AFF-YOLO: A Real-time Industrial Defect Detection method based on Attention Mechanism and Feature Fusion

Preprint · October 2023

DOI: 10.21203/rs.3.rs-3449230/v1

1 author:

Manas Mehta

2 PUBLICATIONS **2** CITATIONS

SEE PROFILE

AFF-YOLO: An Industrial Defect Detection method based on Attention Mechanism and Feature Fusion

Manas Mehta[1]

[1]Author affiliations: none, Bangalore, India

[1]Author email: manasshitalkumar.mehta2019@vitstudent.ac.in

[1]Author orcID: 0009-0000-5077-2998

Abstract

Steel, as a pivotal material in industrial society, demands stringent quality control to ensure its structural integrity and safety. Surface defects in steel pose significant challenges to the manufacturing process, affecting mechanical properties and visual aesthetics. In this paper, we introduce an enhanced version of the YOLOv5 object detection model tailored for the precise identification of surface defects in steel plates. Our proposed architecture incorporates innovative modifications, including the integration of an Effective Channel Attention Network (ECA-Net) specifically within the neck of the network to enhance attention and filtering, allowing the model to focus on relevant steel surface defects while reducing noise and distractions. In addition to the change made in the neck of the architecture, the Bidirectional Feature Pyramid Network (BiFPN) concatenation was added to introduce bidirectional informational flow. Finally, Adaptive Spatial Feature Fusion (ASFF) in the prediction head which enhances feature fusion across different scales, enabling the model to better learn and recognize complex patterns associated with steel defects. These enhancements empower the YOLOv5 network to focus on relevant objects while filtering out distracting information, resulting in improved accuracy and detection speed. To evaluate the model's performance, we conducted experiments using the NEU-DET dataset as a base and then further enhanced it with preprocessing techniques. Our model was compared it with the original YOLOv5 object detection algorithm, and achieved an mAP of 84.7%. Our findings demonstrate a remarkable 6.5% increase in mean Average Precision (mAP) compared to the original YOLOv5 architecture while maintaining a reasonable FPS for real-time usage, affirming the effectiveness of our proposed enhancements.

Keywords: YOLOv5, Steel Surface Defect Detection, Attention Mechanism, Multi-Feature Fusion, Computer Vision, Object Detection

1. Introduction

Steel is one of the most important of all metals in terms of its quantum and variety of use. Steel has played a vital role in advancing industrial societies, and it's often used as a key indicator to assess a country's level of development [1]. As per the World Steel Association, production of crude steel in 2022 was 1,885,738 thousand tons (Mt). When we compare steel to similar materials, it stands out with its cost-effective production process. Extracting iron from ore demands only a quarter of the energy required for aluminium extraction, making steel an energy-efficient choice. Additionally, steel is environmentally friendly due to its recyclability, contributing to sustainability. With 5.6% of the Earth's crust composed of iron, it offers a stable source of raw materials. Impressively, steel production surpasses the combined production of all non-ferrous metals by a factor of 20 [2]. Steel's exceptional strength has revolutionized construction, enabling towering skyscrapers and intricate bridges. It underpins the transportation industry, providing both stability and flexibility for modern designs. Beyond its mechanical prowess, steel fuels industrialization, shaping machinery and tools. In energy, it's vital for power plants and corrosion-resistant pipelines. Steel's recyclability aligns with environmental goals, reducing resource consumption and waste. In an era of heightened environmental consciousness, steel plays a crucial role in promoting sustainability across various sectors.

Quality issues with flat steel can result in substantial economic losses and damage the reputation of steel manufacturers. In the case of thin and wide flat steel, surface defects pose the most significant risk to product quality. Even in situations where internal defects occur sporadically, there is a high likelihood of visible changes in the surface's appearance [3]. Surface defects on steel components, originating from various sources including manufacturing, handling, and environmental exposure, have detrimental effects. Cracks, scratches, and inclusions create stress points, diminishing load-bearing capacity and increasing the risk of premature failure. These defects also heighten susceptibility to corrosion, weakening structural integrity. In industries where aesthetics matter, such as architecture and automotive manufacturing, surface irregularities mar visual appeal. Moreover, defects disrupt manufacturing processes, leading to higher costs, material wastage, and reduced efficiency. They adversely impact wear resistance, corrosion resistance,

fatigue strength, and other vital properties of steel. Therefore, detecting these defects is crucial for safety, reliability, and cost-effectiveness.

This process is usually performed manually in industries, which is unreliable and time-consuming. In order to replace the manual work, it is desirable to allow a machine to automatically inspect surface defects from steel plates with the use of computer vision technologies [4]. These traditional methods [5] also faced problems such as low accuracy and high labour intensity. The conventional breakthrough in machine learning was a significant leap forward from manual examination. It is typically initiated with manually extracting features. Subsequently, these features were extracted and then inputted into a classifier to accomplish defect classification. As stated before, due to the dependence on manually formulated feature extraction rules, this approach resulted in weak resilience and limited ability to adapt to new situations. It was easily affected by external factors and noise, consequently diminishing the accuracy of defect detection. Since 2012, CNNs (convolutional neural networks) have taken over as the standard model for vision tasks in the field of computer vision [6]. Since then, object detection has been broadly classified as either single-stage or single-target detectors, or region-based / two-staged detectors. The YOLO family [11,13-16] is an exemplary technique representing single-stage detection, and the R-CNN [19-21] family represents the two-stage detection algorithms. Deep learning-based industrial research is currently being used in a variety of research fields since it can completely use the data's potential characteristics without the requirement for manually designing them. Luo et al. [7] introduced an algorithm for detecting surface defects using YOLO feature enhancement. This approach enhanced detection speed but exhibited limited accuracy. In a separate study, Liu et al. [8] presented a method for identifying insulators and detecting defects in aerial images by employing the YOLO algorithm by using attention mechanism modules and called it YOLO-SO. By combining these models, they effectively addressed issues related to both the speed and accuracy of insulator defect detection. Shun et al [9] experimented on defect detection using the Yolov5 model which improved their accuracy significantly as compared to the yolov4 model. However, even now, there are quite a few hurdles that we still need to factor in to make improvements.

