Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 05
May 2025

Severity Analysis of Leaf Disease Identification in Pear Using Deep Object Detection Methods

Tahani Alkhudaydi

Computer Science Department

University of Tabuk

Tabuk, 71491, Saudi Arabia

Abstract

Early-stage leaf diseases in plants can significantly impact crop yield and quality. If left untreated, these diseases can spread rapidly. This will lead to widespread damage and potential crop loss. Early detection allows for targeted disease control measures to minimise negative effects on plant health and optimise crop production. This study examines the effectiveness of various object detection methods for identifying early-stage leaf diseases, with a particular focus on pear leaf disease. Advanced machine learning algorithms, including R-CNN detectors and YOLO models, were employed to analyse plant leaf images. The YOLOv8s model emerged as the most effective with an mAP of 88.3. This may be attributed to its robust architecture and its ability to extract features effectively. The cascade-rcnn r50 fpn model demonstrated strong performance, due to the effectiveness of its multi-stage region proposal network and feature pyramid networks. YOLOv8s and cascade-rcnn r50 fpn showed promise in detecting mild disease symptoms, but further research is needed to cover a wider range of plant diseases and make them accessible for farmers to use.

Keywords: *Plant Disease Detection; Agriculture; Object Detection; Deep Learning*

1.INTRODUCTION

The detection of leaf diseases holds significant importance in the field of agriculture. The early detection of diseases in plants may play a crucial role in mitigating the spread of such diseases and reducing the extent of crop losses. Early diagnosis is key in agriculture, where disease control is essential for crop health and production. Early intervention plays a crucial role in the successful control of diseases, since it enables prompt and focused actions that have the potential to prevent significant crop loss [1].

There are several methods for detecting leaf diseases in plants. Visual inspection is the most common method, where farmers observe the plants for any signs of disease. However, this method can be time-consuming and not always reliable [2].

Another method is the use of digital imaging technology. With the help of machine learning algorithms and computer vision, images of the plant leaves can be analysed to detect any signs of disease. This method is more efficient and accurate compared to visual inspection [2].

The majority of existing studies in plant disease detection using deep learning are designed around classification tasks.

This has led to the creation of datasets that are primarily suited for classification algorithms. As a result, there's a notable shortage of datasets geared towards object detection methods in this field. Consequently, there exists a significant lack of datasets created specifically for use with to object detection methods within this domain. This limitation not only limits the use of advanced object detection techniques, but it also establishes the context for the current focus of our research.

Empowering farmers involves giving them intelligent, costeffective, and user-friendly machine learning tools. By using these technologies, agricultural practitioners have the capability to identify and diagnose plant diseases in their early stages via the use of straightforward devices such as cellphone images. The primary objective of this study is to evaluate the efficacy of object detection techniques , with a specific focus on their use in the early identification of leaf diseases. This focus is essential in enhancing agricultural productivity and sustainability across diverse farming practices.

To the best of our knowledge, this is the first study to extensively analyse early-stage leaf disease identification in pears using advanced object detection methods. Our research focuses on evaluating various aspects of multiple deep object detection models. The main contributions of our work are outlined as follows:

- A comprehensive review of leading deep learning architectures specifically adapted for plant disease detection and identification.
- A detailed evaluation of the performance of YOLOv8s, Cascade-rcnn _r50 fpn, and other advanced models in identifying early stages of pear leaf diseases, considering several performance metrics.

The rest of the paper is organised as follows: Section 2 provides a review of relevant literature; Section 3 outlines the materials and methods used in this study; Section 4 provides the results obtained from the experimental study and analyses the main finding; Section 5 outlines the conclusions drawn from the study and suggests potential paths for further research.

2. RELATED WORK

In recent years, there has been significant interest in applying object detection methods in agriculture, particularly with the emergence of deep learning techniques. This section aims to review the literature on the use of deep learning object detection methodologies for identifying diseases in plant leaves.

Traditional approaches for disease detection in agriculture have typically relied on visual inspection by experts, laboratory-based testing, and statistical models [3]. While effective to some extent, these methods often suffer from being time-consuming, prone to subjective errors, and requiring specialised expertise [4], [2].

A. Classical Methods for Disease Detection in Plants

Classical methods have limitations that have led to the development of machine learning-based solutions. Previous studies have utilised algorithms such as Decision Trees [5], [6], K-Nearest Neighbours (K-NN) [7], [6], and Support Vector Machines (SVM) [7], [8], [9], [10], which rely mainly on predefined features. While these techniques hold promise, they face challenges with scalability and struggle to handle complex visual patterns. As Tsaftaris et al. [11] noted, specialised domain knowledge is crucial for expertly designing and extracting features, a process commonly referred to as feature engineering. Furthermore, translating this expertise into image analysis procedures and filters, such as edge detectors, requires a high level of skill in image processing.

B. Deep Classification Methods for Disease Detection in Plants

With the evolution of computational capabilities, deep learning has emerged as a powerful tool for various applications in agriculture, from image recognition to weather prediction [12]. Deep learning uses models with many layers to learn data representations at different levels of complexity [13]. These techniques have significantly advanced various fields, such as speech recognition, visual object recognition, object detection, drug discovery, and genomics [13]. The backpropagation algorithm is utilised to teach machines how to adjust their internal parameters to compute the representation in each layer. Deep convolutional networks deal with images, video, speech, and audio, while recurrent networks analyse sequential data like text and speech [13].

Plants affected by diseases commonly display distinct signs or abnormalities on their leaves, stems, flowers, or fruits. Each disease or pest issue tends to manifest a unique visual pattern that can serve as a diagnostic indicator. Leaves are often the main focus for detecting plant diseases, as they are usually the first to show symptoms [14]. Several studies have focused on applying deep learning architectures for the specific purpose of plant disease detection [15]. These works often provide a comparative analysis of different architectures and evaluate them based on metrics such as accuracy, speed, and complexity.

