiangqun Ni. "Domain-Invariant Feature Learning for General Face Forgery Detection". In: *2023 IEEE International Conference on Multimedia and Expo (ICME)*. 2023, pp. 2321–2326. DOI: 10.1109/ICME55011.2023.00396.

[126] Han Zhao et al. *Principled Hybrids of Generative and Discriminative Domain Adaptation*. 2017. arXiv: 1705.09011 [cs.LG]. URL: https://arxiv.org/abs/1705.09011.