The typical surface imperfections found in steel surfaces, encompass crazing, inclusions, patches, pitting, rolled-in scale, and scratches. These can be seen in figure 1. The intraclass defects in the dataset result in considerable differences in appearance, such as the category scratches exhibiting horizontal, vertical, or slanted scratch defects. Meanwhile, interclass defects such as rolled-in scale, crazing, and pitted surfaces have similar properties to each other. The presence of variations in lighting and material characteristics within grayscale images makes it exceptionally difficult to detect defects that share similarities across different defect categories [10]. Furthermore, considering the diverse range of steel surface defects, some of these defects might overlap in terms of location and similarity in features. In typical classification tasks, the focus is often on identifying defects within a category with the highest confidence level, leading to less precise classification outcomes [4].

Therefore, to solve the problems related to poor detection and classification, as well as improving existing methods for detecting defects in steel surfaces, we introduce the AFF-YOLO (Attention and Feature Fusion based YOLOv5). All the modifications are made in the neck of the architecture. We included an effective channel attention network (ECA-Net) mechanism into the backbone network, connected it in parallel to the C3 module, and termed it ECA-C3 to help YOLOv5 network improve against distracting information and concentrate on useful target objects.

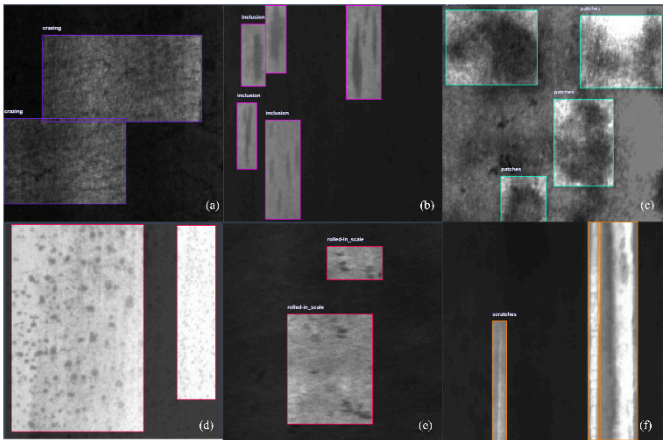

Fig. 1 Different Classes used in the NEU-DET dataset **(a)** Crazing; **(b)** Inclusion; **(c)** Patches; **(d)** Pitted_surface; **(e)** Rolled-in_scale; **(f)** Scratches

In the body, along with the ECA attention module, we introduce the BiFPN feature fusion. BiFPN is a component introduced in the EfficientDet object detection architecture [12]. It is designed to address some limitations of traditional Feature Pyramid Networks (FPN) used in object detection models. BiFPN enhances the feature pyramid by introducing bidirectional connections and combining different levels of features in a more adaptive manner. This allows information to flow both up and down the pyramid, helping to capture more context and details at various scales. The adaptive spatial feature fusion (ASFF) was introduced for feature fusion of different scales in the prediction head. Using the NEU-DET dataset in the experiment, we assessed our algorithm and contrasted it with the original YOLOv5 algorithm. By including additional attention mechanisms and multi-scale feature fusion techniques, our model outperforms the original YOLOv5 architecture. Our contributions to the model are listed below:

i. We introduce the ECA attention module within the body of the network, replacing the original C3 convolution modules. This module is referred to as the ECA-C3 module.
ii. We upgraded the conventional feature pyramid network and by introducing the BiFPN feature fusion, in the body of the network.
iii. The ASFF module was attached before the prediction head to introduce feature fusion of varied scales and improve the networks learning.
iv. Our model outperformed the original Yolov5 model by showing 6.5% increase in the mAP.

The remainder of this paper follows a structured organization. In Section 2, we delve into the existing body of work related to our research. Section 3 offers an in-depth exploration of both the methodology employed in the original YOLOv5 model and the enhancements introduced in our novel AFF-YOLO. In Section 4, we engage in a detailed discussion of our experiments and the outcomes obtained while working with the NEU-DET dataset. This section also includes an ablation study to shed further light on our findings. Finally, Section 5 summarizes the conclusions we have drawn from our experimentation and outlines future improvements.

2. Related Work

2.1. Data Augmentation

In the realm of computer vision and object detection, data augmentation stands as a pivotal technique to enhance model generalization and performance. In the pursuit of refining the YOLOv5 model, a state-of-the-art architecture for real-time object detection, a strategically curated augmentation strategy has been employed. This strategy encompasses various transformations, including horizontal and vertical flipping, which effectively diversify the training dataset, introducing variants of object orientations and perspectives. Furthermore, the augmentation strategy is extended through the innovative incorporation of mosaic augmentation [13]. This method involves synthesizing a novel training image by stitching together four distinct images, each contributing a quarter of the mosaic input. Mosaic augmentation not only enriches the dataset with complex scenes and object interdependencies but also enhances the model's ability to comprehend objects within cluttered environments. This augmentation schema not only helps with a robust learning process but also enables the model to be more resilient to variations encountered in real-world scenarios.

2.2. Object Detection

Over time, considerable progress has been achieved in the field of object detection, thanks to the evolution of both single-stage and two-stage detectors. Single-stage detectors, exemplified by the YOLO (You Only Look Once) model [14], stand out for their efficiency in predicting object classes and bounding box coordinates in a single network pass. This pioneering approach ushered in real-time object detection capabilities and served as inspiration for subsequent evolutions like YOLOv2 [15] and YOLOv3 [16], which focused on enhancing both speed and accuracy. The evolution of the YOLO series continued with YOLOv4 and YOLOv5 [11], with YOLOv5 gaining prominence for its streamlined architecture and remarkable performance improvements. Similarly, SSD (Single Shot MultiBox Detector) [17] introduced a single-stage framework with a multi-scale feature hierarchy for improved detection precision. Extending this line of development, RetinaNet [18] introduced the concept of focal loss, addressing class imbalance and thereby enhancing accuracy, particularly in scenarios with abundant background samples.

On the other hand, two-stage detectors, notably exemplified by Faster R-CNN (Region Convolutional Neural Network) [19], introduced a more intricate multi-step approach that involves region proposal generation followed by object classification and bounding box regression. This pursuit of higher accuracy led to the emergence of diverse R-CNN variants such as R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN [19-22] steadily advancing the performance of object detection. These developments also gave rise to the idea of separating region proposal from detection, enabling greater flexibility and precision. Cascade R-CNN [23] pushed this concept further by introducing a sequential cascade of detectors to iteratively enhance detection quality. Amidst these advancements, other notable contributions have shaped the field of object detection. CornerNet [24] introduced a novel approach by directly predicting object keypoints to improve detection accuracy. CenterNet [25] followed suit by focusing on predicting object centers and sizes, demonstrating impressive results in real-time detection scenarios. DETR (DEtection TRansformers) [26] harnessed transformer architectures to cast object detection as a set prediction problem, showcasing the potential of attention mechanisms in object detection.