Mohanty et al [16] trained a deep neural network using a dataset of 54,306 plant leaf images that can identify 14 crop species and 26 diseases with an accuracy of 99.35%. Their method shows that it is feasible to use deep learning models to diagnose crop diseases on a global scale using smartphones. Lu et al [14] trained CNN-based model on 500 real-world rice plant images, achieved a 95.48% accuracy in identifying 10 common diseases, which outperform traditional methods. A deep convolutional neural network (DCNN) was used to identify four types of cucumber diseases from field images with an accuracy rate of 93.4%. Data augmentation was applied to prevent overfitting on the dataset of 14,208 symptom images. The DCNN outperformed traditional classifiers and proved its efficacy for real-world applications as claimed in [17]. Additionally, extensive research efforts have been undertaken to diagnose diseases affecting a diverse range of plant leaves, such as tomato [18], [19], apple [20], [21], rice [22], [23].

C. Deep Object detection Methods for Disease Detection in Plants

After examining the role of deep learning classifiers for plant disease detection in the previous section, the focus will now shift to explore deep object detection techniques. These methods offer the added advantage of locating as well as identifying diseases on plant leaves. Object detection, as a specialised area within deep learning, has also seen numerous innovations. Architectures like YOLO (You Only Look Once)[24], Faster R-CNN [25], and SSD (Single Shot MultiBox Detector) [26] have been widely used in various domains particularly in agriculture.

Xie et al.[27] propose a real-time grape leaf disease detector, Faster DR-IACNN, which employs improved deep convolutional neural networks and integrates the Inceptionv1 module, Inception-ResNet-v2 module, and SE-blocks for enhanced feature extraction. Experimental results indicate the model achieves an 81.1% mean average precision (mAP) on the Grape Leaf Disease Dataset (GLDD) and operates at a detection speed of 15.01 FPS, which demonstrate its feasibility for real-time diagnosis of grape leaf diseases. Fuentes et al [28] explored the efficacy of various deep learning meta-architectures, such as Faster R-CNN, R-FCN, and SSD, combined with deep feature extractors like VGG net and ResNet, for detecting diseases and pests in tomato plants.

Utilising an extensive dataset of approximately 5,000 images, the system achieves a mAP of 85.98%.

It effectively identified nine different conditions and showing robustness in complex plant environments.

Tian et al [29] leverage DenseNet to optimize YOLO-V3's low-resolution feature layers. Their model outperforms Faster R-CNN with VGG16 network and other benchmarks in real-time detection of anthracnose lesions on apple surfaces.

Utilising advanced optimisation techniques, Yae et al. [30] enhances the YOLOv5 model to accurately and swiftly detect defects in kiwifruits, achieving a significant 94.7% mAP. Similarly, Mathew et al. [31] employs YOLOv5 for real-time detection of bacterial spot disease in bell pepper plants, thereby empowering farmers to take early preventive action based on mobile phone-captured images.

D. Limitations and Conclusion

Nonetheless, limitations in existing works remain, Many of the studies in the field of plant disease detection through deep learning have predominantly framed the problem as a classification task. This orientation has led to the generation of datasets that are largely optimised for classification algorithms, including the labelling and structuring of the data in ways that facilitate classification-based approaches. As a result, the landscape of available datasets is somewhat skewed, with fewer datasets engineered to tackle the challenge as an object detection task. This presents a limitation in the current body of work and points to the need for datasets that are structured to allow object detection methodologies. The scarcity of such datasets designed for object detection constrains the broader application and assessment of these more advanced techniques.

In summary, while extensive research has been conducted in the realm of deep learning for object detection in agriculture, additional avenues for research remain open. The current study aims to contribute to this area by employing object detection as a viable method for diagnosing plant diseases using Pear Leaf disease as a case study. The research places special emphasis on analysing the results based on the severity of the disease.

3. MATERIALS AND METHODS

A. DiaMOS Plant Dataset

For this study, we utilized the recently introduced DiaMOS Plant Dataset [32], which comprises 3,006 leaf images collected from three different trees within the same plot in Italy. The images were captured using a smartphone (Honor 6×) and a DSLR camera (Canon EOS 60D) at a resolution of 2976×3968, as well as two lateral cameras with a resolution of 3456 × 5184.

Figure 1 displays four example images from the dataset representing four classes from left to right: healthy leaf, slug damage, leaf spot, and leaf curl.

The leaf images were captured from the upper side (adaxial) in real-life settings under various lighting, angle, background, and noise conditions throughout the entire growing season. This approach allowed the researchers to obtain realistic leaf images and track the evolution of visual symptoms

[32].

Fig. 1. Example camera images from the DiaMOS Plant Dataset, which contains four classes from left to right: healthy, slug, spot, and curl.

The dataset at hand presents a 2D object detection task with a multi-class problem. The task involves identifying four different types of leaf states: healthy leaf, slug damage, leaf _spot, and leaf curl. Each leaf is assigned a severity score ranging from 0 to 4, with 0 indicating a healthy state and 4 representing a high spread of disease. This is especially important as it allows for a more precise and accurate assessment of the severity of the leaf disease.

The severity score can also help identify the level of treatment required for the leaf. For instance, if the severity score is low, treatment may not be necessary. However, if the score is high, immediate treatment may be required to prevent the further spread of the disease.

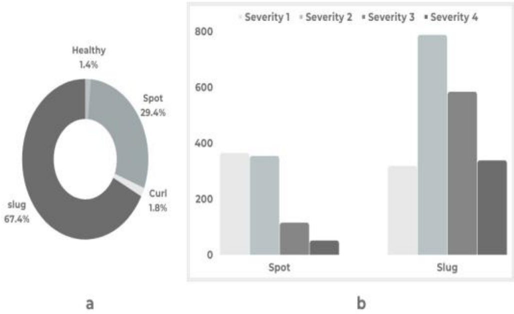

Fig. 2.DiaMOS Plant Dataset Label Distribution:

Figure (2-a) on the left illustrates a statistical overview of the label distribution across the four classes. This information can be used to better understand the frequency of each type of leaf state in the dataset. It can further aid in the development of more accurate and efficient models for identifying and treating leaf diseases.

