

Received 3 November 2023, accepted 13 November 2023, date of publication 16 November 2023, date of current version 22 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3333894

RESEARCH ARTICLE

Steel Strip Quality Assurance With YOLOV7-CSF: A Coordinate Attention and SIoU Fusion Approach

G. DEEPTI RAJ¹ AND B. PRABADEVI¹

School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore 632014, India

Corresponding author: B. Prabadevi (prabadevi.b@vit.ac.in)

ABSTRACT Steel strip can develop surface defects during manufacturing and processing, affecting structural integrity and usability. These defects can be caused by both internal and external factors. However, traditional manual error detection techniques do not meet today’s accuracy standards. Therefore, an improved version of the YOLOv7 algorithm for steel strip surface defect detection is proposed in this work. A lightweight and inexpensive Coordinate Attention (CA) mechanism is built into the structure of the head of YOLOv7. The SCYLLA-Intersection over Union (SIoU) loss function is used to improve detection efficiency. Furthermore, to enhance the dataset, a vertical flip augmentation technique is applied to create the optimal model: YOLOv7-CSF through fusion of CA and SIoU. It has been observed in the experimental findings that the modified YOLOv7-CSF algorithm’s mAP value in the detection is 4.09% better than that of the original YOLOv7 method, reaching 66.1% and a maximum of 96.9% accuracy in a single category of defects. The efficacy and superiority of the updated model are shown by comparing it with the recently announced YOLOv8, other steel strip datasets and other hyper-parameter tuned models, providing a novel way for daily surface defect detection on steel strips.

INDEX TERMS Coordinate attention, SIoU, YOLOv7, steel strip, defect detection.

ABBREVIATIONS

YOLO	You Only Look Once.	CV	Computer Vision.
YOLOv7	You Only Look Once version 7.	VGGNet	Visual Geometry Group Network.
SIoU	SCYLLA Intersection over Union.	SDD	Single Shot Detector.
CNN	Convolutional neural Network.	CBAM	Convolutional Block Attention Module.
R-CNN	Region based Convolutional Neural Network.	RFB	Receptive Field Block.
YOLOv2	You Only Look Once version 2.	PANet	Path Aggregation Network.
YOLOv3	You Only Look Once version 3.	NEU-DET	North Eastern University Steel Strip Dataset.
YOLOv4	You Only Look Once version 4.	GIoU	Generalized Intersection over Union.
YOLOv5	You Only Look Once version 5.	ECA	Efficient Channel Attention.
CSP	Cross Stage Partial Network.	GC10_DET	Dataset collected in the real industry.
EELAN	Extended Efficient Layer Aggregation Network.	BiFPN	Bi-directional Feature Pyramid Network.
Conv	Convolutional.	ML	Machine Learning.
mm	millimeter.	XSDDD	Steel Strip Defect Database.
SVM	Support Vector Machine.	CA	Coordinate Attention.
LBP	Local Binary Pattern.	CIoU	Complete Intersection over Union.
		ELAN	Efficient Layer Aggregation Network.
		SiLU	Sigmoid Weighted Linear Unit.
		MP	MP Convolutional Layer in YOLOv7.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo¹.

SPP	Spatial Pyramid Pooling.
SPPCSPC	CSPNet with SPP Block.
SPPF	Spatial Pyramid Pooling Fusion Network.
BN	Batch Normalization.
NLNet	Non-Local Networks.
GCNet	Non-Local Networks meet Squeeze Excitation Networks.
SE	Squeeze and Excitation attention.
2D	2 dimensional.
ICIoU	Improved Loss based on Complete Intersection over Union.
CPU	Central Processing Unit.
GPU	Graphics Processing Unit.
RAM	Random Access Memory.
GB	GigaBytes.
SGD	Stochastic Gradient Descent.
FPS	Frames Per Second.
GFLOPS	Giga Floating Point Operations Per Second.
mAP	Mean Average Precision.
P	Precision.
R	Recall.
P-R	Precision-Recall.
AP	Average Precision.
TP	True Positive.
TN	True Negative.
FN	False Negative.
FP	False Positive.
LBP	Local Binary Pattern.
HOG	Histogram Oriented Gradient.
SVM	Support Vector Machine.
NNC	Nearest Neighbor Classifier.
ROI	Region of Interest.
IoU	Intersection over Union.

I. INTRODUCTION-OBJECT DETECTION

Computer vision technology has advanced so rapidly that object recognition and object segmentation tasks are widely used in various real-world domains [1], [2], [3], [4], [5], [6]. Over the past two decades, object detection has received much attention as it is an important approach for locating and identifying objects in visual images. The rapid development of deep neural networks has greatly improved the effectiveness of object recognition technology. State-of-the-art object detection techniques based on deep learning can be classified into two main types, namely, two-stage and one-stage methods, depending on how candidate regions are generated [7].