While the interplay between single-stage and two-stage detectors remains significant, the emergence of YOLOv5 and its diverse modifications have introduced a new dynamic to the field. By prioritizing both speed and accuracy, YOLOv5 and its variations have demonstrated that single-stage detectors can excel on both fronts, shaping the discourse around object detection.

2.3. Attention Mechanisms

Attention mechanisms (AMs) represent a revolutionary breakthrough in artificial intelligence and machine learning, enabling models to selectively concentrate on specific elements within datasets and considerably amplifying performance. The seminal work by Vaswani et al. [27] introduced self-attention through the transformer architecture, fundamentally reshaping neural machine translation by capturing contextual associations among words in sequences. Subsequent advancements in AMs have given rise to innovative adaptations such as CBAM [28], harmoniously integrating spatial and channel attention to enhance image classification. Meanwhile, SE blocks [29] recalibrate feature responses across channels, augmenting model flexibility. The halo attention [12] method, showcases the capacity to adeptly capture long-range dependencies within images. ECA [30] establishes an efficient channel attention mechanism adept at capturing interdependencies among channels with computational efficiency. While other attention mechanisms like SE or non-local attention offer similar capabilities as the ECA, ECA achieves it with fewer computational resources making it superior. These varied AM strategies collectively underscore the transformative potential of attention mechanisms in driving forward AI-powered solutions.

2.4. Multi-Feature Fusion

The groundbreaking paper by Lin et al. [31] introduced the idea of fusing multi-scale features to enhance the representation capabilities of convolutional neural networks (CNNs). This technique aims to combine features extracted

from different levels of the network hierarchy, effectively capturing both fine-grained and high-level contextual information. Multi-feature fusion addresses the limitations of traditional single-scale feature extraction by harnessing the strengths of various features, leading to improved object localization, scale invariance, and semantic context awareness.

Over the years, the field of multi-feature fusion has evolved, giving rise to several influential techniques that have significantly impacted object detection architectures. Feature Pyramid Networks (FPN), introduced by Lin et al. [31] and later integrated into YOLOv3 [16] and YOLOv4 [14], paved the way for seamless integration of multi-scale features through a top-down and bottom-up architecture. This innovation greatly improved object detection performance by enabling more accurate localization and enhanced semantic understanding. Progressive Attention Networks (PAnet) refined multi-feature fusion by dynamically assigning attention weights to different scales, enhancing the discriminative capabilities of the fused features.

Another milestone, the BiFPN (Bilateral Feature Pyramid Network), introduced by Tan et al. [12], tackled the challenge of efficient feature fusion by introducing a bidirectional information flow. This mechanism enables high-quality feature fusion at multiple scales, resulting in superior object detection accuracy. Adaptive Spatial Feature Fusion (ASFF), a novel idea for pyramidal feature fusion, proposed by Liu et al. [32], presented a learnable feature fusion approach that adaptively selects features from multiple resolutions, enhancing the model's adaptability to diverse object scales and aspect ratios.

These multi-feature fusion techniques have influenced architecture design, particularly evident in the popular YOLO (You Only Look Once) family of object detection models. Integrating FPN, PAnet, BiFPN, and ASFF into YOLO architectures has enabled these models to achieve state-of-the-art results, striking an optimal balance between accuracy and efficiency.

3. Methodology

3.1. YOLOv5 Architecture

In recent years, YOLO (You Only Look Once) object detection models have gained significant attention for their real-time capabilities and remarkable performance. YOLOv5, an evolution of the YOLO series, stands out as a notable advancement, introducing improvements in both speed and accuracy over its predecessors. YOLOv5's architecture takes advantage of multi-scale features, efficient components, and advanced fusion techniques to deliver state-of-the-art object detection results. YOLOv5 addresses some of the limitations of previous YOLO versions by introducing a lighter and more efficient architecture without compromising accuracy. Its single-stage approach processes the entire image in one pass, making it faster than many other object detection methods. The architecture maintains impressive accuracy by leveraging feature pyramids to capture multi-scale information crucial for detecting objects of various sizes. YOLOv5's flexibility is evident in its ability to handle real-time and large-scale applications, including detection tasks in both images and videos. The architecture of YOLOv5 consists of three main components: the body, neck, and head. The body serves as the feature extraction backbone, the neck fuses multi-scale features, and the head predicts object classes and bounding box coordinates. The architecture of the Original YOLOv5 model is depicted elaborately in Fig. 2.

Body: YOLOv5 employs a modified CSPDarknet53 backbone as its body, optimizing the architecture for efficiency and accuracy. The CSPDarknet53 architecture employs cross-stage feature fusion, enhancing the network's ability to capture both low-level and high-level features. This enables the network to understand object details while maintaining contextual information. Neck: The neck of YOLOv5 utilizes PANet (Path Aggregation Network) to fuse features from different stages of the backbone. This fusion is essential for multi-scale feature representation, enabling the model to detect objects of various sizes and contexts accurately. PANet employs lateral connections and a top-down path to aggregate features from different scales, contributing to robust feature fusion. Head: The head of YOLOv5 consists of anchor-based detection modules. The model predicts the objectness scores, class probabilities, and bounding box coordinates for each anchor. YOLOv5 employs anchor boxes with aspect ratios tailored to specific objects, enhancing detection performance. Additionally, the head includes enhancements from YOLOv4, such as CIoU loss and PANet aggregation, contributing to better localization accuracy and object classification.

The defects present on the surface of steel exhibit irregular shapes, unpredictable positions, and varying sizes. Furthermore, a significant quantity of these smaller-scale targets often exists. Given these circumstances, the original YOLOv5 model falls short of fully fulfilling the detection requirements. To address this, this research enhances the original YOLOv5 network model in several ways. Initially, we made modifications in the neck by integrating attention mechanisms to emphasise critical information while concurrently reducing the impact of unwanted features, upgrading the feature fusion in the neck, and adding an ASFF module before the detection head. These refinements were geared towards enhancing the detection model's adaptability to the identification of minor defects. The culmination of these improvements has resulted in an enhancement in the overall performance of defect detection in the model.