The dataset exhibits a highly imbalanced distribution of objects. The number of Healthy and Curl leaves is significantly lower than the number of Spot and Slug leaves. For this reason, we will focus only on Spot and Slug leaves in this study and exclude Healthy and Curl leaves. It is worth noting that no severity score was given to Curl leaves, which is another reason we will drop this class from our analysis.

In the same figure (2-b) on the right, we can observe the distribution of severity scores for both Spot and Slug leaves. Spot leaves mostly exhibit very early signs of the disease. Most of them score severity 1 or 2. Only 13% score severity 3, and roughly 6% score severity 4. On the other hand, Slug leaves have only 16% scoring Severity 1, while the majority belong to Severity 2, with a percentage of 40.

The 2D bounding box labels in this dataset are tightfitting. They cover the centre leaf in the image. The labels were formatted in YOLO format. We converted the label to conform with the COCO style used in two-stage object detectors for models with a two-stage detection approach.

B. Deep Learning Object Detectors

Object detectors are usually categorised into two primary types: one-stage detectors and two-stage detectors. Two-stage detection frameworks begin by generating candidate regions of interest, which are then classified in a second stage. The most popular two-stage detector is the region-based convolutional neural network (R-CNN) [33],

which mainly uses selective search [34] to generate proposals and then uses CNN to extract features from them. Subsequently, Fast-RCNN [35] facilitates the extraction of features from different proposals through a special pooling layer to reduce computations from a single map. Later, to improve performance, Faster R-CNN [25] introduced a unified end-to-end detector that utilises a region proposal network (RPN) for both feature extraction and proposal generation. FPN (Feature Pyramid Network) [36] merges features extracted at different resolutions and provides scale-specific anchor boxes in object detection tasks. Cascade RCNN [37] is a recent work in the field of object detection that aims to improve the quality of proposals for the COCO AP (Average Precision) metrics, which are used to evaluate the object detection models performance. In this work, we have utilised three variations of R-CNN detectors described in Table 1.

TABLE 1. THE FASTER R-CNN VARIATIONS UTILISED IN THE WORK, INCLUDING THE BACKBONE AND FEATURE EXTRACTION USED

Model Name	Pre-trained Backbone	Detection Head
faster-rcnn r50 c4	ResNet-50	Conv4
faster rcnn r50 fpn	ResNet-50	FPN
cascade-rcnn r50 fpn	ResNet-50	FPN

On the other hand, one-stage detectors predict both the object's position and its class in one step. It introduces a novel object detection approach named YOLO (You Only Look Once) [24]. Unlike previous methods that reuse classifiers for detection, YOLO models object detection as a regression issue that predicts bounding boxes and class probabilities from whole images in just one step. This enables the optimisation of the entire detection pipeline end-to-end for detection performance, as it is a single network [24]. YOLO is known for being very fast due to its approach of modelling object detection as a regression problem, which eliminates the need for a complex pipeline. Instead, the neural network can be run on a new image during testing to predict detection.

In our work, we have selected two variations of the YOLO family: YOLOv5s [38] and YOLOv8s [39]. The two variations come in five sizes: nano (n), small (s), medium (m), large (l), and extra large (x). The convolution layers' width and depth are adjusted to meet certain application and hardware needs. We selected the small size due to the dataset's small size and the model's significantly smaller weight file size. This makes it suitable for embedded devices and real-time detection, such as on mobile phones, which is our intended goal for this work. Additionally, YOLOv5s have high detection accuracy and can achieve a detection speed of up to 140 frames.

The overall design of YOLOv5 Utilising a modified CSPDarknet53 backbone with a stem, the network design has convolutional layers for extracting image features and a spatial pyramid pooling fast (SPPF) layer for expediting calculations by pooling information into a fixed-size map. SiLU activation and batch normalisation are included into every convolution. The head of the network is akin to YOLOv3 , whereas the neck employs SPPF and a modified CSP-PAN [40] [41].

YOLOv8 replaces the CSPLayer from YOLOv5 with a C2f module and uses a spatial pyramid pooling fast (SPPF) layer to pool features into a fixed-size map for faster computation. As in YOLOv5, each convolution has batch normalisation and SiLU activation. An anchor-free approach is employed, where the prediction head independently performs objectness scoring, classification, and bounding box regression [41].

As previously mentioned, mainstream object detectors can be divided into two-stage and one-stage detectors. However, a new family of object detectors based on transformer architecture has been introduced.

Carion et al. [42] proposed DETR, a new object detection method that uses a transformer encoder-decoder architecture and set-based global loss to predict object sets directly without hand-designed components. DETR is comparable to other detectors in terms of accuracy and runtime performance and can also be applied to panoptic segmentation.

Liu et al. [43] proposed a new approach to enhance the performance of the DETR detection model by using box coordinates as queries in Transformer decoders. It led in better results on the MS-COCO benchmark compared to other similar models. By directly learning anchors as queries, the proposed DAB-DETR (Dynamic Anchor Box DETR) offers a novel query formulation that makes it feasible to modify the positional cross-attention map in transformer decoders and execute layer-by-layer dynamic anchor changes by using anchor size.

Using a single ResNet-50 model as the backbone for training 50 epochs, DAB-DETR scored the highest performance among DETR-like architectures on the COCO object detection benchmark, with an AP of 45.7%.

Previous works [44], [45], [46], [47] aimed to enhance DETR through various methods, such as associating each query with a specific location or introducing Gaussian priors or deformable sampling points. However, unlike DAB-DETR, they do not use anchors as queries. In this work, we utilised DAB-DETR (Dynamic Anchor Box DETR) [43].

