Review

In-Depth Review of YOLOv1 to YOLOv10 Variants for Enhanced Photovoltaic Defect Detection

Muhammad Hussain * and Rahima Khanam

Department of Computer Science, University of Huddersfield, Huddersfield HD1 3DH, UK; rahima.khanam@hud.ac.uk
* Correspondence: m.hussain@hud.ac.uk

Abstract: This review presents an investigation into the incremental advancements in the YOLO (You Only Look Once) architecture and its derivatives, with a specific focus on their pivotal contributions to improving quality inspection within the photovoltaic (PV) domain. YOLO's single-stage approach to object detection has made it a preferred option due to its efficiency. The review unearths key drivers of success in each variant, from path aggregation networks to generalised efficient layer aggregation architectures and programmable gradient information, presented in the latest variant, YOLOv10, released in May 2024. Looking ahead, the review predicts a significant trend in future research, indicating a shift toward refining YOLO variants to tackle a wider array of PV fault scenarios. While current discussions mainly centre on micro-crack detection, there is an acknowledged opportunity for expansion. Researchers are expected to delve deeper into attention mechanisms within the YOLO architecture, recognising their potential to greatly enhance detection capabilities, particularly for subtle and intricate faults.

Keywords: computer vision; convolutional neural networks; deep learning; object detection; photovoltaic; quality inspection: manufacturing; YOLO

check for updates

Citation: Hussain, M.; Khanam, R. In-Depth Review of YOLOv1 to YOLOv10 Variants for Enhanced Photovoltaic Defect Detection. *Solar* **2024**, *4*, 351–386. https://doi.org/10.3390/solar4030016

Academic Editor: Jürgen Heinz Werner

Received: 10 May 2024
Revised: 5 June 2024
Accepted: 20 June 2024
Published: 26 June 2024

1. Introduction

The rapid advancements in Artificial Intelligence (AI) have revolutionised various domains, including the manufacturing sector. Within the realm of AI, machine learning (ML) has emerged as a powerful tool for automating complex tasks and improving efficiency. Deep learning (DL) and computer vision (CV), two key subsets of ML, have particularly garnered significant attention due to their ability to process and analyse visual data with unprecedented accuracy [1–9]. Convolutional neural networks (CNNs) [10–12], a type of DL architecture, have proven to be highly effective in tackling computer vision tasks. CNNs have been widely adopted for image classification, object detection, and segmentation, making them a valuable tool in manufacturing for quality inspection and defect detection.

In the context of manufacturing, object detection (OD) plays a crucial role in ensuring product quality and minimising defects. Traditional quality inspection methods, often relying on human operators, have limitations, such as subjectivity, human error, and high labour costs. The application of DL and CV techniques, particularly OD algorithms, offers a promising solution to overcome these challenges and improve the accuracy and efficiency of quality inspection processes. Among the various OD algorithms, the You Only Look Once (YOLO) architecture [13,14] has gained significant attention due to its unique single-stage framework. Unlike two-stage models like region convolutional neural networks (R-CNNs), YOLO streamlines the detection pipeline, resulting in enhanced performance and superior results [15]. The effectiveness of YOLO has been demonstrated across a wide range of applications [16–18], making it a popular choice for researchers and practitioners alike.

In the specific context of photovoltaic (PV) manufacturing [19–21], quality control is of the utmost importance. The global surge in PV installations, driven by the need to address climate change [22], has highlighted the significance of PV systems as a sustainable energy solution [23–25]. PV solar cells, often referred to as "green energy" sources [26,27], have the remarkable ability to absorb and convert large amounts of incident light energy from the sun [28,29]. However, the complex manufacturing process, from silicon extraction to wafer slicing, exposes solar cell surfaces to various intricate and demanding stages [30]. As a result, implementing a stringent quality control regime is crucial for ensuring the reliability and efficiency of PV systems. This review aims to provide a comprehensive introduction to the fundamental concepts of CNNs within the context of PV production. We specifically focus on the remarkable advancements in YOLO variants, which have undergone rapid evolution in a relatively short timeframe. By exploring the application of YOLO architectures in PV defect analysis, we highlight their potential to revolutionise quality inspection processes in the PV manufacturing industry.

1.1. Survey Objective

The global shift toward sustainable energy has positioned PV technology as a key player in meeting the increasing demand for green power. However, the efficiency and reliability of PV systems heavily depend on the quality of solar cells produced during manufacturing. Defective cells that pass quality control can significantly reduce plant efficiency and increase costs. The PV industry faces a critical challenge in ensuring the quality of solar cells during production, as traditional human-led inspection methods have proven to be inadequate in detecting defects consistently and efficiently. These methods face challenges such as subjectivity, high labour costs, inspector fatigue, and human error, resulting in suboptimal PV system performance and increased costs. This research problem highlights the need for innovative solutions that can improve the accuracy and speed of defect detection in PV manufacturing.

Our study focuses on answering the following research questions:

1. How can CV and DL techniques, particularly OD models, be leveraged to enhance defect detection in PV manufacturing?
2. What are the advantages of the You Only Look Once (YOLO) architecture and its variants compared to other OD methods for PV fault detection?
3. How has the evolution of YOLO architectures, from version 1 to version 10, contributed to improved performance in defect detection tasks?
4. What are the current applications and potential future directions of YOLO variants in the PV domain, specifically for quality control during production?

To address these research questions, this review article presents the first comprehensive analysis of YOLO variants and their applications in the PV domain, with a special focus on defect detection during production. We provide an in-depth examination of each YOLO architecture, from its inception in 2015 to the recent release of YOLOv10 in May 2024, highlighting their unique features and improvements. Furthermore, we explore the use cases of YOLO variants across the PV domain, discussing their potential to revolutionise quality control processes.

The main contributions of this review are as follows:

1. We address the growing need for effective quality control in the PV industry, driven by the increasing demand for green energy.
2. We identify the limitations of human-led inspection methods and the need for accurate and efficient alternatives, quantifying their impact on PV production efficiency and cost.
3. We explore the potential of CV and DL techniques, particularly YOLO variants, for non-invasive defect detection in the PV industry.
4. We provide a comprehensive analysis of YOLO's evolution, from its initial development to the latest advancements in YOLOv10, highlighting its unique features and performance improvements.

5. We present a thorough review of existing research on the application of YOLO variants in PV defect detection and discuss their potential to enhance quality control processes.

The findings of this review can inform future research and development efforts in PV quality control, ultimately contributing to the optimisation of solar cell manufacturing and the widespread adoption of PV technology. By leveraging the power of YOLO and its variants, the PV industry can take significant strides toward meeting the growing demand for clean energy while ensuring the highest standards of quality and efficiency.

1.2. Organisation of Paper

This paper is subsequently divided into the following sections: Section 2 presents an overview of CNNs as a preliminary to facilitate readers' understanding of the fundamental principles that stimulate the YOLO framework. Section 3 provides a detailed explanation of OD techniques to deliver a contextual background for the review. The subsequent section, Section 4, focuses on reviewing the application of CNNs in PV fault detection, encompassing an evaluation of research progress, challenges encountered, and opportunities within the field. Next, Section 5 explores the evolution of YOLO architectures comprehensively and coherently, investigating the modifications and advancements introduced in each successive iteration, from YOLOv1 to YOLOv10. This is followed by Section 6, which exclusively analyses the implementation of YOLO variants in PV fault detection applications. Section 7 discusses the findings of the review, culminating in a comprehensive assessment of the architecture's suitability as a viable solution for autonomous PV fault detection. Finally, Section 8 coherently summaries the key points throughout the paper, offering a conclusive evaluation of YOLO's potential and limitations in the context of PV fault detection. The visual representation of the review is illustrated in Figure 1.

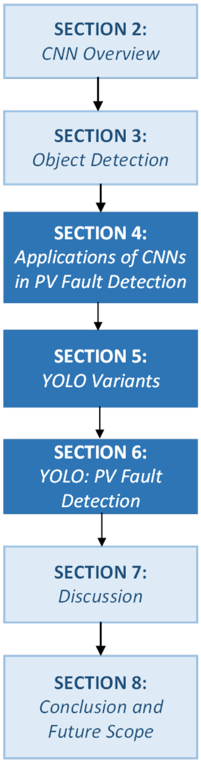

Figure 1. Visual structure of this review.

2. Convolutional Neural Networks (CNNs)

In the early 21st century, DL emerged as a significant advancement within the field of machine learning (ML), alongside widely used techniques such as Support Vector Machines (SVMs), Multilayer Perceptrons (MLPs), and Artificial Neural Networks (ANNs). DL, a subset of ML rooted in Artificial Intelligence (AI), has demonstrated remarkable success across a broad spectrum of disciplines. This is evident in its diverse applications, including biological data analysis [31], gene expression analysis [32], micro-blogging [33], speech recognition [34], character recognition [35], text classification [36], unstructured text data mining with fault classification [37], automatic landslide detection [38], intrusion detection [39], stock market prediction [40], and video processing tasks like caption generation [41]. These applications represent only a glimpse into the expansive potential held by deep learning methodologies.

Within the domain of computer vision (CV), the focus lies on training machines to achieve a sophisticated level of comprehension and interpretation of visual content. This field encompasses diverse subareas, such as object detection [42], image restoration [43], scene recognition [44], pose and motion estimation [45], object segmentation [46], and video tracking [47]. Historically, conventional image processing techniques relied on the manual extraction of features, requiring the definition of specific feature descriptors. However, DL architectures present a compelling alternative by employing deep neural networks, which inherently function as automatic feature extractors. This inherent ability of DL models to learn features directly from data enables researchers to overcome the limitations of conventional image processing methods. Consequently, they can dedicate more resources to refining the application-specific performance of the network, rather than focusing solely on developing feature extraction infrastructure.

DL encompasses a variety of models that have significantly advanced the field of AI. Recurrent Neural Networks (RNNs) [48] are adept at handling sequential data, making them suitable for natural language processing [49]. Their architectural variants, such as Long Short-Term Memory networks (LSTMs) [50] and Gated Recurrent Units (GRUs) [51], address the vanishing gradient problem [52], improving their ability to capture long-term dependencies. Transformer models, like BERT (Bidirectional Encoder Representations from Transformers) [53] and GPT (Generative Pre-trained Transformer) [54], have achieved breakthroughs in natural language understanding. Generative models, including Variational Autoencoders (VAEs) [55] and Generative Adversarial Networks (GANs) [56], focus on generating new data instances. These models collectively contribute to the diverse applications of deep learning, ranging from computer vision and speech recognition to language translation and generative art. Among all the models, CNNs are widely used for image recognition tasks [57–59].

CNNs [60] are a specialised type of ANN that have proven exceptional performance across a broad spectrum of CV tasks like object detection [61], image classification [62], image captioning [63], image segmentation [64], image retrieval [65], speech processing [66], facial recognition [67], pose estimation [68], traffic sign recognition [69], and neural style transfer [70].

CNNs have witnessed a surge in research interest in recent years, although their development has a longer history. A seminal work in this area was published by Hubel and Wiesel in 1959 [71]. Through a series of experiments investigating the functionality of neurons in the visual cortex, the authors revealed its hierarchical organisation, consisting of both simple and complex neurons within the primary visual cortex. Notably, the processing of visual information consistently commences with the detection of fundamental structures, such as oriented edges, with complex cells receiving input from lower-level simple cells via receptive fields. In 1980, Fukushima presented the Neocognitron model [72], the first ANN explicitly inspired by the biological observations of Hubel and Wiesel on simple and complex cells in the visual cortex. Building upon Fukushima's existing work, in 1989, LeCun et al. successfully applied backpropagation [73] to achieve a 1% error rate and approximately 9% rejection rate on the challenging task of recognising handwritten zip-code digits. Further refinements to the CNN architecture were introduced by LeCun et al. in 1998 through the implementation of an error gradient-based learning algorithm [60]. A pivotal moment arrived in 2012 with the introduction of AlexNet by Krizhevsky et al. [74]. This model, the first deep convolutional neural network (DCNN), boasted a more complex architecture compared to previous attempts. AlexNet's remarkable success, achieving significant performance improvements in CV tasks, triggered a revolution in the field, with advancements like the efficient utilisation of GPUs, the adoption of the Rectified Linear Unit (ReLU) activation function [75], the incorporation of a regularisation technique called Dropout [76,77], and the implementation of data augmentation strategies [78].

The core structure of a CNN, at its conceptual level, consists of a sequence of convolution, pooling, and activation functions that progressively transform the input into relevant outputs. Figure 2 presents a schematic representation of the core components comprising a convolutional neural network (CNN) at an abstract level. As depicted, the architecture consists of a sequential arrangement of convolutional blocks, culminating in a series of fully connected layers that ultimately produce the network's output [79]. A crucial aspect of convolutional blocks within a CNN lies in the specification of the number of kernels (or filters) and their spatial dimensions. These parameters hold significant importance, as they directly govern the feature extraction process. Early layers extract low-level spatial features, setting the stage for subsequent layers to develop higher-order semantic representations [80]. The convolution operation involves the element-wise multiplication of the kernel and the overlapping patch of the input image, followed by summation. The mathematical formula for feature extraction is represented in Equation (1):

$$Output[i,j] = \sum_{u=0}^{k_h-1} \sum_{v=0}^{k_w-1} Input[i+u, j+v] \cdot Kernel[u,v] \tag{1}$$

where $output[i,j]$ represents the element at position (i,j) in the output feature map. $Input[i+u, j+v]$ represents the input element at position $(i+u, j+v)$ in the input image. $Kernel[u,v]$ represents the weight at position (u,v) in the kernel. k_h is the height of the kernel. k_w is the width of the kernel.

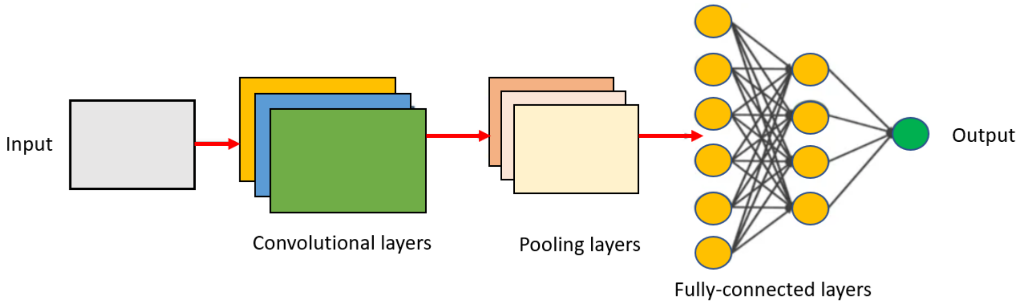

Figure 2. The general structure of a CNN, highlighting convolutional, pooling, and fully connected layers.

Following the convolution operation, the output features undergo a pooling step. This process aims to extract the most salient features through aggregation, effectively downsampling along the spatial dimensions (width and height) of the feature maps. Several pooling mechanisms are available for this purpose [81], including average pooling, sum pooling, and the widely employed max pooling. As an example, the max-pooling function applied to a one-dimensional input can be expressed as follows (Equation (2)):

$$a_x^l = \max(a_{(x-y)}^{(l-1)}, a_{(x+y)}^{(l-1)}) \tag{2}$$

The ReLU activation function, defined in Equation (3), has emerged as the dominant choice within CNN blocks [82]. This preference stems from its computational efficiency due to its inherent simplicity. ReLU operates as a max(0, x) function, making it significantly faster to evaluate compared to alternative activation functions such as the sigmoid and tanh functions, presented in Equations (4) and (5), respectively:

$$f(x) = \max(0, x) \tag{3}$$

$$x \to \frac{1}{1+e^{-x}} \tag{4}$$

$$\tanh(x) = 2\sigma(2x) - 1 \tag{5}$$

3. Object Detection

Designing object detectors presents several challenges for researchers and practitioners. One of the primary challenges is managing variations in image resolutions and aspect ratios. This issue becomes more pronounced when target objects vary significantly in the spatial dimension. Moreover, the presence of class imbalance, especially in scenarios where acquiring a sufficient number of images for specific classes, such as rarely occurring defects on production lines, is challenging, can significantly hinder the performance of object detection models. This phenomenon often leads to biased predictions, as the model priorities the dominant classes with abundant data while under-performing on the underrepresented ones [83].