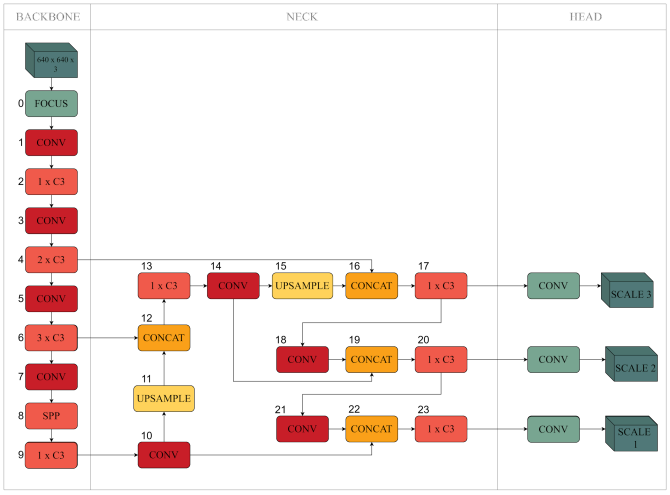

Fig. 2 Architecture of the Original YOLOv5 Model

3.2. AFF-YOLO architecture

The architecture of the improved version of YOLOv5 is depicted in Fig. 5. The improvements on the base model (YOLOv5) have been made to increase the detection accuracy in the NEU DET dataset for steel surface defect detection.

3.2.1. ECA -Attention Mechanism

Attention mechanisms play a crucial role in contemporary neural network architectures, aiming to enhance information processing and connectivity across different segments of the network. These mechanisms enable models to selectively concentrate on pertinent features while disregarding irrelevant ones, emulating the human attention process. Well-established attention mechanisms like BAM, SE, CBAM and ECA-Net have been verified to enhance the performance of detection models. ECA-Net manages to achieve substantial performance improvements with only a minimal increase in complexity, adding just a small number of parameters. By integrating the ECA attention module, YOLOv5 becomes more effective at capturing long-range dependencies between channels, resulting in better recognition and discrimination of relevant features.

Beginning with an input feature map containing multiple channels, each representing distinct features, ECA computes a global context for each channel, signifying its relative importance compared to others. This computation involves a learnable parameter, often represented as a 1D convolutional layer, which calculates channel-specific attention coefficients. These coefficients are learned during training, enabling the network to dynamically adjust the significance of each channel. The kernel size, represented by K plays a crucial role in this process, as it determines the scope and range of information integration. By applying a fast one-dimensional convolution with a specific kernel size, ECA efficiently captures the channel-wise dependencies and interactions across the feature map. The 1D convolutions of size K are represented why the adaptive function as described in equation (1).

$$K = \psi(C) = \left|\frac{(C)}{\gamma} + \frac{b}{\gamma}\right|_{odd} \quad (1)$$

These weights determine the importance of each channel in the overall representation. By incorporating the ECA attention module, YOLOv5 selectively amplifies informative channels while suppressing less relevant ones, leading to a more focused and discriminative feature representation. The integration of the ECA attention module brings several benefits to YOLOv5. It improves context understanding by capturing long-range dependencies, enhances the model's ability to recognize objects by emphasizing important features, and does so without significantly increasing computational complexity. Fig. 3 shows the structural diagram of the ECA module.

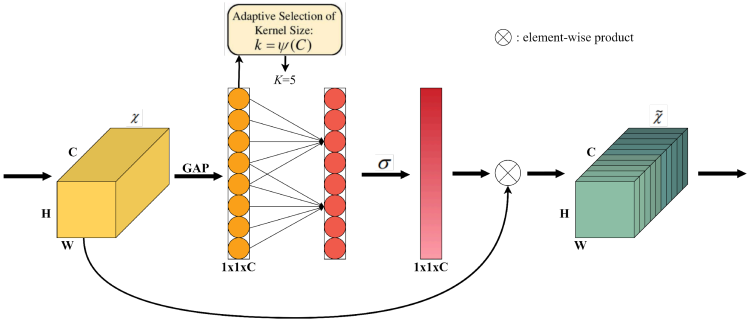

Fig. 3 Schematic diagram of the ECA-Net

In summary, the ECA attention module enhances YOLOv5 by improving channel dependencies, enhancing discriminative power, and ultimately leading to improved object detection performance.

3.2.2. Bifpn Concatenation

Within the model's neck component, substituting the original PANet with BiFPN proves to be very effective. This is particularly evident in scenarios with a dataset containing small-sized images. The BiFPN excels in combining both high-resolution and low-resolution image feature data. This enhancement is especially beneficial for images containing defects like "crazing" and "rolled-in scale," as these defect types typically feature numerous small objects.

FPN, while effective in generating multi-scale feature maps for object detection, struggles with handling fine-grained details and preserving information across different scales efficiently. In contrast, BiFPN is a novel architecture that optimally addresses these shortcomings. BiFPN introduces bidirectional connections and lateral connections between adjacent feature maps, allowing for enhanced information flow in both top-down and bottom-up directions. This bi-directional approach enables improved handling of semantic information and finer object details, leading to more accurate object localization and classification. Fig. 4 depicts the difference between the PANet structure used in the original YOLOv5 and the BiFPN structure which is incorporated in the AFF-YOLO.

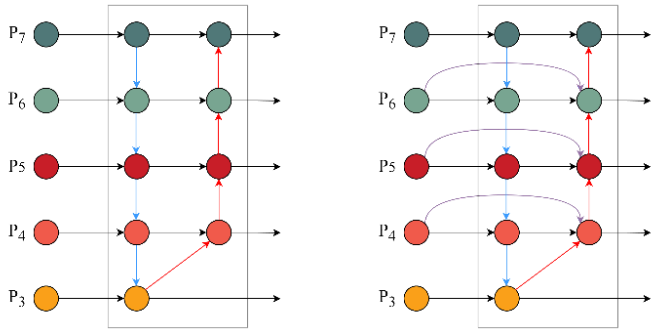

Fig. 4 Comparison of PANet (left) and BiFPN (right)

By incorporating BiFPN, YOLOv5 gains a more context-aware and fine-grained understanding of the image, leading to improved object localization and classification accuracy. The bidirectional nature of BiFPN allows it to capture both high-level semantic features and low-level details, offering a comprehensive view of the scene. This enhanced feature extraction directly translates into better object detection performance.