C. Experimental Setup

As mentioned previously, the study will focus on Spot and Slug leaves, as there are significantly fewer Healthy and Curl leaves. To maintain the percentage of samples for each class, the dataset was split into training, validation, and testing in a ratio of 7:2:1, respectively, using random stratified sampling following the same approach described in [32]. Additionally, in order to conduct an analysis based on the severity of the disease, we divided the test split into four datasets: test _severity1, test _severity2, test _ severity3, and test severity4, each corresponding to a specific severity score.

The Faster R-CNN variations and the DAB-DETR model experiments were conducted using the public repository MMDetection [48] as well as the official repositories of both YOLOv5s and YOLOv8s models [38], [39].

Using transfer learning can be a useful technique for training a large target network without overfitting, particularly when the target dataset is much smaller than the base dataset [49]. Transfer learning usually starts by training a base model and copying its first *n* layers to a new model. The rest of the new model is randomly initialized and trained for the target problem. Errors from the new task can be back-propagated into the base (copied) features to fine-tune them for the new task [49]. In order to improve the performance of the models, we utilised pre-trained weights learned from the COCO task. These weights serve as an excellent starting point as they are already optimised for a similar task. We then fine-tuned the models by training them on the modified DiaMOS Plant Dataset, which allowed the models to learn and adapt to the specific features and characteristics of pear diseases.

Table 2 presents the optimisation, hyperparameters, and augmentation used for training the models, and most of the hyperparameters for all training experiments are in line with the choices outlined in the original papers of the models.

4. RESULTS AND DISCUSSION

In this section, we will perform a performance analysis of selected deep object detection models, detailed in Table 2. These models were evaluated across four distinct test datasets-test severity1, test severity2, test severity3, and test severity4—each of which corresponds to varying levels of disease severity. First, we introduce an overview of the evaluation metrics used in the analysis. Then, we present the overall performance results combined across all four test datasets for both diseases. Lastly, we evaluate the results of each test dataset.

A. Evaluation Metrics

Average Precision (AP) metric is used to evaluate object detection models performance. It captures whether the model balances precision and recall. Precision shows how many objects the model recognizes correctly. Recall shows how many actual objects the model detected. Thus, AP is calculated from the Precision-Recall curve where the area under the curve is measured. Therefore, it is an efficient way to capture the model's ability to detect objects accurately across various confidence thresholds. The higher the AP score, the more effective the object detection model. This indicates fewer false positives and false negatives. The following equation describes AP mathematically:

$$\text{AP} = \int_0^1 P(R)\, dR \quad (1)$$

where $P(R)$ represents precision as a function of recall.

If there are different object classes, the Mean Average Precision (mAP) is computed by taking the average of AP score across all classes:

$$\text{mAP} = \frac{1}{n}\sum_{i=1}^{n} AP_i \quad (2)$$

where n is the total number of classes.

B. Overall Results

Table 3 shows the performance comparison of six object detectors across all test datasets for disease detection results. In evaluating the overall Mean Average Precision (mAP), both cascade-rcnn r50 fpn and YOLOv8s outperform the others, achieving mAPs of 88.2 and 88.3, respectively. YOLOv5s has an mAP of 85.5, with dab-detr r50 slightly behind at 85. On the other hand, faster rcnn r50 fpn and faster-rcnn _r50 c4 have resulted in the lowest mAPs of 83.8 and 83.0, respectively.

When breaking down the Average Precision (AP) for individual diseases, YOLOv8s demonstrates the best performance in the identification of leaves with pear slug with an AP of 89.6. Subsequently, Cascade-rcnn scored an AP of 88.6. The rest of the models demonstrate similar AP values, averaged around 86.0, with the exception of faster-rcnn r50 c4, which trails with an AP of 84.6. Conversely, in detecting leaf _spot leaves, cascade-rcnn leads with an AP of 87.7. YOLOv8s closely trails with an AP of 86.9. Notably, faster _rcnn r50 fpn struggles in this detection task, with the lowest AP at 80.3. The remaining models tend to cluster around an AP of approximately 84.0 when detecting this disease.

Figure 3 illustrates the Confusion Matrix of all models across all test datasets for disease detection results. It is evident that all models did very well regarding detecting leaves with pear slug disease. However, most of the models struggled when detecting leaves with leaf _spot except for YOLOv8s, which achieved the highest TP rate. It is worth noting that all models succeeded in detecting the majority of bounding boxes. The model dab-detr r50 is the only one that has a high rate of not detecting leaves with leaf spot disease, with a false negative rate of 18%. Moreover, the models have a high rate of detecting non-existent affected leaves.

Figure 3 illustrates the confusion matrix for all models across all test datasets. It highlight the disease detection results. It is evident that each model performed well in detecting leaves with pear slug disease. However, most models faced challenges in identifying leaves with leaf spot, with YOLOv8s standing out as it achieved the highest true positive (TP) rate. It is worth noting that all models were successful in detecting the majority of bounding boxes. The model dab-detr r50, in particular, struggled, being the only one with a high false negative (FN) rate of 18% for leaves with leaf spot disease. Furthermore, the models tend to report a high rate of false positives, incorrectly signalling the presence of affected leaves when there are none.