Another significant challenge is the computational complexity of OD architectures, which can be resource-intensive in terms of computational power, memory, and time [84,85]. Figure 3 demonstrates OD for single and multiple objects in an image. Object detectors consisting of a deep internal network demand substantial computational resources for processing complex image datasets and extracting key features.

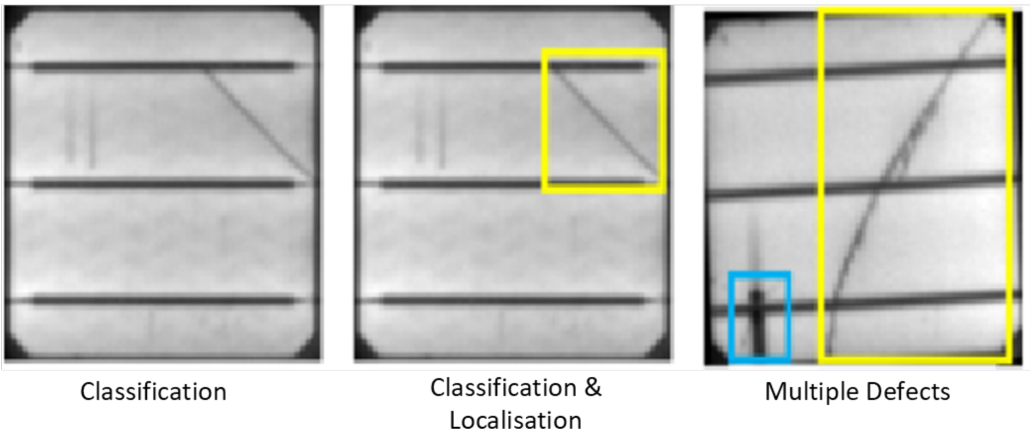

Figure 3. Single and multiple objects in an image: classification, localisation, and segmentation.

OD can be broadly categorised into two-stage object detectors and single-stage detectors. The former can be defined as a class of CNNs that first propose candidate regions in the given image that may contain objects and then perform classification and localisation within those proposed regions. Among the most prominent two-stage detectors are the RCNN (region-based convolutional neural network) [86], Fast R-CNN [87], Faster R-CNN [88] incorporating mechanisms such as ROI (Region of Interest) pooling, and FPN (feature pyramid network) [89].

The **RCNN** [86] was introduced in 2014, and it utilised a selective search to propose potential candidate regions. After candidate generation, a CNN network was utilised for feature extraction, followed by an SVM classifier for the ultimate classification and localisation. Although it provided satisfactory performance in terms of accuracy, it was computationally inefficient due to the two-stage process.

Fast R-CNN [87] tackled the efficiency issues of its predecessor (RCNN) by proposing ROI pooling. Rather than processing individual region proposals, Fast R-CNN employed ROI pooling to extract fixed-size feature maps for each region from the original feature maps. This resulted in substantial computational speed-up, as the feature extraction process became shared across all region proposals.

Faster R-CNN [88] further improved upon Fast R-CNN by proposing the Region Proposal Network (RPN). The RPN was an integral part of the network, generating region proposals directly from the convolutional feature maps, thus eliminating the need for an additional region proposal stage. By integrating the RPN into Fast R-CNN, Faster R-CNN achieved faster and more accurate detection results. ROI pooling was also a critical component in both Fast R-CNN and Faster R-CNN, enabling efficient region-based feature extraction and allowing the networks to handle candidate proposals of varying spatial dimensions and shapes effectively.

The **feature pyramid network (FPN)** [89] can be regarded as an enhancement of two-stage detectors, addressing the challenge of detecting targets at multiple scales. It generates a feature pyramid by incorporating feature maps of varying spatial resolutions from different stages of the network. By enabling the model to detect targets of different scales, the overall performance and robustness of architecture are improved.

Two-stage detectors have demonstrated impressive accuracy and have become foundational building blocks for various applications that require precise and reliable OD; however, their high computational demand limits their application base. Single-stage detectors aim to detect objects in a single pass, eliminating the need for a separate region-proposal step. These detectors directly predict the bounding boxes and class probabilities for all target objects in a single pass, making them more computationally friendly compared to two-stage detectors. The general schematic of single-stage object detectors is exemplified in Figure 4. Notable single-stage detectors include the Single-Shot Multibox Detector (SSD), You Only Look Once (YOLO) variants, RefineDet++, the Deconvolution Single-Shot Detector (DSSD), and RetinaNet.

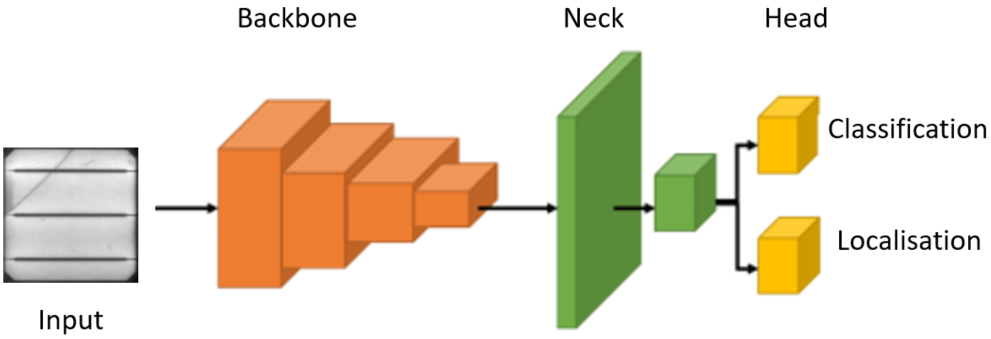

Figure 4. A standard architecture of single-stage object detectors.

The **SSD** [90] employs multiple convolutional feature maps at different scales for predicting bounding boxes and class probability scores. By employing default anchor boxes at several aspect ratios and scales, the SSD can effectively detect objects of different sizes and shapes in a single forward pass through the network.

RefineDet++ [91] is an expansion of the original RefineDet architecture, aimed at refining target proposals in an iterative fashion through multiple stages. RefineDet++ further improves the accuracy by deploying enhanced feature fusion mechanisms and refining target boundaries for better localisation.

DSSD incorporates deconvolution layers to reclaim spatial information lost during feature pooling. This aids in maintaining the spatial resolution of feature maps and allows the DSSD to capture fine-grained details for accurate object localisation.

RetinaNet [92] focuses on addressing the issue of class imbalance by presenting the Focal Loss. The Focal Loss assigns higher weights to hard, misclassified samples, thus improving the architecture's ability to handle class imbalance and boost overall detection performance.

Single-stage detectors offer several advantages, including faster inference speeds and lightweight footprints compared to two-stage detectors. They are a popular choice for resource-constrained environments due to their computational simplicity and real-time inference capacity. Among single-stage detectors, YOLO has emerged as a formidable competitor to both two-stage and previous single-stage object detectors, boasting impressive accuracy and inference speed. The single-stage design of YOLO, coupled with continuous architectural refinements, has established recent YOLO variants as a compelling choice for real-world industrial applications demanding real-time object detection. An extensive exploration of YOLO's architectural advancements is presented in Section 5.

4. Applications of CNNs in PV Fault Detection

Research on AI, notably CV, has been conducted to overcome the limitations of human inspection. This section explores how various researchers have increased the output efficiency of solar cell production by various means and methods.

As demonstrated in Figure 5, a CNN architecture aimed at identifying defective PV cells was introduced by M. Waqar Akram et al. [93]. Notably, the authors achieved a remarkable accuracy rate of 99.23%. This feature was accomplished through an "isolated-model" approach (98.67%), which was subsequently adapted to EL-based images using transfer learning. The authors asserted that employing a deep architecture may potentially lead to overfitting due to the relatively modest dataset size of approximately 800 images. Their research unfolded in two distinct stages.

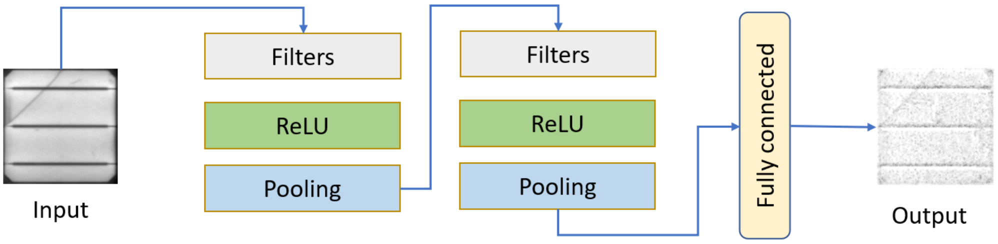

Figure 5. A proposed abstract architecture.

The first stage encompassed the construction of a CNN tailored for EL images. Subsequently, in the second phase, the CNN trained on EL images was repurposed as a pre-trained model. This was further fine-tuned using infrared (IR) images of defective cells. The pre-trained EL-based model exhibited an accuracy of 98.67%. It is worth noting, however, that a closer examination of the methodology, particularly concerning data collection and pre-processing, revealed that the inclusion of false flaws, rather than authentic defects, within the cell images might have occurred. To enhance the size and diversity of the dataset, the authors leveraged data augmentation, culminating in a 6.5% increase in model accuracy.

Upon careful scrutiny of the CNN's design, the authors opted for a four-block CNN architecture coupled with a fully connected layer that feeds into a SoftMax function. Although alternative optimisers such as stochastic gradient descent (SGD) are prevalent, the authors chose to adopt Adaptive Moment Estimation (ADAM) owing to its established reputation and frequent utilisation within the field.

Sachin Mehta et al. [94] proposed a CNN for the purpose of detecting PV soiling and associated faults. Their primary focus centred on identifying the location and presence of defects within PV cells. Historically, the term "OD" has been employed to delineate the classification and localisation aspects of images. A prevailing strategy to tackle OD challenges entails the creation of bounding boxes encompassing specific regions of interest. These bounding boxes subsequently serve as the basis for training supervised models. However, the authors' approach introduces an intriguing departure. Rather than conventional bounding boxes, they advocate substituting these with power loss values for each image. This departure could be considered an innovative contribution to the research landscape. Nonetheless, certain concerns emerge regarding the feasibility of applying this proposed methodology to smaller datasets. Notably, the dataset employed in this study encompassed over 45,000 images for network training, a relatively substantial scale.

Additionally, the model's operational capacity is confined to 22 frames per second (FPS), a limitation highlighted. It is pertinent to note that the authors' assessment of computational load was approximated using an atypical hardware component, the NVIDIA TitanX.

Ahmad et al. [95] undertook a comparative analysis of three distinct models in the context of detecting solar cell defects based on EL images. The trio of models includes the Random Forest (RF) and SVM models, both belonging to the realm of ML, and the CNN, representing a DL approach. In their study, the CNN, featuring two convolutional blocks and a fully connected layer, exhibits the potential to achieve accuracy rates surpassing 99%. The authors attribute a significant portion of this achievement to the employment of dataset augmentation, which effectively augments the original dataset by a factor of four. This augmentation strategy plays a pivotal role in enhancing the model's performance and contributing to impressive model accuracy.

It is crucial to note that, with the strides made in DL techniques, as well as the incorporation of regularisation and advanced data processing methodologies, models yielding accuracies below 90% no longer command the status of groundbreaking research. However, Sergiu Deitsch et al. [96] introduced a novel dimension by proposing the utilisation of both an SVM and a CNN network for the detection of multiple defects within EL-based solar cell images. Notably, the authors report commendable accuracy for both CNN (88.42%) and SVM (82.44%). In their endeavour, the authors embrace the realm of transfer learning by leveraging the VGG-19 architecture. The final fine-tuning is carried out through a two-step process. Initially, the ADAM optimiser is employed for weight updates, with the weights of the fully connected layer initialised randomly. Subsequently, the weights across all layers undergo refinement in the second stage. An intriguing point of consideration is the authors' adjustment of the "momentum" value to 0.9 and the utilisation of stochastic gradient descent (SGD) during the second phase. It is worth clarifying that "momentum" serves as a hyperparameter in the context of SGD with momentum (SGD-M), which differs from the standard SGD optimisation approach. The rationale behind the modification of the ADAM optimiser during the second fine-tuning stage remains ambiguous and warrants further elucidation.

In the pursuit of detecting faults within EL-based solar cell images, Yang Zhao et al. [97] presented an innovative approach involving the utilisation of a Mask R-CNN architecture with a RESNET-101 backbone. The authors' primary objective centres around the classification of 19 distinct types of flaws. Rigorous testing of the model revealed an achieved mean average precision (mAP) of 70.2%, marking a noteworthy milestone. Notably, this metric, considering a 0.5 mAP localisation threshold, signifies a reliable prediction. The predicted bounding box, in this context, maintains a minimum of 50% Intersection over Union (IoU) with the ground-truth bounding box, substantiating its credibility. However, the authors acknowledge the attainment of only moderate accuracy outcomes, prompting the introduction of additional refinements to their methodology. Notably, they introduce three tiers of defect severity into their dataset. Strikingly, the most challenging-to-detect flaws were assigned lower severity ratings and consequently deemed non-flaws. This strategic classification contributed to a substantial mAP increase of over 90%.

Before delving into architectural considerations, Ashfaq Ahmad et al. [98] underscore the pivotal role of data augmentations in expanding datasets and introducing variability. In their quest for fault detection, the authors address the realm of EL-based solar cell images utilising a CNN architecture, achieving an accuracy of 91.58%. The chosen CNN architecture initiates with image input and subsequently traverses through four convolutional blocks, totalling 32 filters. The subsequent two convolutional blocks accommodate 64 filters each, and the final two blocks comprise 128 filters. Collectively, the structured CNN architecture encompasses eight convolutional blocks, culminating in the output being channelled through a single fully connected layer. While the escalation in the number of kernels aligns with the pattern of more intricate models, the rationale for employing eight convolutional blocks lacks explicit justification.

Wuquin Tang et al. [99] presented a CNN-based approach to detecting errors within EL-based cell images. The authors' primary contribution rests on the introduction of a GAN for data augmentation. However, the application of this GAN-based data augmentation bears scrutiny. After employing the GAN for data augmentation, the resultant overall accuracy of 83% raises questions about the appropriateness of this method within this particular context. The rationale behind opting for a GAN to scale the dataset and enhance variance remains unclear.

A comparative analysis could have enriched the assessment of the GAN's efficacy by juxtaposing its accuracy against that of established data augmentation techniques inherent to deep learning frameworks like Keras, TensorFlow, and PyTorch. Moreover, it is worth noting that the GAN's employment significantly escalates computational demands and resource allocation compared to traditional data augmentation. The GAN's computational intensity stems from its role as a network in its own right, utilised to generate new images.

In devising DL models for defective solar cell detection, Christopher Dunderdale et al. [100] embarked on training their models using the VGG-16 [101] and MobileNet [102] architectures for comparison. A noteworthy observation is that, despite the availability of ADAM as a readily available optimiser, the authors opted for an unconventional path by evaluating SGD with Adam. Upon delving into the VGG-16-trained architecture, it became evident that the SGD optimiser, in conjunction with data augmentation techniques involving horizontal flipping, vertical flipping, and rotation, yielded superior performance with an accuracy of 85.6%. Intriguingly, the ADAM optimiser under the same conditions resulted in a subpar accuracy of 27.4%. However, transitioning to the MobileNet architecture brought about the most impressive accuracy of 89.5%, facilitated by data augmentations, including horizontal and vertical flips, and the adoption of the ADAM optimiser.

Pierdicca R et al. [103] introduced a CNN built upon the VGG-16 architecture for the purpose of identifying defective PV cells. The authors justify their choice of the VGG-16 architecture based on its user-friendly nature. However, it is important to note that various frameworks, such as PyTorch and TensorFlow developed by Google and Facebook, facilitate the utilisation of state-of-the-art (SOTA) pre-trained models. Therefore, the selection of an architecture should ideally align with the dataset's characteristics rather than being solely driven by ease of construction.

To address anomalies across diverse surface textures, including solar cells, Haiyong Chen and colleagues [104] introduced a weakly supervised surface-defect-detection architecture. The authors put forth a fused design that amalgamates a CNN with a Random Forest (RF) classifier, asserting its enhanced robustness in advanced background-filtering scenarios. The dataset under scrutiny comprised 15,330 normal and 5915 faulty images. The architecture integrated an attention network and a Random Forest classifier, both positioned after the five convolutional blocks composing the CNN architecture. A noteworthy choice made by the authors was the implementation of K-fold cross-validation with K = 5. Given the pronounced class imbalance within the dataset, this approach is commendable and has led to cross-validation results showcasing an accuracy of 93.23%.