3.2.3. ASFF

The Adaptive Spatial Feature Fusion module addresses the challenge of effectively merging multi-scale features extracted from different layers of a convolutional neural network (CNN). Its advantages are manifold: firstly, it significantly enhances detection performance by dynamically fusing features from various network layers, resulting in more precise object localization and classification. Secondly, ASFF excels in handling scale variations, making it adaptable to scenarios with objects of diverse sizes within the same image. It accomplishes this while maintaining computational efficiency, reducing memory usage, and computational complexity during both training and inference, thus being suitable for real-time applications. The ASFF module functions by applying feature reseizing and adaptive fusion.

Feature resizing: $X^{a \to b}$ signifies the adjustment of the feature map moving from level a to level b. Here, a and b belong to the set {1,2,3}. $ASFF - detect^{l}$ is obtained by combining and incorporating the semantic information from Level 1, Level 2, and Level 3 using distinct weights denoted as α, β, and γ. This combination is mathematically defined in Equation (2).

$$ASFF - detect^{l} = X^{1 \to l} \times \alpha^{l} + X^{2 \to l} \times \beta^{l} + X^{3 \to l} \times \gamma^{l} \quad (2$$

Adaptive Fusion: After the feature resizing, adaptive fusion is applied. $x_{ij}^{a \to l}$ is the feature vector at a location represented by (i, j) and a belongs to the set {1,2,3}. Equation (3) represents the feature fusion for a particular level l.

$$y_{ij}^{l} = \alpha_{ij}^{l} \times x_{ij}^{1 \to l} + \beta_{ij}^{l} \times x_{ij}^{2 \to l} + \gamma_{ij}^{l} \times x_{ij}^{3 \to l} \quad (3)$$

The term y_{ij}^{l} represents the output features at position (i, j) within a particular channel level l. Meanwhile, α_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} denote the spatial significance weights for the three distinct feature mapping levels learned by the network up to level l. These three terms can be straightforward single values that are the same for all channels. Each of these terms can be defined by equation (4) given below.

$$\alpha_{ij}^{l} = \frac{e^{\lambda_{\alpha_{ij}}^{l}}}{e^{\lambda_{\alpha_{ij}}^{l}} + e^{\lambda_{\beta_{ij}}^{l}} + e^{\lambda_{\gamma_{ij}}^{l}}} \quad (4)$$

The values α_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} are determined by the parameters $\lambda_{\alpha_{ij}}^{l}$, $\lambda_{\beta_{ij}}^{l}$ and $\lambda_{\gamma_{ij}}^{l}$, $respec$tively, using the softmax function as the controlling parameters. $X^{1 \to l}$, $X^{2 \to l}$ and $X^{3 \to l}$, help with the calculation of the weights λ_{α}^{l}, λ_{β}^{l} and λ_{γ}^{l} by making the use of some 1×1 convolutional layers. They are then learned in the standard method of using back propagation, like any other neural network. In the improved YOLOv5 model, the features from these three levels are dynamically combined at their respective scales, and the fused features are then fed into the head for steel defect classification and detection.

Thus, with all these enhancements made to YOLOv5 architecture, we created the AFF-YOLO which stands for Attention and Feature Fusion YOLOv5. The final architecture is shown in Fig. 5. The summary of the Updated Model can be seen in Fig. 6. In Fig. 6, "from" represents where the input for the given module is taken from (-1 representing the previous module), "n" represents the number of given modules attached together, "params" represents the number of parameters at that stage of the network. Further, the module name and the exact shape of the input for that module are also given. This figure aims to give a better understanding of the model.

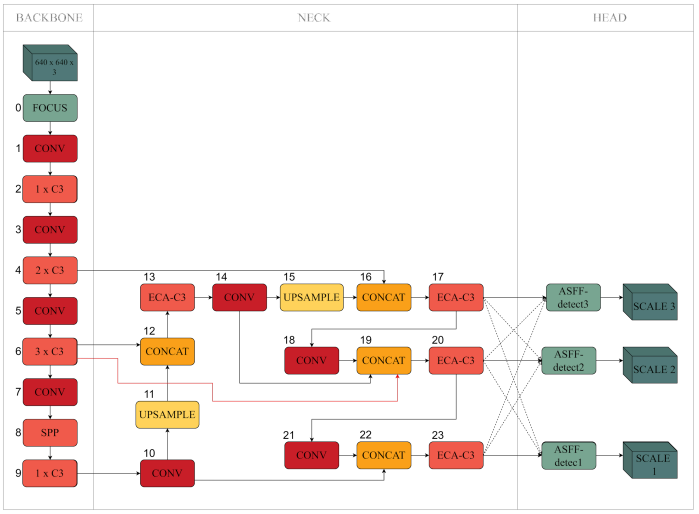

Fig. 5 Architecture of the AFF-YOLO

	from	n	params	module	arguments
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	493059	models.common.ECAC3	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	123651	models.common.ECAC3	[256, 128, 1, False]
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14, 6]	1	0	models.common.Concat	[1]
20	-1	1	493059	models.common.ECAC3	[512, 256, 1, False]
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1707011	models.common.ECAC3	[512, 512, 1, False]
24	[17, 20, 23]	1	5469125	models.yolo.ASFF_Detect	[6, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]

Fig. 6 Summary of AFF-YOLO

4. Experimentation and Results

We use the NEU-DET dataset to evaluate out improved version of yolov5. Our model reported a MaP-50 of 84.7%.

4.1. Experimentation Setup

4.1.1 Experimental setup

The code was run on a Google Colab pro + environment making use of the Nvidia A100 GPU with 40GB memory. In the experimental training, an SGD optimizer with an initial learning rate of 0.01 and a weight decay coefficient of 0.0005 was used. The confidence level was set to 0.5 for mAP-50 and 95 for mAP-95 and the model ran for 300 epochs each time. The batch size was set to 32. The input size of the images was 640×640.