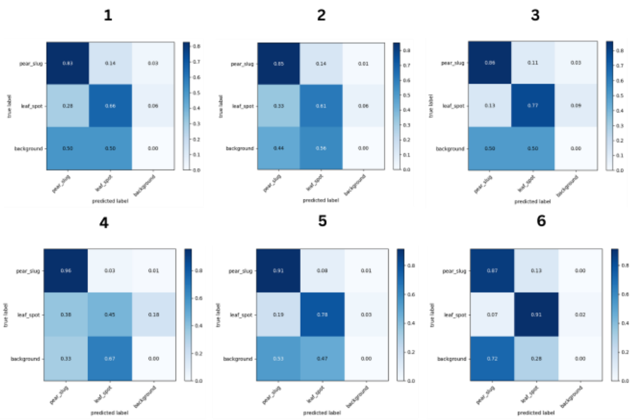

Fig. 3. The Confusion Matrix of six object detectors across all test datasets for disease detection results

C. Severity Analysis

In this section, we assess the ability of each model to detect varying disease severity levels. Table 4 presents the performance metrics for identifying diseases in their early stages for test _severity1, characterised by mild symptoms. Notably, YOLOv8s, with an mAP of 82.4, and cascade-rcnnr50 fpn, with an mAP of 82.1, performed well in detecting leaves with mild disease symptoms. YOLOv5s and dab-detrr50 were close with a mAP of 81.2 and 80.5, respectively. On the lower spectrum, faster rcnn r50 fpn and faster-rcnn r50 c4 scored an mAP of 72.6.

TABLE 2.USED OPTIMISATION, HYPERPARAMETERS, AND AUGMENTATION FOR TRAINING THE MODELS

Model	Optimisation	LR	Epoch	Mini Batch	Augmentation
faster-rcnn r50 c4	SGD	0.0025	12	16	RandomFlip
faster rcnn r50 fpn	SGD	0.02	12	16	RandomFlip
cascade-rcnn r50 fpn	SGD	0.0025	12	16	RandomFlip
dab-detr r50	AdamW [50]	0.0001	20	50	Random Flip Random Crop Random Scale
YOLOv5s	SGD	0.01	20	32	Mosaic Random affine HSV Albumentations [51]
YOLOv8s	SGD	0.01	20	32	Mosaic Random affine HSV Albumentations [51]

TABLE 3. PERFORMANCE COMPARISON OF SIX OBJECT DETECTORS ACROSS ALL TEST DATASETS FOR DISEASE DETECTION RESULTS

Model	pear slug AP	leaf spot AP	mAP
faster rcnn r50 fpn	0.85	0.83	0.84
faster-rcnn r50 c4	0.86	0.80	0.83
cascade-rcnn r50 fpn	0.89	0.88	0.882
dab-detr r50	0.86	0.84	0.85
YOLOv5s	0.86	0.84	0.85
YOLOv8s	0.90	0.87	0.883

TABLE 4.PERFORMANCE COMPARISON OF SIX OBJECT DETECTORS FOR DISEASE DETECTION RESULTS ON TEST _SEVERITY1

Model	pear slug AP	leaf spot AP	mAP
faster rcnn r50 fpn	0.81	0.65	0.73
faster-rcnn r50 c4	0.83	0.63	0.73
cascade-rcnn r50 fpn	0.86	0.78	0.82
dab-detr r50	0.85	0.76	0.81
YOLOv5s	0.84	0.79	0.81
YOLOv8s	0.85	0.80	0.82

Examining the results for each disease in test _ severity1, we can see that the models performed better with detecting pear slug disease, with an average of AP 83.3 compared to detecting leaf spot disease with an average AP of 73.3.

Regarding pear slug disease, cascade-rcnn r50 fpn outperforms the others with an AP of 86. Interestingly, dab-detr _ r50 was the second-best model in identifying leaves with pear slug of this severity with an AP of 85.2, competing with YOLOv8s, which scored 85.1. On the other hand, faster rcnn _r50 fpn scored the lowest AP of 80.5.

In terms of leaf spot disease, YOLOv8s scored the highest AP of 79.7, which is comparatively close to YOLOv5s and cascade-rcnn r50 fpn with an AP of 78.8 and 78.3, respectively. While dab-detr r50 performance deteriorated when detecting this disease, with a score of 75.8. Faster _rcnn r50 fpn and faster-rcnn r50 c4 scored the lowest AP with an average of

63.6.

Figure 4 shows the confusion Matrix of six object detectors for disease detection results on test severity1. It is clear that the models are more accurate when identifying leaves with pear slug disease with an average of 77.33% for TP rather than leaf spot which averaged a TP of 68.33%.

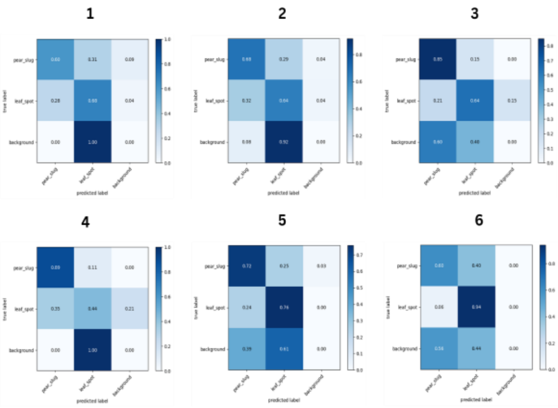

Fig. 4. The Confusion Matrix of six object detectors for disease detection results on test severity1

The performance metrics for identifying diseases in the test severity2 dataset are shown in Table 5. First, with a mAP of 87.1, Cascade-rcnn r50 fpn had the highest mAP, closely followed by YOLOv5 (86.7). The other models have mAPs centered around the value of 83.43. When assessing the results for each disease within test severity2, for the pear _slug disease, both YOLOv8 and Cascade-rcnn r50 fpn achieved notable APs of 92.2 and 91.9, respectively. Conversely, the remaining models resulted in an average AP of 89.2. In the case of the leaf spot disease, cascade-rcnn r50 fpn scored the highest AP score at 82.2, while the other models had an average AP of 80. Notably, YOLOv5s and faster-rcnn r50 c4 performed last with the lowest APs, recording 76.3 and 76.1, respectively.

Figure 5 shows the Confusion Matrix of six object detectors for disease detection results on test severity2. Similarly to test severity1, the models were inclined to identify the pear slug class with greater accuracy than the leaf spot class. The average True Positive rate for pear slug was higher at 89.0% compared to leaf spot's 72.0%. Both classes generally exhibited low False Negative rates. However, the leaf spot class had a notably higher average False Positive rate than the pear slug class.