Bolun Du et al. [105] shared their exploration into detecting flaws in PV cells within production lines. Their approach involves the application of the commonly utilised fine-tuning strategy to train three SOTA models: GoogleNet, LeNet, and VGG-16. Remarkably, GoogleNet emerged as the front-runner, achieving a remarkable 100% accuracy and a loss of 0.002 after 81 epochs. It is worth noting that, despite its exceptional accuracy, GoogleNet's model incorporates 13 million learnable parameters. These findings align with those of Hussain et al. [106], who reinforced the efficacy of GoogleNet's architecture by obtaining impressive results. Here, the authors evaluated each model's accuracy post-deployment and assessed them against architectural criteria. Interestingly, their research underscores GoogleNet's outstanding precision. However, it is notable that GoogleNet's architecture employs 13 million learnable parameters, which deviates from the suggested architecture.

Mustafa Yusuf et al. [107] employ a two-stage approach for the detection of diverse defects within EL-based solar cells. Their methodology begins with the application of various data augmentation techniques, with a specific emphasis on the defective classes, to effectively expand the dataset. Subsequently, feature extraction from this augmented dataset is pursued through the training of four widely recognised models: VGG-16, VGG-19, ResNet-50, and DarkNet-19, employing the transfer learning paradigm. Having distilled the most distinguishing features, the authors proceed to classification employing a range of ML architectures, which encompass Random Forests and SVMs. The authors' strategy resonates with analogous research, further highlighting the substantial computational demands associated with these pre-trained network models.

N. Kellil et al. [108] proposed an IRTI-based approach for identifying and classifying faults in PV modules. Two datasets, binary classification (BC) and multiclass classification (MC), were used to train and assess two models for differentiating between healthy and faulty modules. A simple DCNN architecture achieved an average accuracy of 98.39% for fault detection, which was further improved to 99.91% using transfer learning with a VGG-16 model. For the fault classification of five types, the small-DCNN model achieved an average accuracy of 91.63%, which was significantly boosted to 99.80% with VGG-16 fine-tuning. Both detection and classification models demonstrated high accuracy, making them promising for real-time applications. The authors stated that future work would focus on optimising and integrating these models into low-cost microprocessors/microcontrollers for cost-effective and portable PV system diagnostics. This approach has the potential to significantly improve fault detection and classification in PV systems.

The study by [109] proposed a novel DL architecture for fault detection and diagnosis in PV systems. The architecture integrates CNN and Bidirectional Gated Recurrent Unit (Bi-GRU) layers to leverage their complementary strengths. CNN layers enable feature extraction from data, while Bi-GRU layers capture temporal dependencies. This combination facilitates superior fault classification, including open circuits, short circuits, and partial shading. The approach is further bolstered by a precisely calibrated simulation model and a comprehensive database encompassing normal and abnormal PV operations. Evaluations demonstrate exceptional accuracy exceeding 99% in both fault detection and diagnosis, highlighting the effectiveness of the proposed method. As stated by the authors, future work will investigate the generalisability of the approach on diverse PV systems and explore real-time implementation on hardware platforms.

Summary of CNN-Based PV Fault Detection Models

Table 1 presents a comprehensive overview of various CNN architectures and their applications in detecting faults in PV systems from 2018 to 2024. The studies utilised a range of CNN architectures, including custom CNNs and popular pre-trained models such as VGG16, ResNet50, InceptionV3, and MobileNet, as well as more advanced techniques, like Mask R-CNN and attention mechanisms. The dataset sizes used in these studies vary from a few hundred to tens of thousands of images, demonstrating the scalability and adaptability of CNN-based approaches. The images used include infrared thermal images, EL images, and visual images of PV cells and modules. Several key contributions and innovations are highlighted, such as the use of transfer learning, weakly supervised learning, and the development of novel frameworks like DeepSolarEye and the Distance-Aware Network (DAN). Some studies also compare the performance of CNNs with traditional machine learning techniques like SVM and Random Forests. The reported accuracies for fault detection and classification range from 70% to over 99%, with most studies achieving accuracies above 90%, demonstrating the effectiveness of CNN-based approaches in accurately identifying various types of PV faults, including micro-cracks, soiling, and defects in different types of PV cells (e.g., monocrystalline and polycrystalline).

Table 1. Summary of CNN applications for PV fault detection.

Reference	Year	Architecture	Dataset Size	Contribution	Accuracy
[93]	2020	Custom CNN	893	Detects PV defects in infrared images using DL and transfer learning with a pre-trained model, achieving real-time prediction on CPUs.	98.67%
[94]	2018	CNN	45,000+	DeepSolarEye detects real-time solar panel soiling and performs defect analysis, predicting the power loss, soiling localisation, and category using a weakly supervised approach with a novel BiDIAF block and web-scraped data for soiling type classification.	∼95%
[95]	2019	SVM, RF, and CNN	12,070	A pipeline for PV fault classification using supervised learning with SVM, RF, and CNN algorithms, achieving the best performance with a CNN model trained on an augmented dataset.	<99%
[96]	2019	SVM and CNN	2624	Two contrasting PV cell defect detection methods: a hardware-efficient, SVM-based approach with hand-crafted features and a high-accuracy, GPU-powered CNN.	CNN—88.42%; SVM—82.44%
[97]	2020	Mask R-CNN and ResNet-101-FPN	5983	A DL approach for automatic multi-defect detection in PV modules using EL images.	<95%
[98]	2020	SVM and CNN	2624	Feature extraction with SVM (HOG, KAZE, SIFT, SURF) and a CNN for classifying seven types of solar cell defects.	CNN—91.58%; SVM (68.90–72.74%)
[99]	2020	CNN, VGG16, ResNet50, InceptionV3 and MobileNet	1800	A GAN finds defects in EL-based cell images instead of VGG16, ResNet50, Inception V3, and MobileNet.	83%
[100]	2019	CNN, VGG16, and MobileNet	383	A cost-effective DL and feature-based approach for PV module defect detection and classification using thermal infrared images.	91.2%
[103]	2018	VGG16	3336	Demonstrates degradation issues and assesses the suggested approach.	<70%
[104]	2020	CNN and Random Forest	21,245	A CNN–Random Forest architecture with attention for robust weakly supervised defect detection on diverse surface textures, including solar cells.	93.23%
[105]	2020	LeNet-5, VGG-16, and GoogleNet	720	An automatic Si-PV cell defect detection system using IRT imaging, PCA/ICA/NMF feature extraction, and GoogleNet classification.	97.64%
[107]	2020	DarkNet19, ResNet50, VGG16, and VGG19	2624	A novel DFB framework combining DL feature extraction with SVM.	90.57–94.52%
[108]	2023	DCNN and VGG16	BC—5294; MC—4956	A fine-tuned VGG-16 model achieving high accuracy in fault detection and identification for 5 types of defects in photovoltaic modules using thermographic images.	99.91%
[109]	2024	Custom CNN	∼23,000	A three-step process involving robust PV modelling, comprehensive data creation, and a CNN-Bi-GRU-based feature extractor for effective fault classification.	99%
[110]	2020	ResNet-50	2000	A transfer learning approach using a DAN with MK-MMD for low-cost, high-efficiency defect detection in polycrystalline solar cells, leveraging labelled data from monocrystalline cells.	77%
[111]	2022	Custom CNN	777	A lightweight framework for the automated detection of micro-cracks in PV cell surfaces using EL imaging.	99%

Notable contributions include the development of real-time fault detection systems, the ability to predict power loss due to faults, and the localisation of soiling and defects. Some studies also focus on cost-effective and hardware-efficient solutions, such as lightweight CNNs and the use of feature extraction techniques like HOG, KAZE, SIFT, and SURF in combination with SVMs. More recent studies (2022–2024) demonstrate further advancements in PV fault detection using CNNs, including the development of lightweight frameworks for micro-crack detection and the integration of robust PV modelling and comprehensive data creation with CNN-based feature extractors for effective fault classification. Seeing the current progress, more sophisticated and efficient CNN architectures will be developed to further improve the accuracy, reliability, and cost-effectiveness of PV fault detection systems as research in this field continues to evolve.

5. YOLO Architecture Background

This section of the review will delve into the foundational principles and architecture that underlie YOLO. Subsequently, the distinct advancements associated with each iteration of YOLO will be elucidated. Then, a detailed examination of the distinctive advancements introduced in each successive iteration of YOLO will be presented.

The YOLO algorithm, introduced in 2016 by Joesph Redmon et al. [15], is an acronym for "You Only Look Once". This name stems from its unique approach, where a comprehensive image is examined only once to discern objects and their respective positions. In contrast to conventional methods that adapt classifiers for a two-stage detection process, leading to intricate pipelines requiring separate training for each component, YOLO approaches OD as a regression problem [15].

In the YOLO paradigm, the anticipation of bounding boxes and class probabilities within an image is accomplished using a solitary CNN. This streamlined approach stands in contrast to the more convoluted pipelines associated with traditional methods.

5.1. YOLOv1

The fundamental principle introduced by YOLOv1 involves the introduction of a grid cell with dimensions of "S x S" overlaid onto the image. When the centre of an object of interest falls within one of these grid cells, that specific cell is designated to be responsible for detecting the object. This strategic approach enables other cells to disregard the object's presence in case of multiple occurrences.

For the implementation of OD, each grid cell is tasked with predicting bounding boxes, accompanied by their respective dimensions and confidence scores. This confidence score denotes the likelihood of an object's presence within the given bounding box. Mathematically, the confidence score can be represented as Equation (6):

$$\text{confidence score} = c(\text{object}) \times \text{IoU}_{\text{truth pred}} \tag{6}$$

where $c(object)$ signifies the probability of the object being present, with a range of 0–1, with 0 indicating that the object is not present, and $IoU_{truthpred}$ represents the IoU with the predicted bounding box with respect to the ground-truth bounding box. Each bounding box consists of five components (x, y, w, h, and the confidence score), with the first four components corresponding to the centre coordinates (x, y, width, and height) of the respective bounding box.

The core objective of YOLO, and OD as a whole, revolves around the precise identification and localisation of objects through bounding boxes. This necessitates the utilisation of two sets of bounding box vectors: the ground-truth vector, denoted by vector y, and the predicted vector, denoted by vector $\hat{y}$. To mitigate challenges arising from multiple bounding boxes either containing no objects or representing the same object, YOLO incorporates non-maximum suppression (NMS). This process involves eliminating overlapping predicted bounding boxes that exhibit an IoU value below a defined NMS threshold.

To address the issue of multiple bounding boxes for the same object or bounding boxes with a confidence score of zero (indicating the absence of an object), the authors introduced distinct penalties. Bounding boxes containing objects are significantly penalised ($\gamma_{\text{coord}} = 5$), while those indicating the absence of an object receive a milder penalty ($\gamma_{\text{noobj}} = 0.5$). The cumulative loss function computes the sum of all bounding box parameters, including the coordinates (x, y), width, height, confidence score, and class probability.

The first component of the loss function computes the loss of bounding box predictions concerning ground-truth bounding boxes, specifically focusing on coordinates x_{center} and y_{center}. In this context, obj is set to 1 if an object resides within the j-th bounding box prediction of the i-th cell; otherwise, it is set to 0. The selected predicted bounding box is tasked with predicting the object with the highest IoU, as depicted in Equation (7):

$$\gamma \sum_{S^2} \sum_{B} \text{obj}(x - \hat{x})^2 + (y - \hat{y})^2 \tag{7}$$

The subsequent part of the loss function quantifies the prediction error in the width and height of bounding boxes. The normalisation of width and height to a range between 0 and 1 ensures that their square roots amplify differences for smaller values compared to larger values, as shown in Equation (8):

$$\sum_{S^2} \text{obj} \sqrt{(\hat{w} - w)^2 + (\hat{h} - h)^2} \tag{8}$$

The loss of the confidence score is then computed based on the presence or absence of an object with respect to the bounding box. The penalty for the object confidence error is applied if the predictor is responsible for the ground-truth bounding box. Here, obj is set to 1 when the object is present in the cell; otherwise, it is set to 0. On the contrary, noobj functions inversely, as demonstrated in Equation (9).

$$\sum_{S^2} \sum_{B} \text{obj}(c - \hat{c})^2 + \gamma \sum_{S^2} \sum_{B} \text{noobj}(x - \hat{x})^2 + (c - \hat{c})^2 \tag{9}$$

The final element of the loss function, analogous to normal classification loss, calculates the loss of class (c) probabilities, excluding the obj portion, as detailed in Equation (10):

$$\sum_{S^2} \text{obj} \sum_{i=0}^{\text{classes}} (p(c_i) - \hat{p}(c_i))^2 \tag{10}$$

The initial YOLO architecture, based on the Darknet framework, comprised two subvariants. The first variant featured 24 convolutional layers, culminating in a connection to the first of two fully connected layers. Conversely, the "Fast YOLO" variant comprised only nine convolutional layers, each hosting fewer filters. Drawing inspiration from the inception module in GoogleNet, a sequence of 1 × 1 convolutional layers was implemented to condense the feature space derived from prior layers.

In terms of performance, the simpler version of YOLO (with 24 convolutional layers) trained on the PASCAL VOC datasets (2007 and 2012) achieved an mAP of 63.4% while operating at a speed of 45 FPS. On the other hand, the Fast YOLO variant attained an mAP of 52.7% at an impressive frame rate of 155 FPS. While these results surpassed real-time detectors like DPM-v5, they fell short of the SOTA at that time, exemplified by Faster R-CNN's mAP of 71%.

However, several notable shortcomings were evident and demanded attention. For instance, the YOLO architecture exhibited relatively lower recall and higher localisation errors when compared to Faster R-CNN. Moreover, the architecture encountered challenges in detecting objects in close proximity due to the limitation of each grid cell being restricted to two bounding box proposals. These observed limitations served as crucial insights that influenced the subsequent development of various YOLO variants.

5.2. YOLOv2

Building upon the achievements of YOLOv1, YOLOv2 introduces further advancements in its architecture. This iteration draws inspiration from the Network-in-Network and VGG concepts. The Darknet-19 framework was chosen for YOLOv2, encompassing 19 convolutional layers along with 5 layers dedicated to maximum pooling, as depicted in Table 2. To facilitate downsampling within the network structure, YOLOv2 employs a blend of pooling layers and 1 × 1 convolutions.

Table 2. Darknet-19 framework [112].

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

One critical challenge in OD lies in the scarcity of labelled data, which often limits techniques to classifying a predefined set of categories. YOLOv2 addresses this limitation by leveraging scalability through the merger of ImageNet and the COCO dataset [113], enabling detection across a vast range of over 9418 object instances. To enhance scalability, YOLOv2 employs Word-Tree, a hierarchical classification and detection approach that efficiently manages the increased number of categories.

YOLOv2 introduces significant improvements over V1, incorporating a range of data augmentation methods and novel optimisation techniques. Noteworthy advancements include the following:

- YOLOv2 introduces the ability to predict object dimensions across a spectrum of sizes, from 320 × 320 to 608 × 608. This flexibility is achieved by discarding fully connected layers, which were present in YOLOv1.
- YOLOv2 achieves a 4% mAP increase compared to V1 through the implementation of a higher-resolution classifier. Unlike V1, which enlarged images from 224 × 224 to 448 × 448, the YOLOv2 classifier trains on 448 × 448 images for classification before detection. Subsequent fine-tuning enhances bounding box accuracy for higher-resolution inputs.

- By addressing input distribution inconsistency during training, batch normalisation enhances learning efficiency and acts as a regularisation technique. This innovation results in an approximate 2% mAP improvement.
- YOLOv2 improves upon YOLOv1's direct bounding box coordinate prediction by introducing a method that predicts location coordinates in relation to grid cell locations. This adjustment leads to a 5% mAP increase, along with more uniform bounding box aspect ratios and sizes.
- YOLOv2 employs convolutional layers to extract features and predicts bounding boxes using anchor boxes, replacing fully connected layers. While enhancing recall by 7%, this modification slightly reduces mAP by 0.3%.
- YOLOv2 employs a clustering algorithm based on K-means to group similar bounding boxes. This approach eliminates the need for manually selecting anchor boxes, resulting in improved accuracy.