4.1.2 Dataset

The initial dataset is the NEU-DET steel surface defect dataset, which was released by Northeastern University. The dataset can be accessed via the following link: http://202.118.1.237/yunhyan/NEUsurface-defect-database.html (accessed on 1 September 2022). This dataset was originally introduced in the paper written by He et al. [4] This dataset contains 1800 images and encompasses six defect categories, namely cracks, inclusions, patches, pitted_surface, rolled-in_surface, and scratches, each containing 300 images. All the images within this dataset are of dimensions 200×200 pixels. Upon carefully studying which preprocessing and augmentation technique helps the model learn better, the dataset was expanded using Horizontal flip, Vertical flip and Mosaic Augmentation. Each image was also stretched to 640×640 pixels. Post augmentation, the total number of images wa increased to 4144 and the split was roughly 86:7:7 (train: test: validation) or 3544: 300: 300. Fig. 1 provides illustrations of various defects found in the baseline dataset. The grayscale representations reveal that even within the same defect category, there can be significant variations in appearance. For instance, images of scratches may exhibit both horizontal and vertical scratch patterns.

4.1.3 Evaluation Metrics

In this paper, we used Precision, Recall, mean Average Precision (mAP), and Frames Per Second (FPS), which serve as crucial performance indicators. Precision gauges the accuracy of positive predictions, revealing the proportion of correctly identified instances among all positive predictions. In contrast, recall measures the model's capacity to capture all relevant instances, indicating the proportion of correctly identified instances among all actual positives. mAP, a composite metric, evaluates the overall performance of object detection or recognition systems by calculating the average precision across multiple categories, offering a holistic evaluation of model quality. Lastly, FPS quantifies the processing speed of a model, a vital factor in real-time applications. Collectively, these metrics provide a comprehensive assessment of a model's effectiveness, striking a balance between accuracy, efficiency, and comprehensiveness across diverse applications, from autonomous vehicles to medical image analysis. These metrics are represented as follows:

$$precision = \frac{TP}{TP+FP} \tag{5}$$

$$recall = \frac{TP}{TP+FN} \tag{6}$$

$$accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{7}$$

$$AP = \sum_{n}\left(R_n - R_{n-1}\right)P_n \tag{8}$$

In this context, TP, FP, FN, and TN stand for the counts of true positives, false positives, false negatives, and true negatives, respectively. Precision and Recall are defined by the equations (5) and (6). When the defect prediction category is accurate, and the intersection over union (IoU) exceeds a certain threshold (in our experiments, this threshold is set at 0.5), we regard the detection as accurate. The accuracy can be calculated in terms of TP, TN, FN and FP and is shown in equation (7). AP or Average Precision is calculated as given in equation (8), where R_n and P_n are the Recall and Precision at the nth threshold. The mAP (mean AP) is the mean of AP over all such instances. The AP corresponds to the region under the precision-recall (P-R) curve.

4.2. NEU-DET dataset Comparison Experiment

In this study, our improved model was compared with the original YOLOv5s model and the main metrics for comparison were Precision, Recall and mAP to compare the model accuracy, and FPS to compare the inference speed.

Table 1 Quantitative comparison of our model with the original YOLOv5 model

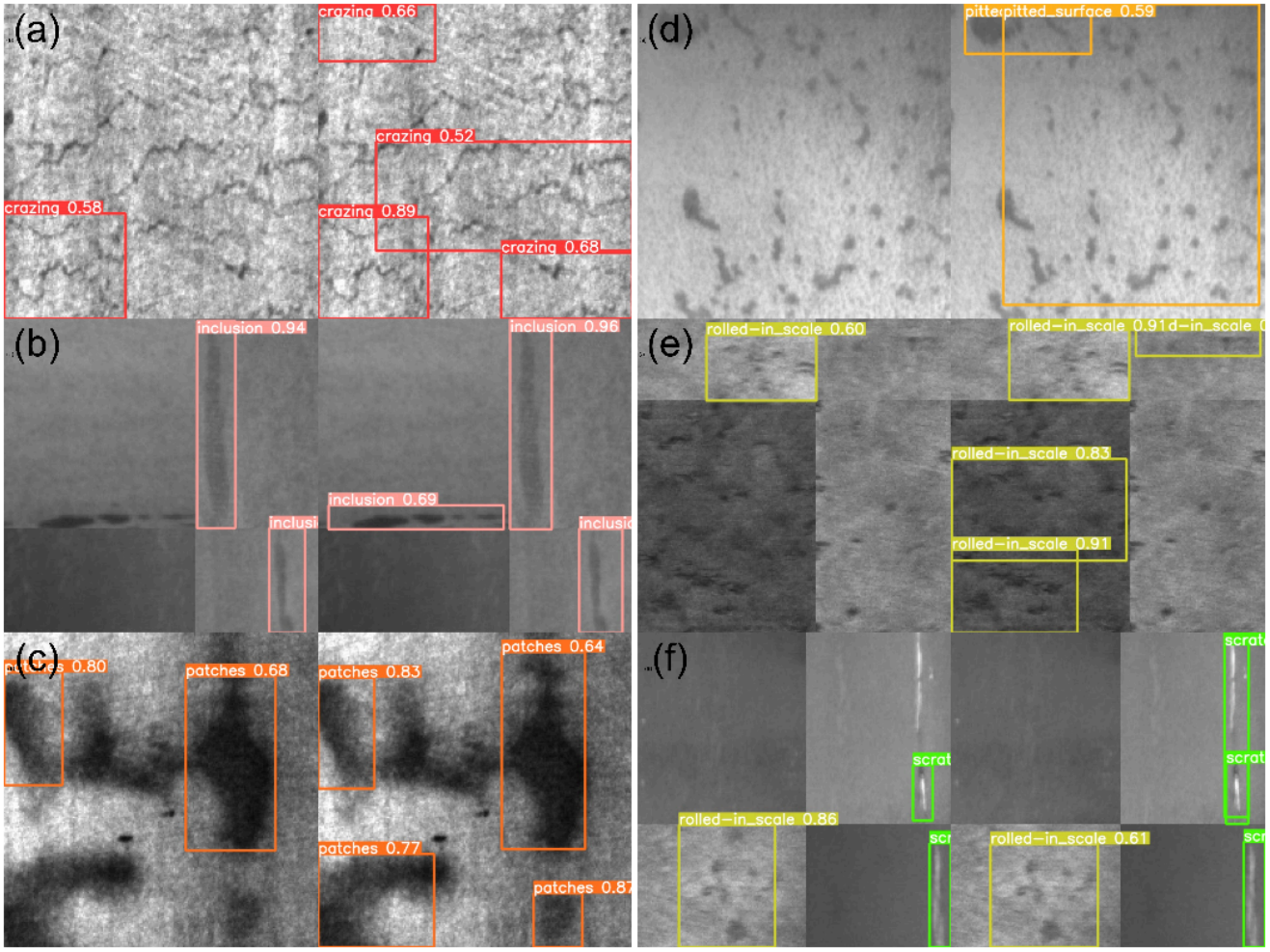

Fig. 7 Comparison of how the model performed on the test dataset. (left) YOLOv5 inference (right) AFF-YOLO inference. **(a)** Crazing; **(b)** Inclusion; **(c)** Patches; **(d)** Pitted_surface; **(e)** Rolled-in_scale; **(f)** Scratches