R-CNN [8] and its derivatives, like Faster R-CNN and Mask R-CNN [9], [10], enhance object detection accuracy by combining manual feature extraction with CNN-based learning in a two-stage process. They select ROI and make category predictions for detected targets. One-stage detectors like YOLO and its derivatives immediately offer object prediction on each position of the feature maps without the need for the cascaded region classification step [11], [12],

[13]. YOLO series is ideal for real-time applications as it can instantaneously train the entire input image and perform the detection in a single neural network forward propagation. The seventh version of YOLO was made available in 2022. The model structure (CSP→EELAN), partial convolution strategy method (Conv→RepConv), and label assignment approach are the three areas where YOLOv7 differs significantly from previous versions [14].

A. STEEL STRIP DEFECT DETECTION

Defect detection and predictive maintenance are essential practices to support sustainable production. Defect detection helps reduce waste generation, optimize resource use, and minimize environmental impact by identifying and eliminating defective products early in manufacturing. By proactively addressing maintenance needs based on real-time monitoring and data analysis, predictive maintenance ensures efficient use of resources, extends equipment life and reduces waste and downtime. These practices promote sustainable production.

Steel is used as a raw material in many other industries, and its quality directly affects the final product, so in recent years, the steel industry has sought stronger and more effective quality control systems. Steel strip surfaces are used in various applications in all fields. It is often used in automotive body panels and engine parts and requires treatment to improve corrosion resistance and paint adhesion. In the construction sector, steel strip surfaces are used in structural members and are often treated with galvanization or coatings to increase durability and prevent corrosion. They are also applied to the packaging of metal cans and lids where surface treatments improve appearance and protect against corrosion. In the electrical and electronics industry, steel strip surfaces treated to improve electrical conductivity and prevent oxidation are used for components such as wires and connectors. It is also essential in metalworking, engineering and power generation, where surface treatments are used to achieve precise dimensions, optimal functional properties and resistance to high temperatures and environmental influences.

Many surface defects in a steel strip greatly affect its quality. One way to regulate quality is to implement a system that can identify these defects early on. It is imperative that this technology be non-invasive and able to detect surface defects without causing any damage to steel strips. The main drawbacks of manual inspection are:

- Time-consuming and ineffective: It takes a lot of time for the interpreter to process a large number of recognition images.
- High false and missed detection rate: Chronic fatigue, poor judgment, operator error, missing data, etc., can cause false positives and false negatives even for experts.
- Non-uniform evaluation findings: Some errors have similar definitions, or even one error can be considered two or three different types, so subjective considerations greatly affect interpretation results.

- Inaccurate assessment of defect grade: Using the already prevalent video images on subjective assessment, it is challenging for the interpreters to assess the extent of the error.

For this reason, it is better to utilize an automated inspection system that does not require human intervention (although a human inspector can be used to ensure the system is functioning properly or set it up to achieve higher precision) and to develop a technique that can automatically identify the pipeline defects, which lowers labour and time costs while increasing the efficiency and quality of the detection. As computing technology advances, artificial intelligence approaches like machine learning and deep learning are applied to defect identification for materials like metals, semiconductors, fruits, etc. Computer vision-based detection approaches outperform traditional detection methods in terms of efficiency and accuracy while reducing labour expenses. This also leads to the automation of the detection process.

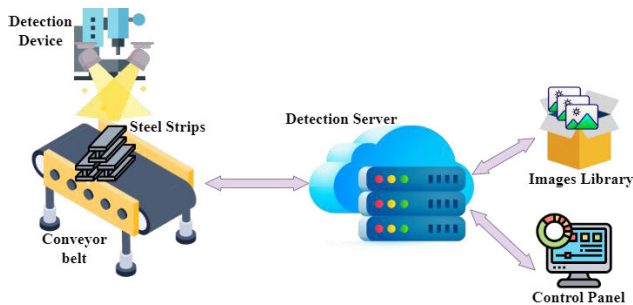


FIGURE 1. Machine vision defect detection system.

Machine vision-detecting equipment typically includes industrial cameras, light sources, protective devices, etc. The device is mounted symmetrically on the top and bottom of the steel strip. A series of industrial cameras must cover the entire steel strip, so proper placement is required. In general, seven cameras are sufficient to cover the entire surface of the steel strip. The field of view of the steel strip camera increases as the distance between the camera and the strip increases. If the surface of the steel strip is larger, then the number of cameras can be reduced. The speed of the strip moving on conveyor rollers can reach 400m/min, so industrial cameras need to record at high speeds to meet real-time needs. The images captured by the industrial cameras are transmitted to the server through optical fibre, suitable algorithms process the images on the server, and the processed images are displayed on the console panel. Fig. 1 shows how a steel strip is passed through a machine vision device, and the detection process is automated. An image library containing captured images is accessed via a server. The control panel takes over access to the collected images, which are subsequently processed for image processing, defect identification, and detection. Automatic detection aims to specify the type of defect and to use a box to indicate the location of the defect. More detailed information can be found in [15] and [16].