To address the challenge of detecting smaller objects, YOLOv2 integrates skip connections, inspired by ResNet. This technique combines high-resolution features with lower-resolution ones, allowing the accurate detection of objects of varying sizes and shapes. This refinement leads to a 1% increase in mAP. For instance, a 26 × 26 × 512 feature map transforms into a 13 × 13 × 2048 feature map, which is then concatenated with the model's output, enabling better object recognition across different dimensions.

5.3. YOLOv3

YOLOv3 [114] was introduced in 2018 by Joesph Redmon et al. This iteration brought significant enhancements that aligned with the latest technological advancements while retaining its real-time processing capability. An expanded architecture is presented in Table 3. Much like YOLOv2, YOLOv3 also predicts four coordinates for each bounding box. However, YOLOv3 introduces an objectness score for each box, determined through logistic regression. This score assumes values of 1 or 0, indicating whether the anchor box has the highest overlap with the ground truth (1) or other anchor boxes (0). Unlike Faster R-CNN [115], YOLOv3 associates a lone anchor box with each ground-truth object. In cases where no anchor box is associated, only the classification loss is incurred, excluding localisation and confidence losses.

Rather than utilising SoftMax for classification, they opted for binary cross-entropy, enabling the assignment of multiple labels to a single box. They also introduced a more extensive feature extractor consisting of 53 convolutional layers integrated with residual connections. This architecture was referred to as Darknet-53, which involved substituting all max-pooling layers with stride convolutions and integrating residual connections. Comprising 53 convolutional layers, this backbone architecture emerged at a point when the primary benchmark for OD transitioned from PASCAL VOC [116] to Microsoft COCO [113]. Consequently, all subsequent YOLO models were evaluated using the MS COCO dataset.

K-means [113] was utilised to determine eight prior boxes distributed across the three scale feature maps. Notably, larger-scale feature maps incorporated progressively smaller prior boxes. Further enhancements included a modified spatial pyramid pooling (SPP) block within the backbone to accommodate a broader receptive field. In YOLOv3, feature maps are structured with three scales, (416 × 416), (13 × 13), (26 × 26), and (52 × 52), for input, with three prior boxes for each position, as shown in Figure 6. These improvements collectively led to a 2.7% enhancement in the AP-50 metric. YOLOv3 achieved notable results: an AP of 36.2% and an AP-50 of 60.6% at a processing speed of 20 FPS, surpassing the pace of previous SOTA models.

Table 3. YOLOv3 architecture [117].

Layer	Filters	Size	Repeats	Output Size
Image	—	—	—	416 × 416
Conv	32	3 × 3 / 1	1	416 × 416
Conv	64	3 × 3 / 2	1	208 × 208
Conv	32	1 × 1 / 1	Conv × 1	208 × 208
Conv	64	3 × 3 / 1	Conv × 1	208 × 208
Residual	—	—	Residual × 1	208 × 208
Conv	128	3 × 3 / 2	1	104 × 104
Conv	64	1 × 1 / 1	Conv × 2	104 × 104
Conv	128	3 × 3 / 1	Conv × 2	104 × 104
Residual	—	—	Residual × 2	104 × 104
Conv	256	3 × 3 / 2	1	52 × 52
Conv	128	1 × 1 / 1	Conv × 8	52 × 52
Conv	256	3 × 3 / 1	Conv × 8	52 × 52
Residual	—	—	Residual × 8	52 × 52
Conv	512	3 × 3 / 2	1	26 × 26
Conv	256	1 × 1 / 1	Conv × 8	26 × 26
Conv	512	3 × 3 / 1	Conv × 8	26 × 26
Residual	—	—	Residual × 8	26 × 26
Conv	1024	3 × 3 / 2	1	13 × 13
Conv	512	1 × 1 / 1	Conv × 4	13 × 13
Conv	1024	3 × 3 / 1	Conv × 4	13 × 13
Residual	—	—	Residual × 4	13 × 13

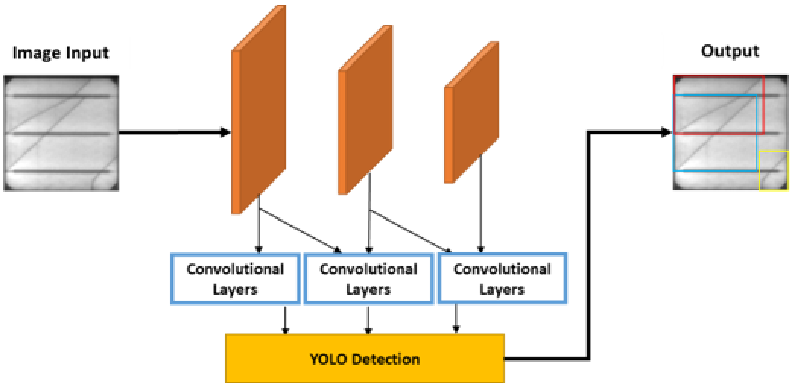

Figure 6. YOLOv3 architecture.

5.4. YOLOV4

In April 2020, the work by Alexey Bochkovskiy and colleagues [118] unveiled YOLOv4. Despite a change in authorship, the new researchers adhered to the foundational principles set by its predecessor, aiming for both high accuracy and real-time performance.

YOLOv4 marks a substantial departure from its previous iterations, introducing radical architectural transformations that yield remarkable performance enhancements. This version amalgamates several key components, including the CSP Darknet53 SPP structure [119], PANet architecture [120], CBN integration [121], and SAM incorporation [122]. The result is an efficient and robust OD model that excels in both speed and accuracy.

The Complete Intersection over Union (CIoU) loss is utilised in YOLOv4 and subsequent variants to optimise the localisation accuracy by considering factors such as IoU, maximum IoU, and regularisation, collectively facilitating the refinement of bounding box predictions. This loss function enhances the ability of YOLOv4 to precisely locate and delineate objects in the images, contributing to improved object detection performance. The formula is demonstrated in Equation (11):

$$\begin{aligned} L_{CIoU} = 1 - \mathrm{IoU}(b, \hat{b}) + \frac{(\rho^2 - \mathrm{IoU}(b, \hat{b})^2)}{\rho^2} \\ + \alpha \cdot \frac{v}{(1 - \mathrm{IoU}(b, \hat{b}) + v)} \end{aligned} \tag{11}$$

where L_{CIoU} is the CIoU loss, b represents the predicted bounding box, $\hat{b}$ is the ground-truth bounding box, $IoU(b, \hat{b})$ calculates the Intersection over Union (IoU) between the predicted and ground-truth boxes, ρ^2 is a parameter for the maximum possible IoU, α is a balancing factor, and V is used to account for small bounding boxes.

The YOLOv4 model prioritises ease of use and accessibility during the training process, catering to individuals with diverse technical backgrounds. The study also validated the effectiveness of contemporary SOTA methodologies, encompassing the bag-of-freebies (BoF) and bag-of-specials (BoS) techniques, to enhance the efficiency of the training pipeline. BoF techniques enhance model performance without incurring additional computational burden during inference, at the cost of longer training times. Conversely, BoS methods introduce modest inference time overheads but yield significant gains in detection accuracy. These methods are summarised in Table 4 [123].

Distinguishing itself from other target detection frameworks, the YOLOv4 architecture segments the model into distinct input, backbone, neck, and head components. Unlike YOLOv3, where a single anchor point was responsible for detecting the ground truth, YOLOv4 implements multiple anchor points for a single ground-truth detection. This approach elevates the selection ratio of positive samples and mitigates the imbalance between positive and negative samples. Furthermore, it eradicates grid sensitivity issues, thereby elevating boundary detection accuracy.

Table 4. YOLOv4 bag-of-freebies and bag-of-specials comparison.

	Backbone	Detector
Bag-of-specials	• Multi-input weighted residual connections • Cross-stage partial connections • Mish activation	• Distance-IoU non-maximum suppression • Spatial attention module (SAM) • Mish activation • Spatial pyramid pooling block • Path aggregation network (PAN)
Bag-of-freebies	• Class label smoothing • Data augmentation (mosaic, CutMix) • Regularisation (DropBlock)	• Cross mini-batch normalisation (CmBN) • Data augmentation (mosaic, Self-Adversarial Training) • Multiple anchors for single ground truth • Elimination of grid sensitivity • Cosine annealing scheduler • Random training shapes • Optimal hyperparameters • CIoU loss

5.5. YOLOv5

In the year 2020, Glenn Jocher introduced YOLOv5, shortly following the release of YOLOv4 [121]. YOLOv5, managed by Ultralytics, diverged from its predecessor, YOLOv4, in several ways. Notably, YOLOv5 embraced PyTorch instead of Darknet for its development, a strategic move that broadened its appeal to a wider user base due to PyTorch's user-friendly characteristics. This framework shift leveraged PyTorch's intuitive nature to facilitate greater adoption.

Several advancements contribute to the heightened efficacy of YOLOv5 in OD tasks. At its core, YOLOv5 features a Cross-Stage Partial (CSP) Net, a derivative of the ResNet architecture. This incorporates a CSP connection, elevating network efficiency and computational reduction. The CSPNet is augmented by multiple spatial pyramid pooling (SPP) blocks, which facilitate feature extraction at varying scales.

The architecture's neck incorporates a path aggregation network (PAN) module, as well as subsequent upsampling layers to enhance feature map resolution [124]. The head of YOLOv5 employs a series of convolutional layers to predict bounding boxes and class labels. YOLOv5 operates through anchor-based predictions, associating each bounding box with a set of predetermined anchor boxes of specific shapes and sizes.

To compute the loss function, a combination of two distinct loss components is utilised. Binary cross-entropy is employed for the calculation of class and objectness losses, while Complete Intersection over Union (CIoU) is incorporated to assess loss related to localisation accuracy. The formula for determining the loss function is expressed in Equation (12):

$$\text{loss} = \lambda_1 \cdot L_{\text{cls}} + \lambda_2 \cdot L_{\text{obj}} + \lambda_3 \cdot L_{\text{loc}} \tag{12}$$

where L_{cls} represents the binary cross-entropy loss for class predictions, L_{obj} is the binary cross-entropy loss for objectness predictions, and L_{loc} is the CIoU loss for localisation. The λ values represent weighting factors for each loss component.

The overarching aim of the YOLOv5 architecture is heightened efficiency and accuracy, surpassing previous iterations of YOLO. It introduces enhancements in feature extraction, feature aggregation, and anchor-based predictions. Moreover, it offers a smoother transition from PyTorch to ONNX and CoreML frameworks, compatible with IoS devices. This seamless integration empowers developers to incorporate YOLOv5 into their mobile applications without extensive modifications or additional frameworks.

When subjected to evaluation on the MS COCO dataset's test-dev 2017 split, YOLOv5x attained an AP score of 50.7% using a 640-pixel image size. Impressively, the model demonstrated a rapid processing speed, achieving 200 FPS with a batch size of 32 on an NVIDIA V100. When assessed with a larger input size of 1536 pixels, YOLOv5 achieved an even higher AP score of 55.8%. This attests to the model's ability to accurately detect objects, even at higher resolutions. The variant comparison of YOLOv5 is presented in Table 5.

Table 5. YOLOv5 variant comparison [124].

Model	AP@50	Parameters	FLOPs
YOLO-v5s	55.8%	7.5 M	13.2 B
YOLO-v5m	62.4%	21.8 M	39.4 B
YOLO-v5l	65.4%	47.8 M	88.1 B
YOLO-v5x	66.9%	89.0 M	166.4 B

5.6. YOLOv6

In September 2022, the Meituan Vision AI Department unveiled YOLOv6, releasing a variety of adaptations of the core architecture that were notably tailored to suit industrial deployment scenarios. This version showcased substantial advancements and refinements within its architecture. A significant development was the introduction of CSPDarknet as the new backbone architecture, surpassing the efficiency and speed benchmarks set by its predecessors YOLOv4 and YOLOv5.

A particularly noteworthy enhancement in YOLOv6 lay in its incorporation of a feature pyramid network (FPN) [125]. This addition led to the integration of a broader spectrum of feature scales, resulting in a tangible enhancement in detection accuracy, underscoring the commitment to augmenting performance. Furthermore, YOLOv6 was designed for optimal performance in real-time OD scenarios, exhibiting impressive frame rates on both central processing units (CPUs) and graphics processing units (GPUs).

A pivotal evolution in the YOLOv6 architecture involved the decoupling of the classification and box regression heads. This strategic architectural revision introduced supplementary layers within the network, effectively segregating these pivotal functions from the final head [126]. Empirical evidence substantiated this refinement's impact on elevating the overall model's performance, fortifying its capabilities [127].

Collectively, YOLOv6 represents a significant leap forward in the evolution of YOLO architectures, encompassing a comprehensive spectrum of improvements spanning speed, accuracy, and operational efficiency. Rigorous evaluation on the MS COCO dataset's test-dev 2017 subset showcased the prowess of the YOLOv6L model, yielding an AP of 52.5% and an AP-50 of 70%. Impressively, this commendable performance was achieved while maintaining a processing speed of approximately 50 FPS on an NVIDIA Tesla T4 GPU.

YOLOv6 is presented in three distinct variants, which are outlined in Table 6. Among them, YOLOv6nano stands out as the smallest and fastest alternative, boasting a minimal parameter count. This characteristic renders it particularly suitable for real-time OD tasks on devices with limited computational capabilities. Moving upstream, instances necessitating greater accuracy and the identification of smaller objects could lead to a preference for YOLOv6tiny or YOLOv6small. The choice of which variant to employ hinges on the unique use case, desired accuracy threshold, and available computational resources.

Table 6. YOLOv6 variant comparison [125].

Model	Size (Pixels)	mAP@50	Parameters	FLOPs
YOLO-v6-nano	416–640	30.8–35.0%	4.3 M	4.7–11.1 G
YOLO-v6-tiny	640	41.3%	15 M	36.7 G
YOLO-v6-small	640	43.1%	17.2 M	44.2 G

5.7. YOLOv7

Published in July 2022, YOLOv7 [128] emerged as a significant advancement over its predecessors, exhibiting heightened accuracy and speed improvements ranging from 5 FPS to 160 FPS. The focus of these enhancements revolved around bolstering efficiency and scalability, driven by the integration of the Extended Efficient Layer Aggregation Network (E-ELAN) [129] and a scalable approach for concatenation-based architectures. E-ELAN plays a crucial role in controlling the gradient path, thereby enhancing model learning and convergence. This technique is versatile, applicable to models with stacked computational blocks, and adeptly shuffles and merges features from distinct groups while maintaining the integrity of the gradient path.

Model scaling constitutes another pivotal component in YOLOv7, facilitating the creation of models of varying sizes. The devised scaling strategy adjusts the depth and width of the blocks by a uniform factor. This approach preserves the optimal model structure while mitigating hardware resource consumption.

The integration of various techniques collectively referred to as "bag-of-freebies" further amplifies the YOLOv7 model's performance. One such technique mirrors the re-parameterised convolution concept employed in YOLOv6. However, the RepConvN approach was introduced in YOLOv7 due to issues identified with the identity connection in RepConv [130] and concatenation in DenseNet [131]. Additionally, coarse label assignment is employed for the auxiliary head, while fine label assignment is reserved for the lead head. The auxiliary head contributes to the training process, while the lead head yields the final output. Furthermore, batch normalisation is harnessed, amalgamating the mean and variance of batch normalisation into the convolutional layer's bias and weight during inference, ultimately enhancing model performance [132].

In a rigorous evaluation on the MS COCO dataset's test-dev 2017, YOLOv7E6 garnered remarkable results, achieving an AP of 55.9% and an AP for an IoU threshold of 0.5 (AP-50) of 73.5%, as eloquently demonstrated in Table 7.

Table 7. YOLOv7 variant comparison [133].

Model	Size (Pixels)	mAP@50	Parameters	FLOPs
YOLO-v7 tiny	640	52.8%	6.2 M	5.8 G
YOLO-v7	640	69.7%	36.9 M	104.7 G
YOLO-v7X	640	71.1%	71.3 M	189.9 G
YOLO-v7E6	1280	73.5%	97.2 M	515.2 G
YOLO-v7D6	1280	73.8%	154.7 M	806.8 G

5.8. YOLOv8

In January 2023, Ultralytics unveiled YOLOv8, marked by its introduction in the field of CV [134]. Demonstrating an impressive degree of precision, the YOLOv8 model's performance was gauged through evaluations on both COCO and Roboflow 100 datasets [134]. What sets YOLOv8 apart is its user-oriented features, such as a user-friendly command-line interface and a well-structured Python package. The expanding and supportive YOLO community offers substantial resources for those engaging with the model.