Model	Class	Precision (%)	Recall (%)	mAP-50 (%)	mAP50-95 (%)
YOLOv5	all	72.9	77.7	78.2	47.8
	crazing	51.9	51.2	46.8	20.5
	inclusion	74.3	91.7	84.8	48.2
	patches	84.9	93.3	94.2	59.0
	pitted_surface	76.0	70.4	80.1	48.7
	rolled-in_scale	63.5	66.2	67.2	38.3
	scratches	86.9	93.3	96.0	72.1
AFF-YOLO	all	83.6	81.3	84.7	**54.5**
	crazing	67.1	65.8	66.8	**31.5**
	inclusion	83.0	89.3	84.6	**52.9**
	patches	89.2	94.0	96.7	**63.5**
	pitted_surface	88.3	76.5	88.0	**53.5**
	rolled-in_scale	80.2	71.3	75.5	**49.3**
	scratches	93.8	90.9	96.6	**76.3**

After rigorously training our model based on multiple rounds of testing and training, we ran the final improved version of yolov5 model on the test set of images and compared it with what the original yolov5 model detected in these test images. These images were selectively chosen for each class to give a better understanding of how the AFF-YOLO outperforms the standard yolov5. Fig. 7 depicts this visual analysis by selecting one example for each class. Our model, the AFF-YOLO, was trained to keep in mind the problems of steel defect detection. The specific combination of the feature fusion and attention mechanism in the neck and the head, helped the model detect defects of different scales (as seen in Fig. 7 under crazing, patches, rolled-in_scale and scratches) as well as orientation (as seen in Fig.7 under Inclusion and rolled-in_scale), much

better than the original model. It also improved the general capabilities of detection (as seen in Fig. 7 under pitted_surface). The quantitative analysis of these two models can be seen in Table 1. As shown in the table, Our model, or the AFF-YOLO outperforms the original YOLOv5 model in terms of mAP for every class. In terms of mAP-50% aggregated over all classes, there was an increase in accuracy by 6.5%. The precision over all classes showed an improvement of 10.8% along with a 3.6% increase in Recall.

4.3 Ablation Study

In the realm of computer vision models enhanced through the integration of new modules, we employ ablation studies to systematically assess the specific contributions of these modules to the overall performance of the model. In these studies, we methodically remove individual modules or components from the model and observe the resulting impact on its accuracy, robustness, and efficiency. Through these rigorous experiments, we gain valuable insights into the relative importance of each module, guiding us in fine-tuning the model's architecture and optimizing its capabilities. Ablation studies thus serve as an indispensable tool in our iterative model development process, ensuring that the incorporation of new modules indeed leads to substantial improvements in the field of computer vision. These experiments were also conducted on the NEU-DET dataset.

We use 7 variations, each representing an improvement to the YOLOv5 architecture, as shown in Table 2:

Table 2 List of variations proposed for the ablation study

As shown in Table 2, the ECA-C3 module refers to the ECA attention module that was added to the neck of the architecture, BiFPN refers to the BiFPN concatenation module which implements BiFPN feature fusion into the neck of the architecture and ASFF detection represents the ASFF detection head attached right before the prediction module in the head of the architecture. To summarize:

i. Variation 1: original YOLOv5 model with no variations
ii. Variation 2: represents improvements made only through the introduction of the ECA-C3 module, which replaces the C3 module in the neck of the architecture.
iii. Variation 3: represents improvements made only through BiFPN concatenation, which is again done in the neck of the architecture.
iv. Variation 4: represents improvements made only through adding the ASFF detection head before the prediction head in the head of the architecture.
v. Variation 5: represents improvements made through the combination of the replacement of the C3 module with the ECA-C3 module (in the neck) and the introduction of the ASFF detection module (in the head).
vi. Variation 6: represents improvements made through the combination of the replacement of the C3 module with the ECA-C3 module (in the neck) and the introduction of BiFPN concatenation (in the neck).
vii. Variation 7: represents final improvements made by further introducing the BiFPN concatenation, to the architecture specified in variation 5.

The exact improvements made in terms of Precision, Recall and mAP, in all 6 variations are depicted in Table 3. Each class is represented by a letter a through g, representing all classes, crazing, inclusion, patches, pitted_surfaces, rolled-in_scale and scratches respectively. Upon studying Table 3, we can make a few observations. Each component individually is helping the model learn more and perform better, except BiFPN concatenation. This particular module, when paired with the attention mechanism ECA or the ASFF module, helps them enhance their performance. As shown in the table, ECA module and ASFF module attachment help increase the map of the base model by 3% and 3.1% respectively. However, when the BiFPN concatenation is introduced to variation 2 and variation 4, the increase in mAP as compared to the base model improves by another 2.3% and 1.1% respectively. The combination of all three modules into the base model increases the mAP by a total of 6.5%. Thus, our model performs much better and has increased learning capabilities.