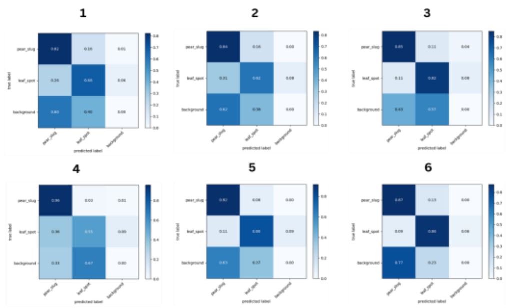

Fig. 5. The Confusion Matrix of six object detectors for disease detection results on test severity2

Table 6 displays the performance metrics for disease identification using the test _ severity3 dataset. YOLOv8s outperformed all other models, scoring an mAP of 95.0. Following this, cascade-rcnn r50 fpn registered an mAP of 93.5, and YOLOv5s recorded 92.7. Examining the results for individual diseases, YOLOv8s achieved the highest AP for pear slug at 94.4. In contrast, faster-rcnn r50 c4 recorded the lowest AP for this disease, of a score of 88.6.

For the leaf _spot disease, YOLOv8s again led with an AP of 95.6, while cascadercnn r50 fpn closely followed with an AP of 95.1. The lowest AP for leaf spot was 88.9, as scored by faster-rcnn r50 c4.

Figure 6 shows the confusion Matrix of six object detectors for disease detection results on test severity3.In the evaluation of the prediction matrices for the given models, it was observed that the pear _ slug class consistently had a TP rate of approximately 98%. For leaf spot, TP rates varied, with a peak of about 77%. Both classes showed low FN rates. The leaf spot class had a higher FP rate compared to pear slug in some instances.

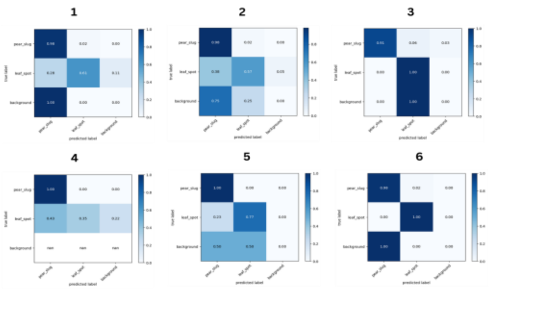

Fig. 6. The Confusion Matrix of six object detectors for disease detection results on test severity3

Table 7 displays the performance metrics for disease identification in the test severity4 dataset. YOLOv5s and faster-rcnn r50 c4 achieved the highest mAP values, recording 90.0 and 89.0, respectively. In contrast, dab-detr r50 resulted in the lowest mAP at 85.3. With regard to the pear slug disease, faster-rcnn _r50 c4 led with an AP of 86.6. Surprisingly, YOLOv8s showed the lowest AP at 79.8, while the majority of models achieved an AP close to 82.73. For the leaf_spot disease, there was a noticeable improvement across all models. YOLOv5s and YOLOv8s achieved closely matched APs of 95.3 and 94.7, respectively. Cascade-rcnn r50 fpn achieved an AP of 93.7, whereas dab-detr _r50 underperform with the lowest AP of 87.9.

Figure 7 shows the Confusion Matrix of six object detectors for disease detection results on test severity4. It is evident that the pearslug class has an average TP rate of 97.0%, whereas the leafspot class holds a rate of 76%. Both classes maintain minimal FN rates. Nevertheless, the leaf _spot class shows a higher FP rate compared to the pear slug class.

D. Discussion

The findings indicate a disparity in the performance of the models when detecting and classifying between severity1 and test severity2. For the pear slug disease AP, there isn't a consistent pattern across the severity levels for all models. Some show a slight decline from Severity 1 to Severity 4, while others maintain approximately the same performance levels. In contrast, the leaf _ spot AP disease generally demonstrates improved performance as the severity increases. Specifically, the results for Severity 3 and Severity 4 tend to surpass those of Severity 1 and Severity 2. This is also reflected in the mAP metric, where it appears higher for Severities 3 and 4, which shows that the models are more effective at identifying cases of higher severity.

These findings indicate that the models have a greater ability to accurately detect and classify cases of leaf spot disease as their severity increases. It's evident that many models frequently achieve peak performance at Severity 3. This could mean that the traits or characteristics that are unique to Severity 3 cases are more clear, making them easier for the models to spot than traits or characteristics that are unique to other severity levels.The best overall performance was achieved by YOLOv8s, which prove that its architecture and feature extraction capabilities are appropriate for this task. This might be due to the advances made in comparison to prior iterations of YOLO.

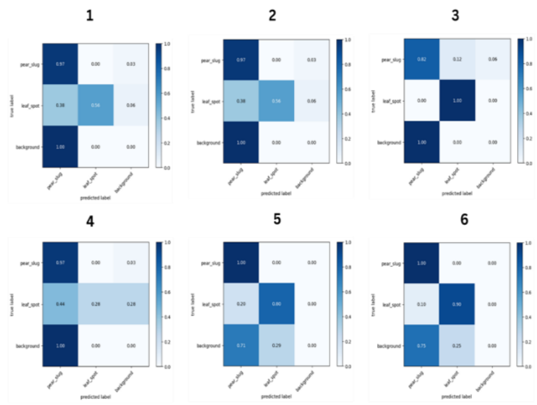

Fig. 7. The Confusion Matrix of six object detectors for disease detection results on test severity4

TABLE 5. PERFORMANCE COMPARISON OF SIX OBJECT DETECTORS FOR DISEASE DETECTION RESULTS ON TEST _SEVERITY2

Model	pear slug AP	leaf spot AP	mAP
faster rcnn r50 fpn	0.90	0.80	0.85
faster-rcnn r50 c4	0.89	0.76	0.83
cascade-rcnn r50 fpn	0.92	0.82	0.87
dab-detr r50	0.89	0.78	0.84
YOLOv5s	0.89	0.76	0.83
YOLOv8s	0.92	0.81	0.87