The innovation within YOLOv8, as outlined in its approach [135], is its deviation from conventional anchor-based methods. Instead of relying on predetermined anchor boxes, YOLOv8 employs an anchor-free approach by predicting the object's centre. This adjustment addresses the challenge posed by anchor boxes that might not accurately represent custom dataset distributions. The benefits of this approach include a reduction in the number of box predictions and an acceleration of the post-processing step involving non-maximum suppression. Notably, the training routine of YOLOv8, encompassing techniques like online image augmentation, including mosaic augmentation, enhances the model's aptitude for detecting objects across diverse conditions and novel spatial arrangements.

In its architectural evolution from its predecessor, YOLOv5 (likewise authored by the same individuals), YOLOv8 introduces changes across its components. For instance, in the neck segment, YOLOv8 directly concatenates features without enforcing uniform channel dimensions. This strategy contributes to a reduction in the parameter count and overall tensor size.

When tested on the MS COCO dataset's test-dev 2017 subset, YOLOv8x delivered an AP of 53.9% at an image size of 640 pixels, compared to YOLOv5's AP of 50.7% with the same input size. Furthermore, YOLOv8x exhibited remarkable processing speed, achieving 280 FPS using an NVIDIA A100 with TensorRT. Notably, YOLOv8 is available in a range of five distinct variants, each tailored to specific accuracy and computational requisites, as showcased in Table 8.

Table 8. YOLOv8 variant comparison [134].

Model	Size (Pixels)	mAP@50	Parameters	FLOPs
YOLO-v8n	640	37.3%	3.2 M	8.7 G
YOLO-v8s	640	44.9%	11.2 M	28.6 G
YOLO-v8m	640	50.2%	25.9 M	78.9 G
YOLO-v8l	640	52.9%	43.7 M	165.2 G
YOLO-v8x	640	53.9%	68.2 M	257.8 G

5.9. YOLOv9

Wang et al. [136] introduced YOLOv9 in February 2024, which is the newest iteration of the YOLO object detection model family. YOLOv9 boasts two key innovations: the programmable gradient information (PGI) framework and the generalised efficient layer aggregation network (GELAN).

The PGI framework addresses the inherent information bottleneck problem in deep neural networks while also facilitating the compatibility of deep supervision mechanisms with lightweight architectures. By incorporating PGI, both lightweight and deep architectures can achieve significant performance improvements in terms of accuracy. This is attributed to PGI's ability to ensure reliable gradient information propagation during training, thereby enhancing the learning capacity and prediction accuracy of these architectures.

The proposed GELAN builds upon the gradient path optimisation principles of both the CSPNet [137] and ELAN [138] neural network architectures. This novel architecture prioritises a balance between model lightweightness, inference speed, and accuracy. The detailed architecture of the GELAN is depicted in Figure 7. This intentional design choice enables the GELAN to consistently deliver high performance across diverse computational blocks and depth configurations. Consequently, the GELAN demonstrates its versatility and applicability for deployment on a wide range of inference devices, including resource-constrained edge devices.

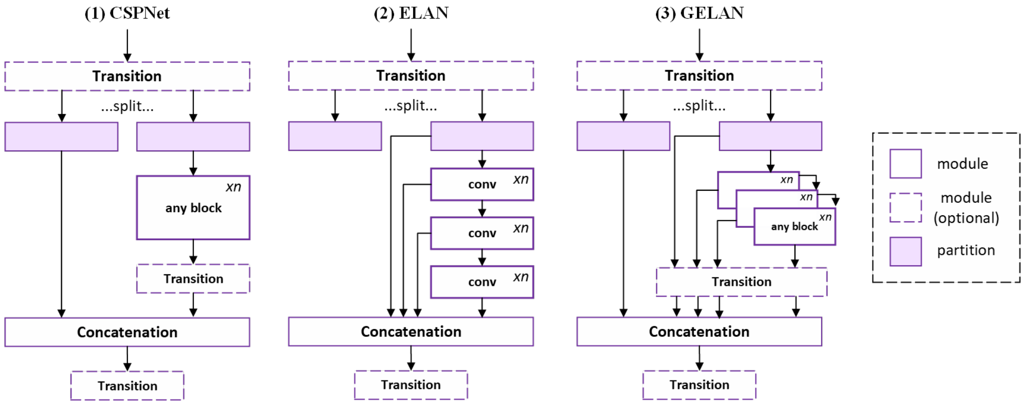

Figure 7. GELAN architecture [136].

Building upon the strengths of the PGI and GELAN frameworks, YOLOv9 represents a noteworthy stride forward in the domain of lightweight object detection. Despite its nascent stage of development, YOLOv9 exhibits remarkable competitiveness, surpassing YOLOv8 in terms of parameter reduction and computational efficiency, while achieving a noteworthy improvement of 0.6% in AP on the MS COCO dataset. The performance metrics of different YOLOv9 models are depicted in Table 9 [139].

Table 9. Performance metrics of YOLOv9 models [136].

Model	Size (Pixels)	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	Param.	FLOPs
YOLOv9-S	640	46.8%	63.4%	50.7%	7.2 M	26.7 G
YOLOv9-M	640	51.4%	68.1%	56.1%	20.1 M	76.8 G
YOLOv9-C	640	53.0%	70.2%	57.8%	25.5 M	102.8 G
YOLOv9-E	640	55.6%	72.8%	60.6%	58.1 M	192.5 G

5.10. YOLOv10

YOLOv10, developed by researchers at Tsinghua University and released in May 2024 [140], represents a significant advancement in the field of real-time OD. This novel architecture addresses a critical challenge in OD: balancing accuracy with computational efficiency. YOLOv10 achieves this through a combination of innovative training strategies, architectural modifications, and a range of model variants.

YOLOv10 tackles both accuracy and efficiency through a combination of training strategies and architectural innovations. The core concept lies in "Consistent Dual Assignments" during training, allowing the model to learn from rich supervision while eliminating the need for computationally expensive non-maximum suppression (NMS) during inference. As depicted in Figure 8, it significantly reduces processing time. YOLOv10 further enhances efficiency with the Parallel Split-Attention (PSA) module and the Compact Inverted Bottleneck (CIB) block, enabling efficient multi-scale feature processing and effective attention mechanisms. Finally, to boost accuracy, the Scaled Residual Connection and Scaled Weight

Shortcut techniques improve information flow within the network, leading to superior object detection performance.

Extensive evaluations demonstrate that YOLOv10 surpasses previous YOLO versions and other SOTA models in terms of the accuracy–efficiency trade-off. For instance, YOLOv10-S achieves faster processing speeds compared to RT-DETR-R18 while maintaining similar accuracy. Similarly, YOLOv10-B offers significant reductions in latency and parameter count compared to YOLOv9-C at equivalent performance levels. Moreover, YOLOv10-L and YOLOv10-X variants outperform their YOLOv8 counterparts in terms of accuracy while requiring fewer parameters. Table 10 summarizes the performance metrics of YOLOv10 models.

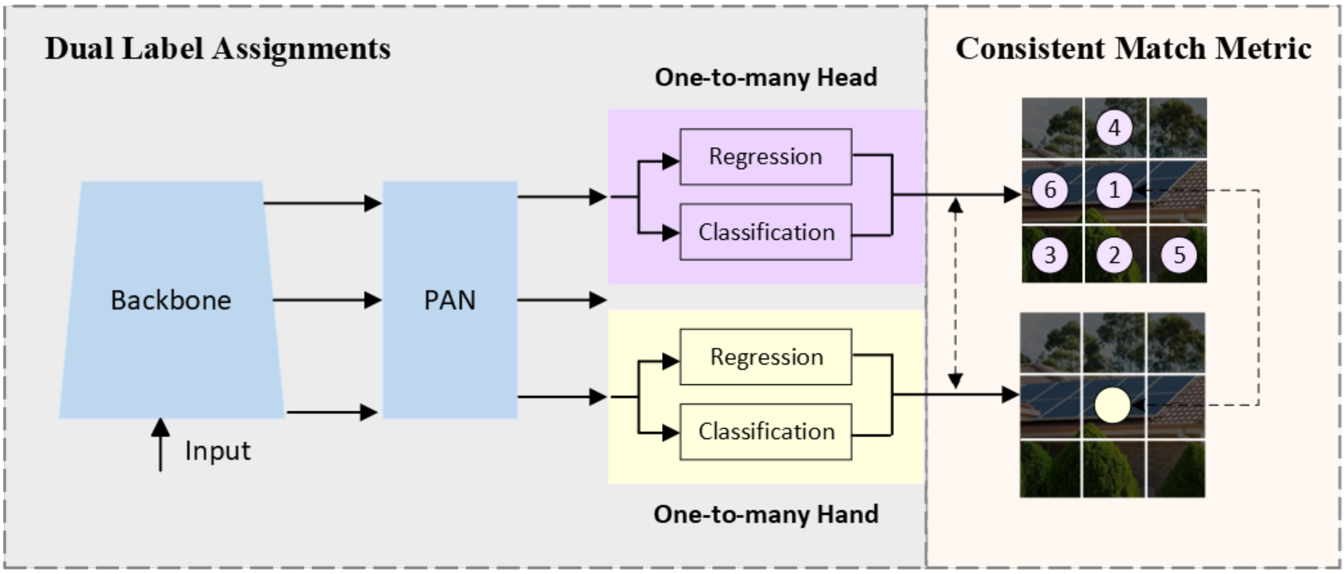

Figure 8. Consistent dual assignments.

Table 10. Performance metrics of YOLOv10 models [141].

Model	Size (Pixels)	APval (%)	FLOPs (G)	Latency (ms)
YOLOv10-N	640	38.5	6.7	1.84
YOLOv10-S	640	46.3	21.6	2.49
YOLOv10-M	640	51.1	59.1	4.74
YOLOv10-B	640	52.5	92.0	5.74
YOLOv10-L	640	53.2	120.3	7.28
YOLOv10-X	640	54.4	160.4	10.70

5.11. Model Comparison

It is evident that the YOLO series of object detectors has undergone several iterations, each contributing to the SOTA in CV. YOLOv1 (2015) laid the foundation for single-stage OD with the Darknet framework. Subsequent versions, including YOLOv2 and v3, introduced innovations such as anchor boxes, batch normalisation, and feature pyramid networks within the Darknet framework. YOLOv4 (2020) brought improvements in the form of the Mish activation function and CSPDarknet-53 backbone. YOLOv5 (2020) transitioned to the PyTorch framework and introduced anchor-free detection, SWISH activation, and PANet. YOLOv6, v7, and v8 (2022–2023) expanded on these innovations, incorporating self-attention, transformers, E-ELAN reparameterisation, and Generative Adversarial Networks (GANs), maintaining a PyTorch-based approach. YOLOv9 distinguishes itself through the integration of the PGI framework and the GELAN, contributing to improved OD performance. YOLOv10 continues to improve real-time OD by bringing in breakthrough innovations like consistent dual assignments for NMS-free training, along with efficiency-driven modules (PSA, CIB) and accuracy-enhancing techniques like Scaled Residual Connections and the Scaled Weight Shortcut. The key milestones of these diverse YOLO versions are summarised in Table 11.

Table 11. Summary of YOLO versions.

Version	Year	Contributions	Framework
v1	2015	Single-stage object detector	Darknet
v2	2016	Multi-scale training, dimension clustering	Darknet
v3	2018	SPP block, Darknet-53 backbone	Darknet
v4	2020	Mish activation, CSPDarknet-53 backbone	Darknet
v5	2020	Anchor-free detection, SWISH activation, PANet	PyTorch
v6	2022	Self-attention, anchor-free OD	PyTorch
v7	2022	Transformers, E-ELAN reparameterisation	PyTorch
v8	2023	GANs, anchor-free detection	PyTorch
v9	2024	PGI and GELAN	PyTorch
v10	2024	Consistent dual assignments for NMS-free training	PyTorch

6. PV Fault Detection via YOLO

This section presents a critical review of the existing literature focusing on the application of various YOLO variants for the detection of photovoltaic (PV) system faults.

The study conducted by N. Prajapati et al. [142] revolves around the utilisation of thermal images using a CNN learning algorithm, YOLO, for the detection and classification of faults in PV modules. The primary objective of the research was to discern shading [143] and bypass diode faults [144] in PV cells. The algorithm was implemented with the identification of four different types of faults: temporary hotspot fault, permanent hotspot fault, bypass diode fault, and cracks/wear and tear. The study achieved a maximum mAP of 83.86% and an average training loss of 0.0453%. Notably, the authors emphasised the absence of underfitting in their dataset. However, it is worth noting that the precision for bypass diode faults was comparatively lower. The authors attributed this discrepancy to the infrequent occurrence of bypass faults within the dataset, posing a notable limitation. They suggested that a larger dataset featuring an increased number of faults would likely enhance overall precision and performance.

A study conducted by Tahmid Tajwar et al. [145] delved into the realm of hotspot detection within PV modules using YOLOv3 and infrared thermography (IRT). While multiple methods encompassing electrical characterisation, EL imaging, and IR imaging are available for hotspot detection [146], this study specifically opted for IR imaging due to its widespread recognition within the domain [147]. Three training iterations were conducted using datasets of 5, 10, and 14 images, respectively. The results demonstrate that the detector trained on the largest dataset (14 images) exhibited superior accuracy and identified the greatest number of hotspots. These findings suggest a positive correlation between the diversity of the training dataset and the detector's precision in identifying hotspots. They emphasised that the inclusion of a more diversified dataset would likely lead to enhancements in both the accuracy and the quality of hotspot detection results. This research contributes to the field of photovoltaic condition monitoring, offering insights for improving hotspot detection within PV modules.

Antonio Greco et al. [148] underscored the often-neglected aspect of PV panel detection and the absence of a comprehensive performance evaluation framework. To bridge this gap, the authors established a set of criteria that define an ideal detection algorithm. These criteria encompass quantitative accuracy, real-time operability, the ability to analyse thermal images without relying on calibrated RGB cameras, and a plug-and-play functionality that eliminates the need for plant-specific configurations. The authors opted for YOLOv3, as it fulfils these established criteria. The dataset employed in their study comprises thermal camera footage captured by unmanned aerial vehicles (UAVs), encompassing diverse PV plants. Building upon the dataset referenced in a prior work [149], this dataset comprises 18 videos containing 50,449 panels and 4939 instances of hotspots. Employing their YOLOv3-based approach, the authors significantly elevated the precision level to 92%, surpassing the 83% achieved in a previous study [150]. Furthermore, an impressive F-score of 91% was achieved for plant types not previously encountered in the dataset, thereby further validating the approach's efficacy. Remarkably, when presented with plant-specific imagery, the model attained an accuracy of 95%. This study represents a pioneering application of DL in the domain of PV panel detection, effectively demonstrating its superiority.

The work conducted by H. Wang et al. [151] proposed a cloud-edge technique that leverages the YOLOv3-tiny algorithm and employed transfer learning to detect defects within PV components. In their pursuit of enhancing the algorithm's proficiency in detecting small targets, the authors ingeniously integrated the stitching layer from the second detection scale, shallow feature information, and a residual module into a third prediction layer. This strategic inclusion of the residual module bolsters the depth and learning capacity of the network model, enabling more effective extraction of target features. The method's efficacy was substantiated through evaluation, which revealed remarkable recall and accuracy rates of 93.7% and 95.5%, respectively, for the identification of flaws in PV components. Impressively, the detection of a single panoramic image was accomplished within a mere 6.3 milliseconds, utilising only 64 MB of the model's memory. The implementation of cloud-edge learning led to a 66% increase in the training time of a local sample model, resulting in a commendable accuracy level of 99.78%. While the authors acknowledge the time-saving advantages of fine-tuning at the edge, they posit that the introduction of additional training data could substantially augment the efficacy of their approach.