Variation	ECA-C3 module	BiFPN	ASFF detection
1	□	□	□
2	✔	□	□
3	□	✔	□
4	□	□	✔
5	✔	□	✔
6	✔	✔	□
7	✔	✔	✔

Variation	Precision (%)							Recall (%)							mAP
	a	b	c	d	e	f	g	a	b	c	d	e	f	g	(all)
1	72.9	51.9	74.3	84.9	76.0	63.5	86.9	77.7	51.2	91.7	93.3	70.4	66.2	93.3	78.2
2	76.6	61.4	74.6	83.6	84.6	68.8	86.8	80.5	62.0	90.8	93.3	**76.5**	66.7	93.9	81.2
3	74.2	46.9	74.0	85.2	86.2	64.3	88.3	77.3	48.2	90.5	93.3	71.6	69.0	91.5	78.2
4	79.2	59.0	77.1	86.5	**88.4**	72.1	91.9	80.8	58.7	90.5	96.0	75.4	69.0	**95.1**	81.3
5	80.0	64.6	79.4	88.1	84.2	74.2	89.4	**83.2**	**70.2**	**92.9**	**96.7**	72.6	**73.7**	92.8	83.5

6	78.3	62.3	77.4	84.2	87.7	70.1	87.7	79.6	54.5	91.7	94.0	**76.5**	69.1	91.5	82.4
7	**83.6**	**67.1**	**83.0**	**89.2**	88.3	**80.2**	**93.8**	81.3	65.8	89.3	94.0	**76.5**	71.3	90.9	**84.7**

Table 3 Quantitative comparison of the different variations proposed for the ablation study

5. Conclusion

In this research paper, we present an enhanced version of the YOLOv5 architecture tailored to address the specific challenges encountered in steel surface defect detection within the realm of modern deep learning models. Our refined model, known as AFF-YOLO, incorporates three key modifications to the base model. Specifically, we introduce the ECA (Efficient Channel Attention) mechanism module within the network's neck, alongside a BiFPN (Bi-directional Feature Pyramid Network) path aggregation network to facilitate effective feature fusion. Additionally, an ASFF (Aggregated Spatial Feature Fusion) detection head is incorporated before the detection layers in the head. These enhancements have proven instrumental in overcoming various challenges inherent to steel surface defect detection, including issues related to the small size of defects, the presence of multiple scales of similar defect types, variations in orientations, and the existence of similar features among distinct classes. By strategically integrating these modules at designated locations, AFF-YOLO demonstrates a remarkable performance boost over the base model, achieving a substantial 6.5% increase in mAP-50 (mean Average Precision at IoU 0.5). Moreover, precision across all classes witnesses an impressive improvement of 10.8%, coupled with a notable 3.6% increase in recall. Notably, AFF-YOLO maintains a remarkable real-time applicability with a detection speed of 97 frames per second (FPS), rendering it suitable for real-time detection applications. The considerable advancements achieved by AFF-YOLO in comparison to the base model underscore its effectiveness in improving detection capabilities while ensuring practical feasibility for real-world, time-sensitive scenarios.

Although our model outperforms the base model in terms of accuracy, future improvements need to be made to integrate even more lightweight and efficient modules to make the architecture computationally less expensive, while maintaining a speed sufficient for real time scenarios.

Data availability statement:

- Data Publicly available in a google drive link:

  NEU-DET dataset, after modification is available as a zip file. In case the dataset is not accessible, a roboflow link for the dataset can be requested by emailing the primary author. Additionally, weight files of every model trained during the experimentation, including the ablation study is available as a .pt file. Inference of the base YOLOv5 model and the final AFF-YOLO model on the test image set is also available in the link. Google Drive link:

  https://drive.google.com/drive/folders/1Kg0DC2HxO-Huks2LKU1I5JSn1LV-msWo?usp=drive_link

- Code repository is publicly accessible on the following Github link:

  https://github.com/Manas-Mehta/AFF-YOLOv5

Funding

The authors did not receive support from any organization for the submitted work.

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Neogi, Nirbhar, Dusmanta K. Mohanta, and Pranab K. Dutta. "Review of vision-based steel surface inspection systems." EURASIP Journal on Image and Video Processing 2014.1 (2014): 1-19.

[2] Importance of Steel. SAIL. (n.d.). https://sail.co.in/en/learning-center/importance-steel

[3] Luo, Qiwu, et al. "Automated visual defect detection for flat steel surface: A survey." IEEE Transactions on Instrumentation and Measurement 69.3 (2020): 626-644.

[4] He, Yu, et al. "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features." IEEE transactions on instrumentation and measurement 69.4 (2019): 1493-1504.

[5] Jeon, Yong-Ju, et al. "Steel-surface defect detection using a switching-lighting scheme." Applied Optics 55.1 (2016): 47-57.

[6] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012).

[7] Luo, Huilan, et al. "Small Object Detection Network Based on Feature Information Enhancement." Computational Intelligence and Neuroscience 2022 (2022).

[8] Liu, Jiayi, et al. "Defect detection for metal base of TO-Can packaged laser diode based on improved YOLO algorithm." Electronics 11.10 (2022): 1561.

[9] Li, Shun, and Xiaoqiang Wang. "YOLOv5-based defect detection model for hot rolled strip steel." Journal of Physics: Conference Series. Vol. 2171. No. 1. IOP Publishing, 2022.

[10] Anthony, Ashwin, et al. "A Review and Benchmark on State-of-the-Art Steel Defects Detection." Available at SSRN 4121951.

[11] Ultralytics (2020) YOLOv5 2020 Available from: https://github.com/ultralytics/yolov5

[12] Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "Efficientdet: Scalable and efficient object detection."

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
[13] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).
[14] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[15] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[16] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018)
[17] Liu, Wei, et al. "Ssd: Single shot multibox detector." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016.
[18] Lin, Tsung-Yi, et al. "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.
[19] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).
[20] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
[21] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
[22] He, Kaiming, et al. "Mask r-cnn." Proceedings of the IEEE international conference on computer vision. 2017.
[23] Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[24] Law, Hei, and Jia Deng. "Cornernet: Detecting objects as paired keypoints." Proceedings of the European conference on computer vision (ECCV). 2018.
[25] Duan, Kaiwen, et al. "Centernet: Keypoint triplets for object detection." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
[26] Carion, Nicolas, et al. "End-to-end object detection with transformers." European conference on computer vision. Cham: Springer International Publishing, 2020.
[27] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[28] Woo, Sanghyun, et al. "Cbam: Convolutional block attention module." Proceedings of the European conference on computer vision (ECCV). 2018.
[29] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[30] Wang, Qilong, et al. "ECA-Net: Efficient channel attention for deep convolutional neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
[31] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[32] Liu, Songtao, Di Huang, and Yunhong Wang. "Learning spatial fusion for single-shot object detection." arXiv preprint arXiv:1911.09516 (2019).