TABLE 6. PERFORMANCE COMPARISON OF SIX OBJECT DETECTORS FOR DISEASE DETECTION RESULTS ON TEST _SEVERITY3

Model	pear slug AP	leaf spot AP	mAP
faster rcnn r50 fpn	0.89	0.93	0.91
faster-rcnn r50 c4	0.89	0.89	0.89
cascade-rcnn r50 fpn	0.92	0.95	0.93
dab-detr r50	0.87	0.91	0.89
YOLOv5s	0.90	0.95	0.93
YOLOv8s	0.94	0.96	0.95

TABLE 7. PERFORMANCE COMPARISON OF SIX OBJECT DETECTORS FOR DISEASE DETECTION RESULTS ON TEST _SEVERITY4

Model	pear slug AP	leaf spot AP	mAP
faster rcnn r50 fpn	0.80	0.95	0.87
faster-rcnn r50 c4	0.83	0.94	0.88
cascade-rcnn r50 fpn	0.85	0.95	0.90
dab-detr r50	0.81	0.92	0.86
YOLOv5s	0.83	0.88	0.85
YOLOv8s	0.87	0.91	0.89

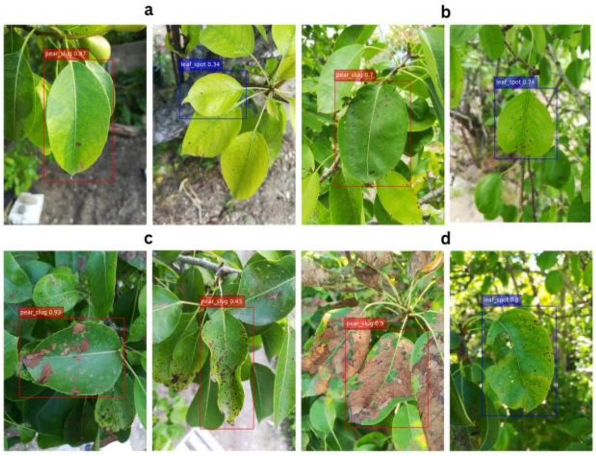

Fig. 8. YOLOv8s prediction visualization results on randomly selected images from each severity dataset in all groups. The first image contains a leaf with pear slug disease, while the second image contains a leaf with leaf spot disease. The group a represents two example images from test severity1. The group b represents two example images from test severity2. The group c represents two example images from test severity3. The group d represents two example images from test severity4.

The cascade rcnn r50 fpn model demonstrates competitive performance, which suggest that its strategy of using a multi-stage region proposal network and utilising feature pyramid networks (FPNs) for extracting features at various scales is additionally effective for this particular task. However, it is worth noting that YOLOv8s outperforms cascade-rcnn r50 fpn.

Figure 8 illustrates the YOLOv8s prediction visualisation results on randomly selected images from each severity dataset in all groups. The first image contains a leaf with pear slug disease, while the second image contain a leaf with leaf spot disease. The group a represents two example images from test severity1. The group b represents two example images from test severity2. The group c represents two example images from test severity3. The group d represents two example images from test severity4. It is clear the model was proficient at locating the centre disease with high confidence in predicting the correct class in most of the cases. The only case where one of the images is mistakenly classified as a different disease is in group c for the leaf with leaf spot disease, and the model has been predicted as pear slug. This confusion may be due to the fact that the symptoms on the leaves of the disease are similar to those caused by pear slug.

Because the dataset was annotated to detect the centre leaf, the models are inclined to detect leaves in the background, which leads to a high rate of false positives.

Adjusting the dataset annotations to include side leaves could potentially improve the accuracy of the models in detecting them.

5. CONCLUSION

Early-stage leaf diseases in plants can have significant impacts on crop yield and quality. If left undetected and untreated, these diseases can spread rapidly, which led to widespread damage and potential crop loss. Early detection allows for timely intervention and targeted disease control measures to minimis the negative effects on plant health and optimising crop production. Additionally, early-stage disease detection can help farmers adopt more natural and safe disease control methods, reducing the reliance on chemical pesticides and promoting sustainable farming practices.

This article has critically examined the effectiveness of various object detection methods in identifying early-stage leaf diseases in plants, with a particular focus on pear leaf disease. Our exploration utilised advanced machine learning algorithms, including several variants of R-CNN detectors and YOLO models, to analyse plant leaf images. The key findings reveal a nuanced performance disparity across different models when detecting and classifying pear pear slug and leaf spot diseases, particularly across varying severity levels.

Notably, the YOLOv8s model emerged as the most effective, with an mAP of 88.3. This emphasises the potential of YOLOv8s in agricultural applications, especially in aiding farmers to adopt more natural and safe disease control methods. However, the cascade-rcnn r50 fpn model also displayed high performance, which highlight the effectiveness of its multi-stage region proposal network and feature pyramid networks in handling diverse scales of feature extraction.

When it comes to how well these models work at finding diseases early on, YOLOv8s and cascade-rcnn _r50 fpn did a great job of finding mild disease symptoms in test severity1, with mAPs of 82.4 and 82.1, respectively. Their performance was especially noteworthy in detecting pear slug disease, with an average AP of 83.3, compared to leaf spot disease, with an average AP of 73.3. In the test severity2 dataset, Cascade rcnn r50 fpn led with the highest mAP of 87.1.

While models like YOLOv8s and Cascade-RCNN show promise in detecting early leaf diseases in pears, there's a need to expand this research to include more plant types. Currently, our focus is limited and doesn't cover the wide range of diseases affecting different crops. It's crucial to adapt these models for various plant diseases and make them easy for farmers to use. Developing simple, practical tools, like mobile apps, from these models can help farmers quickly spot and treat plant diseases. This step is key to improving crop health and promoting sustainable farming.