A study conducted by A. D Tommaso et al. [152] focused on the automatic detection of faults in PV panels through the utilisation of drones in two PV plants situated in southern Italy. The inspection process was carried out employing a Sigma Ingegneria Efesto MKII drone equipped with a DJI A2 flight controller (DJI, Shenzhen, China) and gimbal system. Two cameras were affixed to the drone: one captured thermal infrared images (LWIR) and low-resolution visible spectrum (VIS-LR) images, while the other was responsible for high-resolution visible spectrum (VIS-HR) images. This study significantly stands out as one of the pioneering works that harness UAV inspections through the utilisation of a YOLOv3 model. The detection endeavours encompassed a diverse array of faults, including soiling, delamination, bird droppings, and the identification of risks posed by puddles post-rainfall. Encouragingly, the authors reported positive outcomes, achieving a remarkable accuracy of 98% (AP@0.5) in PV panel detection for both PV plants. Regarding hotspot detection using infrared thermal imagery, the study achieved noteworthy performance, boasting an AP@0.4 of 88.3% and an AP@0.5 of 66.9%. However, amidst this success, the authors did identify a positive bias in the prediction of soiling areas. Additionally, they recognised a necessity for future work to rectify defect localisation errors by substituting GPS in drones with GNSS-RTK receivers. Notwithstanding these observations, the study exhibits promising results in terms of reducing operation and maintenance costs for PV modules, thereby underscoring the potential impact of this innovative approach.

In a study conducted by A. Gerd Imenes et al. [153], the aim was to enhance the detection and classification of faults in PV modules through image processing using multi-wavelength composite images. The selection of YOLOv3 as the detection algorithm was underpinned by its favourable trade-off between computational cost and performance. The researchers devised a strategy involving the creation of three-layered composite images, which merged both visible and infrared images. This approach using composite images had previously demonstrated its efficacy in various applications, including shadow detection [154].

In their research, J.-T. Zou et al. [155] advocated for the deployment of a 5G-enabled drone equipped with a thermal camera for the inspection of PV panels. The study involved the utilisation of a quadcopter drone equipped with a FLIR DUO PRO R thermal camera, integrated into a three-axis gimbal. To facilitate their approach, the authors harnessed a combination of Python, OpenCV, and Darknet YOLOv4 while incorporating real-time GPS location tracking. The dataset employed encompasses 1000 thermal images, of which

641 instances showcase cell failures. The authors report that achieving an mAP of 100% corresponds to an 89% confidence level, underscoring the precision of their proposed method.

Z. Meng et al. [156] introduced YOLO-PV, a solution rooted in the YOLOv4 framework, aimed at addressing the precision and speed constraints inherent to Electroluminescence (EL) image detection in PV modules. With the objective of countering the inconsistent detection standards observed in production lines and power stations, the authors categorised the defects into four distinct groups: material defects, cracks, scratches, and other anomalies. The YOLO-PV framework encompasses the quintessential components of object detection models, featuring a backbone, neck, and head architecture. Notably, the authors employ SPAN, a simplified rendition of PAN, depicted in Figure 9, that retains a single-size feature map to mitigate computational overhead. In their validation experiment, YOLO-PV attained an AP of 91.34%, marking a 0.64% enhancement over CSP-PV. This improvement holds significance, as YOLO-PV manages to simultaneously trim down output feature maps while bolstering AP. Furthermore, YOLO-PV succeeds in reducing processing speed by 36.36% compared to its YOLOv4 counterpart. In order to forestall overfitting, the authors judiciously incorporated techniques such as random rotation, the mosaic method, and random exposure adjustment. Ultimately, YOLO-PV's test performance culminated in an impressive accuracy of 94.55%.

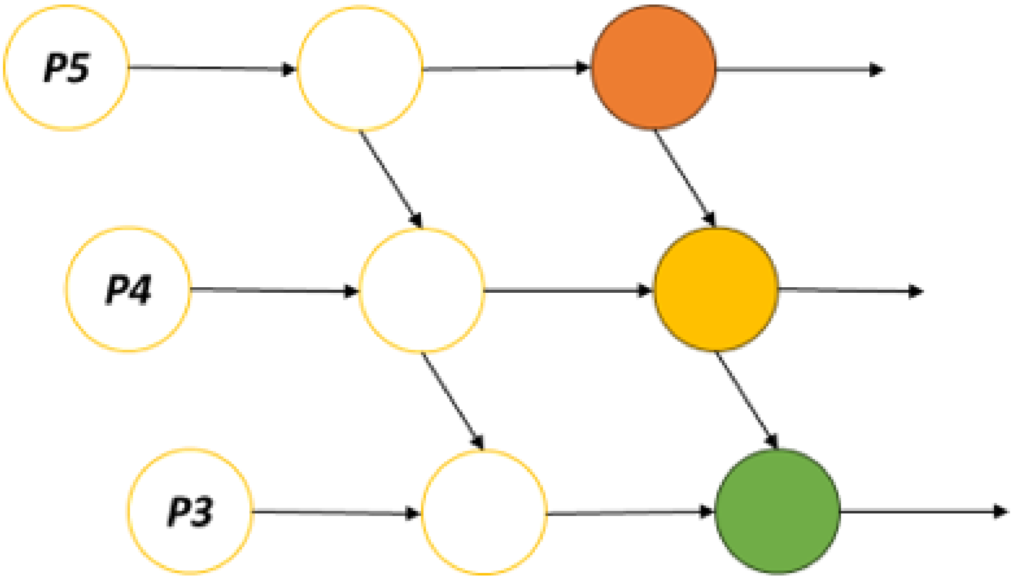

Figure 9. PANet configuration.

L.Li et al. [157] utilised YOLOv5 for the detection of defects on PV panels. The researchers endeavoured to enhance the YOLOv5 architecture by incorporating a BottleneckCSP module to amplify detection accuracy. They further adopted Ghost Convolution in lieu of traditional convolution. Addressing the challenge of detecting tiny targets, an additional prediction head was introduced to handle scale variations and mitigate misidentifications of small targets. For classification after feature extraction, the FPN and PAN were harnessed. This innovative approach, christened GBH-YOLOv5, is supported by a new dataset named PV-Multi-Defect, accessible for specialised applications in this domain. The dataset encompasses 1108 images featuring five defect types, with 886 images earmarked for training and 222 images for validation. Notably, GBH-YOLOv5 achieved an impressive mAP of 97.8 ± 0.02, outperforming five other major models. Significantly, it demonstrated a noteworthy 27.8% enhancement in mAP when compared to Fast R-CNN [87].

F. Hong et al. [158] presented a distinctive framework for PV fault detection, leveraging YOLOv5 and ResNet. Their approach involves the fusion of visible and infrared images captured from the same angle and altitude. The study underscores the significance of maintaining a low flight altitude for effective PV-array image capture. The proposed framework is structured into four stages: image acquisition, image segmentation, defect detection, and defect warning display. The dataset employed is sourced from a single PV power station in China's Hainan Province, comprising 3000 original images taken at various times of the day, distributed in a ratio of 22:7:1 for training, testing, and validation, respectively. The defect detection model attains an accuracy score of 95%, surpassing VGG's performance of 93%. Evidently, the proposed framework exhibits promising potential in PV fault detection, exhibiting superior accuracy compared to existing models.

M. Zhang et al. [159] undertook a study to enhance the fault detection approach for PV modules using YOLOv5. The authors introduced deformable convolutions, replacing certain traditional convolutions in the CSP module. This alteration facilitated the extraction of features of diverse sizes and shapes, augmenting the model's aptitude to detect various defect types. Additionally, the authors integrated the neck with ECA-Net and expanded the prediction heads to four, empowering shallow features to effectively identify small defects. To enhance network training efficiency, the authors utilised k-means++ for anchor box clustering, expediting convergence. The loss function was replaced with CIOU to elevate prediction box accuracy. The improved approach achieved an mAP of 89.64%, marking a substantial 7.85% improvement over the original model.

Q. Zheng [160] introduced S-YOLOv5, an enhancement of YOLOv5, tailored for the detection of PV panels and hotspots. This lightweight model incorporates adaptive scaling and normalisation for efficient feature extraction in the backbone, as well as the fusion of features in the neck and prediction stages. The loss function and gradient descent were harnessed to optimise the model's weights and biases, minimising loss values. A UAV equipped with a thermal camera captures images at varied resolutions, forming an aerial image dataset. ShuffleNetv2 and focus techniques constitute the backbone of S-YOLOv5. The input image is transformed into 12 channels through the focus operation. The proposed approach achieved an mAP of 98.1%, outperforming comparative object detection models like YOLOv5l (97.1%), YOLOc5x (96.5%), and YOLOv3 (96.4%) while maintaining a detection speed of 49 FPS. Importantly, the model boasts significantly fewer parameters than YOLOv5x, making it a lightweight yet potent choice.

In their study, X. Zhang et al. [161] introduced an innovative approach to detecting defects in PV panels by utilising thermal images captured by a DJI M300RTK UAV equipped with a DJI H20T thermal camera. The authors established a customised knowledge base encompassing a variety of PV panel defects. Employing the YOLOv5 algorithm, the researchers trained the model on a dataset of 10,772 IR images from eight PV plants. The model achieved an mAP of 80.88% at a confidence threshold of 0.5 during evaluation.

Q. B. Phan et al. [162] focused on PV fault detection using YOLOv8. To enhance the model's performance, Particle Swarm Optimisation (PSO) [163,164] was integrated to optimise essential parameters such as batch size, anchor box size, and learning rate. The dataset consisted of 2624 normalised and labelled solar panel cell images collected from various PV modules. The PSO algorithm improved YOLOv8's performance over YOLOv7's, achieving an mAP of 94% compared to 88% for YOLOv7.

The wide application of YOLO variants for PV fault detection is evident from Table 12, showcasing the diverse range of models and methodologies adopted for this critical task. What stands out is the consistently high level of accuracy achieved across different domains, including mAP, AP, and F1 score. Several YOLO variants, such as YOLOv4 and YOLOv5, consistently attain impressive results, with detection accuracies nearing 99% in certain cases. These remarkable outcomes underscore the robustness and adaptability of YOLO-based models in detecting faults within PV systems. Whether through the utilisation of drones, thermal imaging, or data augmentation techniques, YOLO variants demonstrate their effectiveness in addressing the challenges of PV fault detection. The ability to attain such high accuracy levels in diverse scenarios not only bolsters the reliability of PV systems but also offers promising implications for the broader application of YOLO variants in the field of CV and OD.

Table 12. PV fault detection models and results.

Ref.	Model	Characteristics	Results
[165]	PV-YOLO	PV-YOLO is combined with a transformer-based PVTv2 network to obtain edge details of faults.	92.56% mAP.
[142]	YOLO	Thermal images of PV modules on a learning algorithm using CNN based YOLO	83.86% mAP.
[145]	YOLOv3	IRT images of PV modules are utilized to identify hotspots of PV modules and used to validate the outcome of the detector.	More diversified data generates better precision and hotspot detection
[148]	YOLOv2 and YOLOv3	Detects PV panels in aerial imagery gathered from thermal cameras on board of UAVs using CNN based framework-YOLO	YOLOv2 = 89% F1 score. YOLOv3 = 91% F1 score.
[151]	YOLOv3-tiny	A cloud-based technique is used based on transfer learning to detect the fault.	95.5% accuracy.
[152]	YOLOv3	Model detects defects in PV using aerial images, validated on large PV plants in Italy, achieving high accuracy and efficiency in both thermographic and visible spectra.	98% AP @ 0.5.
[153]	YOLOv3	This paper evaluates the use of CNNs and multi-wavelength composite images for automating fault detection and classification in large-scale PV module installations, demonstrating successful fault detection but limited improvement in classification accuracy	75% mAP.
[166]	YOLOv3	YOLOv3 is used to detect the faulty region or hotspot of the PV while not considered as the best model. Faster R-CNN was chosen as the best OB model.	34% mAP.
[167]	YOLOv3-tiny	The proposal consists of using UAV equipped with a thermal camera and GPS with YOLOv3 to detect faults.	96.5% accuracy.
[155]	YOLOv4	A drone is used to inspect solar panels with the help of 5G and CV techniques.	100% mAP is achieved but with a confidence of 89%.
[156]	YOLOv4	The PAN network is used for feature fusion, which increases the model's performance.	94.55% AP.
[168]	YOLOv4, YOLOv4-tiny	In this paper, YOLOv4 and YOLOv4-tiny with spatial pyramid pooling is used with to solve PV fault detection.	YOLOv4 = 98.8% mAP. YOLOv4-tiny = 91.0% mAP.
[169]	YOLOv5	An EL image dataset of monocrystalline panels is used with the YOLOv5 DNN.	Appx.77% mAP.
[170]	YOLOv5	Developed YOLOv5s model enhanced with C3_cbam and SPP_eca units for accurate detection of cracks and fragments in PV modules from EL images	92.3% mAP.
[158]	YOLOv5	YOLOv5 is combined with ResNet to perform image segmentation and fault detection. The author claims that the model works perfectly in all bright conditions.	95% accuracy.
[159]	YOLOv5	This proposal uses mosaic and mix-up fusion data enhancement, K-means clustering, and the CIOU loss function to obtain optimised results.	89.64% mAP.
[160]	YOLOv5	Real-time hot-spot fault detection integrating a lightweight focus structure and ShuffleNetv2, suitable for deployment on UAV platforms.	98.1% mAP
[161]	YOLOv5	An AI-based UAV inspection and classification system using thermal imaging, YOLOV5 improves efficiency and safety in detecting defects in PV power plants	80.88% mAP @ 0.5.
[162]	YOLOv8	Particle Swarm Optimisation and YOLOv8 are combined to detect faults.	94% mAP.

7. Discussion

This review highlights the synergy between PV fault detection and the YOLO architecture. PV fault detection, particularly during manufacturing, presents a complex challenge due to stringent compliance requirements and the need for specific external conditions to ensure robust quality inspection. Several key insights emerge from this analysis, as described below.

7.1. Advantages of YOLO Methods in PV Inspection

7.1.1. Non-Invasive Inspection Demands

The PV manufacturing process comprises multiple stages, necessitating quality inspection as a sequential and gatekeeping process. Inspections must occur after the completion of a process but before the initiation of subsequent stages. Traditional sensor-based solutions, which entail integrating multiple sensors along production lines for data extraction, can be cost-prohibitive in terms of setup and maintenance. YOLO offers an appealing alternative by providing a non-invasive approach, eliminating the need for numerous sensor streams. This makes YOLO attractive for smaller or medium-sized enterprises seeking automated quality inspection without incurring high deployment and upkeep costs.

7.1.2. Single-Stage Detection Efficiency

While the advantage of non-invasive inference applies to various CNNs, YOLO possesses an additional advantage due to its architectural design centred on detection and classification through a single forward pass, unlike dual-stage detectors. This design feature grants YOLO a significant inference speed advantage over two-stage detectors, which is appealing to PV manufacturing facilities, as it reduces latency.

7.1.3. Real-Time Inference

YOLO's evolution is grounded in achieving rapid inference while maintaining high accuracy, an objective that aligns well with PV fault detection's requirements. This focus addresses the shortcomings of human-led inspection, namely, higher error rates and latency. As demonstrated in Table 12, researchers have achieved remarkable results in both accuracy and speed.

7.1.4. PyTorch Implementation

A significant driver of the widespread adoption of YOLO implementation is the transition from DarkNet to PyTorch, introduced in YOLOv5. The user-friendly nature of the PyTorch framework has enabled more researchers to explore, design, and develop YOLO architectures tailored to the PV defect detection domain. Despite YOLOv5 being introduced almost five years after the original YOLO, it has gained popularity as researchers' preferred choice due to its lightweight profile and, notably, its user-friendly development environment.

7.2. Limitations of YOLO Methods in PV Inspection

7.2.1. Need for Large Annotated Datasets

One of the main challenges in applying YOLO methods to PV fault detection is the need for large, annotated datasets to train the models effectively. Collecting and annotating PV fault data can be a time-consuming and resource-intensive process, requiring expertise in both PV manufacturing and computer vision. This can be a significant hurdle for smaller PV manufacturers or research groups with limited resources.

7.2.2. Detecting Subtle or Rare Defects

YOLO models may struggle with detecting subtle or rare defects in PV cells, especially if these defects are not well represented in the training data. Imbalanced datasets, where certain types of faults are more prevalent than others, can lead to biased models that fail to generalise well to real-world scenarios. Addressing this issue may require techniques such as data augmentation, class-weighting, or few-shot learning approaches.

7.2.3. Computational Requirements

While YOLO models are known for their real-time inference capabilities, they still require significant computational resources, especially when dealing with high-resolution PV imagery. The trade-off between model complexity, inference speed, and hardware requirements must be carefully considered when deploying YOLO models in PV manufacturing settings. Balancing these factors may necessitate the use of specialised hardware, such as GPUs or edge computing devices, which can increase implementation costs.

8. Conclusions and Future Scope

In conclusion, this review has delved into the extensive landscape of employing the YOLO architecture for PV fault detection. The synthesis of findings underscores the resounding suitability of YOLO as the optimal choice for addressing the distinctive demands of PV fault detection applications. This assertion gains strength through an in-depth examination of a multitude of research endeavours that have harnessed various YOLO variants to advance PV fault detection methodologies.

Notably, the architectural underpinnings of YOLO variants, with YOLOv5 exemplifying this trend, offer a compelling blend of architectural sophistication and inference efficiency. This synergy adeptly addresses the intrinsic challenges of human-centric inspection processes, characterised by inherent error rates and latency concerns. The seminal work by [160] stands as a testament to this symbiosis, showcasing an impressive 49 FPS inference speed coupled with a commendable 98.1% mAP.

Anticipating the trajectory of future research, a discernible trend emerges in which the focus will likely pivot toward refining the architectural landscape of YOLO variants for an even broader array of PV fault scenarios. While the current discourse has predominantly centred around micro-crack detection, the domain is ripe for expansion. In this vein, researchers are poised to delve deeper into the realm of attention mechanisms within the YOLO architecture. These attention mechanisms hold the potential to significantly enhance the detection process, particularly for subtle and intricate faults that require careful scrutiny.

By weaving attention mechanisms into YOLO's architecture, researchers aim to amplify the model's sensitivity to nuanced anomalies, thus enabling the identification of diverse fault manifestations with heightened precision. As the field progresses, the convergence of attention-driven architectures and YOLO variants is poised to chart new frontiers in the domain of PV fault detection.

Author Contributions: Data analysis, M.H. and R.K.; Original draft, M.H.; Review and editing, M.H. and R.K.; Project admin, M.H.; Investigation, M.H., and R.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Review article so no new data was generated.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hussain, M.; Chen, T.; Hill, R. Moving toward Smart Manufacturing with an Autonomous Pallet Racking Inspection System Based on MobileNetV2. *J. Manuf. Mater. Process.* **2022**, *6*, 75. [CrossRef]
2. Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Alsboui, T. Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections. *Sensors* **2022**, *22*, 6927. [CrossRef] [PubMed]
3. Hussain, M.; Hill, R. Custom lightweight convolutional neural network architecture for automated detection of damaged pallet racking in warehousing & distribution centers. *IEEE Access* **2023**, *11*, 58879–58889.
4. Hussain, M. YOLO-v5 Variant Selection Algorithm Coupled with Representative Augmentations for Modelling Production-Based Variance in Automated Lightweight Pallet Racking Inspection. *Big Data Cogn. Comput.* **2023**, *7*, 120. [CrossRef]
5. Talu, M.F.; Hanbay, K.; Varjovi, M.H. CNN-based fabric defect detection system on loom fabric inspection. *Text. Appar.* **2022**, *32*, 208–219. [CrossRef]
6. Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Parkinson, S. Exudate Regeneration for Automated Exudate Detection in Retinal Fundus Images. *IEEE Access* **2022**, *11*, 83934–83945. [CrossRef]
7. Ansari, M.A.; Crampton, A.; Parkinson, S. A Layer-Wise Surface Deformation Defect Detection by Convolutional Neural Networks in Laser Powder-Bed Fusion Images. *Materials* **2022**, *15*, 7166. [CrossRef] [PubMed]
8. Mehta, P.L.; Kumar, A. Livai: A Novel Resource-Efficient Real-Time Facial Emotion Recognition System Based on a Custom Deep Cnn Model. *SSRN Electron. J.* **2022**. [CrossRef]
9. Hussain, M. When, Where, and Which?: Navigating the Intersection of Computer Vision and Generative AI for Strategic Business Integration. *IEEE Access* **2023**, *11*, 127202–127215. [CrossRef]

10. Hussain, M.; Al-Aqrabi, H. Child Emotion Recognition via Custom Lightweight CNN Architecture. In *Kids Cybersecurity Using Computational Intelligence Techniques*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 165–174.
11. Aydin, B.A.; Hussain, M.; Hill, R.; Al-Aqrabi, H. Domain modelling for a lightweight convolutional network focused on automated exudate detection in retinal fundus images. In Proceedings of the 2023 9th International Conference on Information Technology Trends (ITT), Dubai, United Arab Emirates, 24–25 May 2023; IEEE: New York, NY, USA, 2023; pp. 145–150.
12. Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R. Feature mapping for rice leaf defect detection based on a custom convolutional architecture. *Foods* **2022**, *11*, 3914. [CrossRef]
13. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object Detection using YOLO: Challenges, Architectural Successors, Datasets and Applications. *Multimed. Tools Appl.* **2022**, *82*, 9243–9275. [CrossRef] [PubMed]
14. Hussain, M. YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO. *IEEE Access* **2024**, *12*, 42816–42833. [CrossRef]
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. Sultana, F.; Sufian, A.; Dutta, P. A Review of Object Detection Models Based on Convolutional Neural Network. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–16. [CrossRef]
17. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of YOLO Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
18. Ahmed, T.; Maaz, A.; Mahmood, D.; ul Abideen, Z.; Arshad, U.; Ali, R.H. The YOLOv8 Edge: Harnessing Custom Datasets for Superior Real-Time Detection. In Proceedings of the 2023 18th International Conference on Emerging Technologies (ICET), Xi'an, China, 6–7 November 2023; IEEE: New York, NY, USA, 2023; pp. 38–43.
19. Animashaun, D.; Hussain, M. Automated Micro-Crack Detection within Photovoltaic Manufacturing Facility via Ground Modelling for a Regularized Convolutional Network. *Sensors* **2023**, *23*, 6235. [CrossRef] [PubMed]
20. Zahid, A.; Hussain, M.; Hill, R.; Al-Aqrabi, H. Lightweight convolutional network for automated photovoltaic defect detection. In Proceedings of the 2023 9th International Conference on Information Technology Trends (ITT), Dubai, United Arab Emirates, 24–25 May 2023; IEEE: New York, NY, USA, 2023; pp. 133–138.
21. Hussain, M.; Al-Aqrabi, H.; Hill, R. Statistical Analysis and Development of an Ensemble-Based Machine Learning Model for Photovoltaic Fault Detection. *Energies* **2022**, *15*, 5492. [CrossRef]
22. Kabir, E.; Kumar, P.; Kumar, S.; Adelodun, A.A.; Kim, K.H. Solar Energy: Potential and Future Prospects. *Renew. Sustain. Energy Rev.* **2018**, *82*, 894–900. [CrossRef]
23. How Is electricity generated using solar? *National Grid ESO*. Available online: https://www.nationalgrideso.com/electricity-explained/how-electricity-generated/how-electricity-generated-using-solar (accessed on 17 May 2023).
24. Bagher, A.M. Types of Solar Cells and Application. *Am. J. Opt. Photonics* **2015**, *3*, 94. [CrossRef]
25. Inganäs, O.; Sundström, V. Solar Energy for Electricity and Fuels. *Ambio* **2015**, *45*, 15–23. [CrossRef] [PubMed]
26. Shaikh, M.R.S. A Review Paper on Electricity Generation from Solar Energy. 2017. Available online: http://hdl.handle.net/20.500.12323/4326 (accessed on 17 May 2023).
27. Sharma, S.; Jain, K.K.; Sharma, A. Solar Cells: In Research and Applications—A Review. *Mater. Sci. Appl.* **2015**, *06*, 1145–1155. [CrossRef]
28. Chu, Y.; Meisen, P. *Review and Comparison of Different Solar Energy Technologies*; Global Energy Network Institute: San Diego, CA, USA, 2011.
29. Choubey, P.C.; Oudhia, A.; Dewangan, R. A Review: Solar Cell Current Scenario and Future Trends. *Recent Res. Sci. Technol.* **2012**, *4*, 99–101.
30. Dhimsih, M.; Mather, P. Development of Novel Solar Cell Micro Crack Detection Technique. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 277–285. [CrossRef]
31. Liao, S.; Wang, J.; Yu, R.; Sato, K.; Cheng, Z. CNN for Situations Understanding Based on Sentiment Analysis of Twitter Data. *Procedia Comput. Sci.* **2017**, *111*, 376–381. [CrossRef]
32. Quang, D.; Xie, X. DanQ: A Hybrid Convolutional and Recurrent Deep Neural Network for Quantifying the Function of DNA Sequences. *Nucleic Acids Res.* **2016**, *44*, e107. [CrossRef] [PubMed]
33. Zhang, Y.; Tong, Y.; Jiang, Y. Study of Sentiment Classification for Chinese Microblog Based on Recurrent Neural Network. *Chinese J. Electron.* **2016**, *25*, 601–607. [CrossRef]
34. Sak, H.; Senior, A.; Rao, K.; Beaufays, F. Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition. *arXiv* **2015**, arXiv:1507.06947.
35. Zhang, X.Y.; Yin, F.; Zhang, Y.M.; Liu, C.L.; Bengio, Y. Drawing and Recognizing Chinese Characters with Recurrent Neural Network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 849–862. [CrossRef] [PubMed]
36. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29. [CrossRef]
37. Wei, D.; Wang, B.; Lin, G.; Liu, D.; Dong, Z.; Liu, H.; Liu, Y. Research on Unstructured Text Data Mining and Fault Classification Based on RNN-LSTM with Malfunction Inspection Report. *Energies* **2017**, *10*, 406. [CrossRef]
38. Mezaal, M.R.; Pradhan, B.; Sameen, M.I.; Shafri, H.Z.M.; Yusoff, Z.M. Optimized Neural Architecture for Automatic Landslide Detection from High-Resolution Airborne Laser Scanning Data. *Appl. Sci.* **2017**, *7*, 730. [CrossRef]

39. Kim, J.; Kim, J.; Thu, H.L.; Kim, H. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Republic of Korea, 15–17 February 2016. [CrossRef]
40. Rather, A.M.; Agarwal, A.; Sastry, V.N. Recurrent Neural Network and a Hybrid Model for Prediction of Stock Returns. *Expert Syst. Appl.* **2015**, *42*, 3234–3241. [CrossRef]
41. Xu, N.; Liu, A.A.; Wong, Y.; Zhang, Y.; Nie, W.; Su, Y.; Kankanhalli, M. Dual-Stream Recurrent Neural Network for Video Captioning. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2482–2493. [CrossRef]
42. Liang, M.; Hu, X. Recurrent Convolutional Neural Network for Object Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
43. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3929–3938.
44. Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; Oliva, A. Learning deep features for scene recognition using places database. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1–9.
45. Murphy-Chutorian, E.; Trivedi, M.M. Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 607–626. [CrossRef]
46. Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 724–732.
47. Brunetti, A.; Buongiorno, D.; Trotta, G.F.; Bevilacqua, V. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing* **2018**, *300*, 17–33. [CrossRef]
48. Medsker, L.; Jain, L.C. *Recurrent Neural Networks: Design and Applications*; CRC Press: Boca Raton, FL, USA, 1999.
49. Tarwani, K.M.; Edem, S. Survey on recurrent neural network in natural language processing. *Int. J. Eng. Trends Technol.* **2017**, *48*, 301–304. [CrossRef]
50. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [CrossRef]
51. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
52. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1998**, *6*, 107–116. [CrossRef]
53. Alaparthi, S.; Mishra, M. Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey. *arXiv* **2020**, arXiv:2007.01127.
54. Chavez, M.R.; Butler, T.S.; Rekawek, P.; Heo, H.; Kinzler, W.L. Chat Generative Pre-trained Transformer: Why we should embrace this technology. *Am. J. Obstet. Gynecol.* **2023**, *228*, 706–711. [CrossRef]
55. Kingma, D.P.; Welling, M. An introduction to variational autoencoders. *Found. Trends Mach. Learn.* **2019**, *12*, 307–392. [CrossRef]
56. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
57. Hijazi, S.; Kumar, R.; Rowen, C. *Using Convolutional Neural Networks for Image Recognition*; Cadence Design Systems Inc.: San Jose, CA, USA, 2015; Volume 9.
58. Liu, Q.; Zhang, N.; Yang, W.; Wang, S.; Cui, Z.; Chen, X.; Chen, L. A review of image recognition with deep convolutional neural network. In Proceedings of the Intelligent Computing Theories and Application: 13th International Conference, ICIC 2017, Liverpool, UK, 7–10 August 2017; Proceedings, Part I 13; Springer: Berlin/Heidelberg, Germany, 2017; pp. 69–80.
59. Chauhan, R.; Ghanshala, K.K.; Joshi, R. Convolutional neural network (CNN) for image detection and recognition. In Proceedings of the 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 15–17 December 2018; IEEE: New York, NY, USA, 2018; pp. 278–282.
60. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
61. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
62. Guo, T.; Dong, J.; Li, H.; Gao, Y. Simple convolutional neural network on image classification. In Proceedings of the 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, China, 10–12 March 2017; IEEE: New York, NY, USA, 2017; pp. 721–724.
63. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3156–3164.
64. Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 1915–1929. [CrossRef] [PubMed]
65. Krizhevsky, A.; Hinton, G.E. Using very deep autoencoders for content-based image retrieval. *ESANN* **2011**, *1*, 2.
66. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; Volume 3361, pp. 255–258.

67. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
68. Toshev, A.; Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660.
69. Sermanet, P.; LeCun, Y. Traffic sign recognition with multi-scale convolutional networks. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; IEEE: New York, NY, USA, 2011; pp. 2809–2813.
70. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**, arXiv:1508.06576.
71. Hubel, D.H.; Wiesel, T.N. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* **1959**, *148*, 574. [CrossRef] [PubMed]
72. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **1980**, *36*, 193–202. [CrossRef]
73. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* **1989**, *2*, 396–404.
74. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *60*, 84–90. [CrossRef]
75. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
76. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
77. Wu, H.; Gu, X. Towards dropout training for convolutional neural networks. *Neural Netw.* **2015**, *71*, 1–10. [CrossRef] [PubMed]
78. Taylor, L.; Nitschke, G. Improving deep learning with generic data augmentation. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bengaluru, India, 18–21 November 2018; IEEE: New York, NY, USA, 2018; pp. 1542–1547.
79. Chen, K.; Franko, K.; Sang, R. Structured Model Pruning of Convolutional Networks on Tensor Processing Units. *arXiv* **2021**, arXiv:2107.04191.
80. Bengio, Y.; Courville, A.; Vincent, P. Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives. 2012. Available online: https://api.semanticscholar.org/CorpusID:4493778 (accessed on 17 May 2023).
81. Ujjwal. An Intuitive Explanation of Convolutional Neural Networks. 2017. Available online: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ (accessed on 17 May 2023).
82. Krichen, M. Convolutional Neural Networks: A Survey. *Computers* **2023**, *12*, 151. [CrossRef]
83. Agarwal, S.; Terrail, J.O.D.; Jurie, F. Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks. 2019. Available online: https://api.semanticscholar.org/CorpusID:52183570 (accessed on 17 May 2023).
84. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int. J. Comput. Vis.* **2018**, *28*, 261–318. [CrossRef]
85. Hassan, M.; Hussain, F.; Khan, S.D.; Ullah, M.; Yamin, M.; Ullah, H. Crowd Counting Using Deep Learning Based Head Detection. *Electron. Imaging* **2023**, *35*, 293-1. [CrossRef]
86. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 3520–3529.
87. Girshick, R. Fast R-CNN. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1137–1149.
88. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
89. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
90. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
91. Sun, C.; Ai, Y.; Wang, S.; Zhang, W. Dense-RefineDet for traffic sign detection and classification. *Sensors* **2020**, *20*, 6570. [CrossRef]
92. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2017**, arXiv:1708.02002. [CrossRef]
93. Akram, M.W.; Li, G.; Jin, Y.; Chen, X.; Zhu, C.; Ahmad, A. Automatic Detection of Photovoltaic Module Defects in Infrared Images with Isolated and Develop-Model Transfer Deep Learning. *Sol. Energy* **2020**, *198*, 175–186. [CrossRef]
94. Mehta, S.; Azad, A.P.; Chemmengath, S.A.; Raykar, V.C.; Kalyanaraman, S. DeepSolarEye: Power Loss Prediction and Weakly Supervised Soiling Localization via Fully Convolutional Networks for Solar Panels. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018. [CrossRef]
95. Karimi, A.M.; Fada, J.S.; Hossain, M.A.; Yang, S.; Peshek, T.J.; Braid, J.L.; French, R.H. Automated Pipeline for Photovoltaic Module Electroluminescence Image Processing and Degradation Feature Classification. *IEEE J. Photovolt.* **2019**, *9*, 1324–1335. [CrossRef]
96. Deitsch, S.; Christlein, V.; Berger, S.; Buerhop-Lutz, C.; Maier, A.; Gallwitz, F.; Riess, C. Automatic Classification of Defective Photovoltaic Module Cells in Electroluminescence Images. *Sol. Energy* **2019**, *185*, 455–468. [CrossRef]