REFERENCES

[1] S. Attaluri and R. Dharavath, "Novel plant disease detection techniquesa brief review," *Molecular Biology Reports*, pp. 1–14, 2023.

[2] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, pp. 1–18, 2021.

[3] A. Khakimov, I. Salakhutdinov, A. Omolikov, and S. Utaganov, "Traditional and current-prospective methods of agricultural plant diseases detection: A review," in *IOP Conference series: earth and environmental science*, vol. 951, no. 1. IOP Publishing, 2022, p. 012002.

[4] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning—a review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021.

[5] J. Chopda, H. Raveshiya, S. Nakum, and V. Nakrani, "Cotton crop disease detection using decision tree classifier," in *2018 International Conference on Smart City and Emerging Technology (ICSCET)*. IEEE, 2018, pp. 1–5.

[6] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, 2019.

[7] A. S. Deshapande, S. G. Giraddi, K. Karibasappa, and S. D. Desai, "Fungal disease detection in maize leaves using haar wavelet features," in *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 1*. Springer, 2019, pp. 275–286.

[8] S. Kaur, S. Pandey, and S. Goel, "Plants disease identification and classification through leaf images: A survey," *Archives of Computational Methods in Engineering*, vol. 26, pp. 507–530, 2019.

[9] S. Kumar and R. Kaur, "Plant disease detection using image processinga review," *International Journal of Computer Applications*, vol. 124, no. 16, 2015.

[10] M. Martineau, D. Conte, R. Raveaux, I. Arnault, D. Munier, and G. Venturini, "A survey on image-based insect classification," *Pattern Recognition*, vol. 65, pp. 273–284, 2017.

[11] S. A. Tsaftaris, M. Minervini, and H. Scharr, "Machine learning for plant phenotyping needs image processing," *Trends in plant science*, vol. 21, no. 12, pp. 989–991, 2016.

[12] A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture:´ A survey," *Computers and electronics in agriculture*, vol. 147, pp. 70– 90, 2018.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.

[15] C. Jackulin and S. Murugavalli, "A comprehensive review on detection of plant disease using machine learning and deep learning approaches," *Measurement: Sensors*, p. 100441, 2022.

[16] S. P. Mohanty, D. P. Hughes, and M. Salathe,´ "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, 2016. [Online]. Available:
https://www.frontiersin.org/articles/10.3389/fpls.2016.01419

[17] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on

deep convolutional neural network," *Computers and electronics in agriculture*, vol. 154, pp. 18–24, 2018.

[18] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Computers and Electronics in Agriculture*, vol. 161, pp. 280–290, 2019.

[19] R. Thangaraj, S. Anandamurugan, and V. K. Kaliappan, "Automated tomato leaf disease classification using transfer learning-based deep convolution neural network," *Journal of Plant Diseases and Protection*, vol. 128, no. 1, pp. 73–86, 2021.

[20] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[21] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang, and Y. Shi, "Mobilenet based apple leaf diseases identification," *Mobile Networks and Applications*, pp. 1–9, 2022.

[22] W.-j. Liang, H. Zhang, G.-f. Zhang, and H.-x. Cao, "Rice blast disease recognition using a deep convolutional neural network," *Scientific reports*, vol. 9, no. 1, p. 2869, 2019.

[23] D. Li, R. Wang, C. Xie, L. Liu, J. Zhang, R. Li, F. Wang, M. Zhou, and W. Liu, "A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network," *Sensors*, vol. 20, no. 3, p. 578, 2020.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779– 788.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision– ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[27] X. Xie, Y. Ma, B. Liu, J. He, S. Li, and H. Wang, "A deep-learningbased real-time detector for grape leaf diseases using improved convolutional neural networks," *Frontiers in plant science*, vol. 11, p. 751, 2020.

[28] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deeplearning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, 2017.

[29] Y. Tian, G. Yang, Z. Wang, E. Li, Z. Liang *et al.*, "Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense," *Journal of Sensors*, vol. 2019, 2019.

[30] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A real-time detection algorithm for kiwifruit defects based on yolov5," *Electronics*, vol. 10, no. 14, p. 1711, 2021.

[31] M. P. Mathew and T. Y. Mahesh, "Leaf-based disease detection in bell pepper plant using yolo v5," *Signal, Image and Video Processing*, pp. 1–7, 2022.

[32] G. Fenu and F. M. Malloci, "Diamos plant: A dataset for diagnosis and monitoring plant disease," *Agronomy*, vol. 11, no. 11, p. 2107, 2021.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[34] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, pp. 154–171, 2013.

[35] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[36] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie,´ "Feature pyramid networks for object detection," in *Proceedings of the IEEE*

conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[37] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2019. [Online]. Available: http://dx.doi.org/10.1109/tpami.2019.2956516

[38] Ultralytics, "YOLOv5: A state-of-the-art real-time object detection system," https://docs.ultralytics.com, 2021, accessed: August,30,2023.

[39] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023.

[Online]. Available: https://github.com/ultralytics/ultralytics

[40] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement, arxiv. org.(2018)," *arXiv preprint arXiv:1804.02767*, 2022.

[41] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo: From yolov1 to yolov8 and beyond," *arXiv preprint arXiv:2304.00501*, 2023.

[42] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in

European conference on computer vision. Springer, 2020, pp. 213–229.

[43] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "Dab-detr: Dynamic anchor boxes are better queries for detr," *arXiv preprint arXiv:2201.12329*, 2022.

[44] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[45] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of detr with spatially modulated co-attention," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3621–3630.

[46] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and

J. Wang, "Conditional detr for fast training convergence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3651–3660.

[47] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor detr: Query design for transformer-based object detection," *arXiv preprint arXiv:2109.07107*, vol. 3, no. 6, 2021.

[48] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng,

Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[49] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[50] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[51] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.