97. Zhao, Y.; Zhan, K.; Wang, Z.; Shen, W. Deep Learning-Based Automatic Detection of Multitype Defects in Photovoltaic Modules and Application in Real Production Line. *Prog. Photovolt. Res. Appl.* **2021**, *29*, 471–484. [CrossRef]
98. Ahmad, A.; Jin, Y.; Zhu, C.; Javed, I.; Maqsood, A.; Akram, M.W. Photovoltaic Cell Defect Classification Using Convolutional Neural Network and Support Vector Machine. *IET Renew. Power Gener.* **2020**, *14*, 2693–2702. [CrossRef]
99. Tang, W.; Yang, Q.; Xiong, K.; Yan, W. Deep Learning Based Automatic Defect Identification of Photovoltaic Module Using Electroluminescence Images. *Sol. Energy* **2020**, *201*, 453–460. [CrossRef]
100. Dunderdale, C.; Brettenny, W.; Clohessy, C.; Dyk, E.E. Photovoltaic Defect Classification Through Thermal Infrared Imaging Using a Machine Learning Approach. *Prog. Photovolt. Res. Appl.* **2019**, *28*, 177–188. [CrossRef]
101. Yap, X.Y.; Chia, K.S.; Tee, K.S. A Portable Gas Pressure Control and Data Acquisition System Using Regression Models. *Int. J. Electr. Eng. Inform.* **2021**, *13*, 242–251. [CrossRef]
102. Zhang, J.; Feng, Y. Advanced Chinese Character Detection for Natural Scene Based on EAST. *J. Phys. Conf. Ser.* **2020**, *1550*, 032050. [CrossRef]
103. Pierdicca, R.; Malinverni, E.S.; Piccinini, F.; Paolanti, M.; Felicetti, A.; Zingaretti, P. Deep Convolutional Neural Network for Automatic Detection of Damaged Photovoltaic Cells. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII–2*, 893–900. [CrossRef]
104. Chen, H.; Hu, Q.; Zhai, B.; Chen, H.; Liu, K. A Robust Weakly Supervised Learning of Deep Conv-Nets for Surface Defect Inspection. *Neural Comput. Appl.* **2020**, *32*, 11229–11244. [CrossRef]
105. Du, B.; He, Y.; He, Y.; Duan, J.; Zhang, Y. Intelligent Classification of Silicon Photovoltaic Cell Defects Based on Eddy Current Thermography and Convolution Neural Network. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6242–6251. [CrossRef]
106. Hussain, T.; Hussain, M.; Al-Aqrabi, H.; Alsboui, T.; Hill, R. A Review on Defect Detection of Electroluminescence-Based Photovoltaic Cell Surface Images Using Computer Vision. *Energies* **2023**, *16*, 4012. [CrossRef]
107. Demirci, M.Y.; Beşli, N.; Gümüşçü, A. Efficient Deep Feature Extraction and Classification for Identifying Defective Photovoltaic Module Cells in Electroluminescence Images. *Expert Syst. Appl.* **2021**, *175*, 114810. [CrossRef]
108. Kellil, N.; Aissat, A.; Mellit, A. Fault diagnosis of photovoltaic modules using deep neural networks and infrared images under Algerian climatic conditions. *Energy* **2023**, *263*, 125902. [CrossRef]
109. Amiri, A.F.; Kichou, S.; Oudira, H.; Chouder, A.; Silvestre, S. Fault detection and diagnosis of a photovoltaic system based on deep learning using the combination of a convolutional neural network (cnn) and bidirectional gated recurrent unit (Bi-GRU). *Sustainability* **2024**, *16*, 1012. [CrossRef]
110. Zhang, N.; Shan, S.; Wei, H.; Zhang, K. Micro-cracks Detection of Polycrystalline Solar Cells with Transfer Learning. *J. Phys. Conf. Ser.* **2020**, *1651*, 012118. [CrossRef]
111. Hussain, M.; Al-Aqrabi, H.; Hill, R. PV-CrackNet architecture for filter induced augmentation and micro-cracks detection within a photovoltaic manufacturing facility. *Energies* **2022**, *15*, 8667. [CrossRef]
112. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
113. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312.
114. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
115. Won, J.H.; Lee, D.H.; Lee, K.M.; Lin, C.H. An Improved YOLOv3-based Neural Network for De-identification Technology. In Proceedings of the 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Jeju, Republic of Korea, 23–26 June 2019.
116. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
117. Chakure, A. All You Need to Know about YOLO v3 (You Only Look Once). 2019. Available online: https://dev.to/afrozchakure/all-you-need-to-know-about-yolo-v3-you-only-look-once-e4m (accessed on 5 June 2024).
118. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020. Available online: https://arxiv.org/abs/2004.10934 (accessed on 17 May 2023)
119. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
120. Ma, Z.; Li, M.; Wang, Y. PAN: Path Integral Based Convolution for Deep Graph Neural Networks. 2019. Available online: https://arxiv.org/abs/1904.10996 (accessed on 17 May 2023).
121. Yao, Z.; Cao, Y.; Zheng, S.; Huang, G.; Lin, S. Cross-Iteration Batch Normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 11–17 October 2021.
122. He, S.; Bao, R.; Li, J.; Grant, P.E.; Ou, Y. Accuracy of Segment-Anything Model (SAM) in Medical Image Segmentation Tasks. 2023. Available online: https://api.semanticscholar.org/CorpusID:258212977 (accessed on 4 June 2024).
123. Terven, J.; Cordova-Esparza, D. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and beyond. 2023. Available online: https://arxiv.org/abs/2304.00501v1 (accessed on 4 June 2024).
124. Solawetz, J.; Roboflow Blog. What IS YOLOv5? A Guide for Beginners. 2020. Available online: https://blog.roboflow.com/yolov5-improvements-and-evaluation/ (accessed on 3 June 2024).

125. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
126. Wang, Z.; Chen, Z.; Li, Y.; Guo, Y.; Yu, J.; Gong, M.; Liu, T. Mosaic Representation Learning for Self-Supervised Visual Pre-Training. 2023. Available online: https://openreview.net/forum?id=JAezPMehaUu (accessed on 4 June 2024).
127. Solawetz, J.; Nelson, J. What Is YOLOv6? The Ultimate Guide. 2024. Available online: https://blog.roboflow.com/yolov6/ (accessed on 4 June 2024).
128. Xu, X.; Jiang, Y.; Chen, W.; Huang, Y.; Zhang, Y.; Sun, X. DAMO-YOLO: A Report on Real-Time Object Detection Design. *arXiv* **2022**, arXiv:2211.15444.
129. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-Style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021.
130. Huang, G.; Liu, Z.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2016**, arXiv:1608.06993. https://doi.org/10.48550/arXiv.1608.06993.
131. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. Available online: https://github.com/ultralytics/ultralytics (accessed on 30 February 2024).
132. Solawetz, J. What Is YOLOv7? A Complete Guide. 2022. Available online: https://blog.roboflow.com/yolov7-breakdown/ (accessed on 4 June 2024).
133. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
134. Solawetz, J.; Francesco. What Is YOLOv8? The Ultimate Guide. 2023. Available online: https://blog.roboflow.com/whats-new-in-yolov8/ (accessed on 4 June 2024).
135. Jocher, G.; Stoken, A.; Borovec, J.; Chaurasia, A.; Changyu, L.; Hogan, A.; Hajek, J.; Diaconu, L.; Kwon, Y.; Defretin, Y.; et al. *Ultralytics/Yolov5: v5.0-YOLOv5-P6 1280 Models, AWS, Supervise.ly and YouTube Integrations*; Zenodo: Genève, Switzerland, 2021. [CrossRef]
136. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
137. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
138. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing network design strategies through gradient path analysis. *arXiv* **2022**, arXiv:2211.04800.
139. Wong, K.Y. YOLOv9 GitHub Repository. Available online: https://github.com/WongKinYiu/yolov9 (accessed on 4 June 2024).
140. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
141. Ultralytics. YOLOv10: Real-Time End-to-End Object Detection. Available online: https://docs.ultralytics.com/models/yolov10/#model-variants (accessed on 3 June 2024).
142. Prajapati, N.; Aiyar, R.; Raj, A.; Paraye, M. Detection and Identification of Faults in a PV Module Using CNN Based Algorithm. In Proceedings of the 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 27–29 May 2022; pp. 1–5.
143. Salazar, A.M.; Macabebe, E.Q.B. Hotspots Detection in Photovoltaic Modules Using Infrared Thermography. *MATEC Web Conf.* **2016**, *70*, 10015. [CrossRef]
144. Shin, W.; Ko, S.; Song, H.; Ju, Y.; Hwang, H.; Kang, G. Origin of Bypass Diode Fault in c-Si Photovoltaic Modules: Leakage Current under High Surrounding Temperature. *Energies* **2018**, *11*, 2416. [CrossRef]
145. Tajwar, T.; Mobin, O.H.; Khan, F.R.; Hossain, S.F.; Islam, M.; Rahman, M.M. Infrared Thermography Based Hotspot Detection Of Photovoltaic Module using YOLO. In Proceedings of the 2021 IEEE 12th Energy Conversion Congress & Exposition-Asia (ECCE-Asia), Singapore, 11–14 May 2021; pp. 1542–1547.
146. Schuss, C.; Leppänen, K.; Saarela, J.; Fabritius, T.; Eichberger, B.; Rahkonen, T. Detecting defects in photovoltaic modules with the help of experimental verification and synchronized thermography. In Proceedings of the 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Pisa, Italy, 11–14 May 2015.
147. Haque, A.; Bharath, K.V.S.; Khan, M.A.; Khan, I.; Jaffery, Z.A. Fault diagnosis of Photovoltaic Modules. *Energy Sci. Eng.* **2019**, *7*, 622–644. [CrossRef]
148. Greco, A.; Pironti, C.; Vento, M.; Vigilante, V. A deep learning based approach for detecting panels in photovoltaic plants. In Proceedings of the 3rd International Conference on Applications of Intelligent Systems, Las Palmas de Gran Canaria, Spain, 7–12 January 2020. [CrossRef]
149. Shinde, S.; Kothari, A.; Gupta, V. YOLO based Human Action Recognition and Localization. *Procedia Comput. Sci.* **2018**, *133*, 831–838. [CrossRef]
150. Carletti, V.; Greco, A.; Saggese, A.; Vento, M. An intelligent flying system for automatic detection of faults in photovoltaic plants. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *11*, 2027–2040. [CrossRef]
151. Wang, H.; Li, F.; Mo, W.; Tao, P.; Shen, H.; Wu, Y.; Zhang, Y.; Deng, F. Novel Cloud-Edge Collaborative Detection Technique for Detecting Defects in PV Components, Based on Transfer Learning. *Energies* **2022**, *15*, 7924. [CrossRef]

152. Tommaso, A.D.; Betti, A.; Fontanelli, G.; Michelozzi, B. A multi-stage model based on YOLOv3 for defect detection in PV panels based on IR and visible imaging by unmanned aerial vehicle. *Renew. Energy* **2022**, *193*, 941–962. [CrossRef]
153. Imenes, A.G.; Noori, N.S.; Uthaug, O.A.N.; Kröni, R.; Bianchi, F.; Belbachir, N. A Deep Learning Approach for Automated Fault Detection on Solar Modules Using Image Composites. In Proceedings of the 2021 IEEE 48th Photovoltaic Specialists Conference (PVSC), Fort Lauderdale, FL, USA, 20–25 June 2021.
154. Teke, M.; Baseski, E.; Ok, A.O.; Yuksel, B.; Şenaras, Ç. Multi-spectral False Color Shadow Detection. In Proceedings of the ISPRS Conference on Photogrammetric Image Analysis, Munich, Germany, 5–7 October 2011. [CrossRef]
155. Zou, J.T.; Rajveer, G.V. Drone-Based Solar Panel Inspection with 5G and AI Technologies. In Proceedings of the 2022 8th International Conference on Applied System Innovation (ICASI), Nantou, Taiwan, 22–23 April 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 174–178.
156. Meng, Z.; Xu, S.; Wang, L.; Gong, Y.; Zhang, X.; Zhao, Y. Defect object detection algorithm for electroluminescence image defects of photovoltaic modules based on deep learning. *Energy Sci. Eng.* **2022**, *10*, 800–813. [CrossRef]
157. Li, L.; Wang, Z.; Zhang, T. Photovoltaic Panel Defect Detection Based on Ghost Convolution with BottleneckCSP and Tiny Target Prediction Head Incorporating YOLOv5. *arXiv* **2023**, arXiv:2303.00886.
158. Hong, F.; Song, J.; Meng, H.; Wang, R.; Fang, F.; Zhang, G. A novel framework on intelligent detection for module defects of PV plant combining the visible and infrared images. *Sol. Energy* **2022**, *236*, 406–416. [CrossRef]
159. Zhang, M.; Yin, L. Solar Cell Surface Defect Detection Based on Improved YOLO v5. *IEEE Access* **2022**, *10*, 80804–80815. [CrossRef]
160. Zheng, Q.; Ma, J.; Liu, M.; Liu, Y.; Li, Y.; Shi, G. Lightweight Hot-Spot Fault Detection Model of Photovoltaic Panels in UAV Remote-Sensing Image. *Sensors* **2022**, *22*, 4617. [CrossRef]
161. Zhang, X.; Zou, P.; Ma, C.; Zhang, Z.; Guo, H.; Chen, Y.; Cheng, Z. Inspection and Classification System of Photovoltaic Module Defects Based on UAV and Thermal Imaging. In Proceedings of the 2022 7th International Conference on Power and Renewable Energy (ICPRE), Shanghai, China, 23–26 September 2022.
162. Phan, Q.B.; Nguyen, T. A Novel Approach for PV Cell Fault Detection using YOLOv8 and Particle Swarm Optimization. In Proceedings of the 2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS), Tempe, AZ, USA, 6–9 August 2023. [CrossRef]
163. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 2019. [CrossRef]
164. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73. [CrossRef]
165. Yin, W.; Lingxin, S.; Maohuan, L.; Qianlai, S.; Xiaosong, L. PV-YOLO: Lightweight YOLO for Photovoltaic Panel Fault Detection. *IEEE Access* **2023**, *11*, 10966–10976. [CrossRef]
166. Pathak, S.P.; Patil, D.S.; Patel, S. Solar panel hotspot localization and fault classification using deep learning approach. *Procedia Comput. Sci.* **2022**, *204*, 698–705. [CrossRef]
167. Han, S.H.; Rahim, T.; Shin, S.Y. Detection of Faults in Solar Panels Using Deep Learning. In Proceedings of the 2021 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Republic of Korea, 31 January–3 February 2021. [CrossRef]
168. Binomairah, A.; Abdullah, A.; Khoo, B.E.; Mahdavipour, Z.; Teo, T.W.; Noor, N.S.M.; Abdullah, A.; Binomairah, M.Z. Detection of microcracks and dark spots in monocrystalline PERC cells using photoluminescene imaging and YO-LO-based CNN with spatial pyramid pooling. *EPJ Photovolt.* **2022**, *13*, 27. [CrossRef]
169. Rodriguez, A.R.; Holicza, B.; Nagy, A.M.; Vörösházi, Z.; Bereczky, G.; Czúni, L. Segmentation and Error Detection of PV Modules. In Proceedings of the 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, Germany, 6–9 September 2022.
170. Xu, S.; Qian, H.; Shen, W.; Wang, F.; Liu, X.; Xu, Z. Defect detection for PV Modules based on the improved YOLOv5s. In *2022 China Automation Congress (CAC)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1431–1436.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.