Many researchers have recently become interested in detecting steel strip surface defects using machine learning techniques. For the purpose of identifying steel strip defects, the k-nearest neighbour approach has been described by Karthikeyan et al. [17] and Zaghdoudi et al. [18]. The efficiency of various enhanced SVM versions for identifying surface defects in steel strips was discussed by Schleif et al. [19] and Gong et al. [20]. The LBP method was used by Liu et al. [21] to identify steel strip surface defects. The aforementioned standard machine learning techniques can achieve good results. Yet, they usually at first request feature extraction, leading to algorithms whose output is bound by the outcomes of feature extraction. Since 2014, with the advancement of deep learning technology, more and more researchers are using deep learning methods along with soft computing and computer vision to detect and label surface defects in steel strips.

The CV community has seen the emergence of a number of new architectures, including the popular GoogleNet [22], VGGNet [23], RCNN [8], Fast RCNN [9], and Faster RCNN [9]. The authors, Liang T. et.al [24], have enhanced sparse R-CNN by incorporating a coordinate attention block with ResNet and constructing a feature pyramid to modify the backbone. By utilizing this approach, they have successfully developed an enhanced model for identifying regions of interest in images. The model extracts relevant features and prioritizes important information, thereby significantly enhancing the accuracy of the detection process. In their study, Bao et al. [25] employ the ClassDecoder technique to enhance category sensitivity and improve detection performance specifically for autonomous driving applications. The distribution of object categories within particular scene backgrounds aligns with the connection between objects and the image context. Pan et al. [26] proposes an anchor-free lightweight object detector called ALODAD for autonomous driving, which incorporates an attention scheme into the lightweight neural network GhostNet and builds an anchor-free detection framework to achieve lower computational costs and provide parameters with high detection accuracy. In the proposed method, the authors also add an IoU branch to the decoupled detector to rank the vast number of candidate detections accurately and have achieved significant accuracy.

The above architecture has extensive applications in medicine, alternative energy, and self-driving cars. However, these architectures require additional training time and are implemented in multiple phases, making them unsuitable for real-time deployment. Therefore, lightweight architectures are necessary to improve detection accuracy and optimize inference speed.

Following the above premise, CV researchers are keen to develop new lightweight architectures that can be deployed on edge devices while utilizing limited computing resources. The most popular designs that focus on speed and accuracy for edge devices in this regard are SSD, MobileNet [27], and YOLO. The YOLO architecture uses two fully connected

layers, making it easier to deploy than differently-sized convolutional layers in SSD networks. YOLO has evolved through various iterations, such as YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6 and most recently YOLOv7. The author's Zheng et al. [28] focus on inspecting industrial products in semiconductor, steel and fabric manufacturing processes. They investigated recent advances in deep learning-based inspection algorithms. They presented their applications in the steel, fabric, and semi-conductor industries and provided information on publicly available datasets containing surface image samples to facilitate the research on deep learning-based surface inspection. The use of deep learning (DL) in intelligent machining and tool monitoring for smart manufacturing is explored by Nasir et al. [29]. Furthermore, Various DL models, including autoencoders, deep belief networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs), are examined along with their applications in this field. A novel defect detection approach based on K-nearest neighbour (KNN) and Euclidean clustering segmentation to identify the surface defects of lithium batteries is proposed by Liu et al. [30], and an industrial application example of lithium battery production is demonstrated, which meets the industrial application requirements. Table 1 below shows the detection of strip defects by YOLO in recent years.

TABLE 1. Existing YOLO-based steel strip defect detection.

Ref	Model Name	Dataset	Attention Mechanism	Loss Function
[31]	YOLOv4	NEU-DET	CBAM	IoU
[32]	YOLOv4	Iron- Defects	Not Used	IoU
[33]	YOLOv4	Chips-Defects	SA	IoU
[34]	YOLOv5	NEU-DET	Not Used	GIoU
[35]	YOLOv5	Memory modules	Not Used	IoU
[36]	YOLOv5	NEU-CLS	CBAM	IoU
[37]	YOLOv7	NEU-DET, GC10-DET	ECA	SIoU

Small target detection is a challenge for deep learning-based approaches, and efficient model deployment to mobile and embedded devices requires lightweight models that are trainable, effective and have fast detection speeds. Understanding the effectiveness of object detection that leads to predictive maintenance is one of the major challenges in computer vision tasks that require the specification of a loss function. A loss function measures how well an ML model can predict the predicted outcome. Achieving standard accuracy requires fine-tuning the hyper-parameters. Especially due to recent technological developments, YOLO seems to be the technology that has received the most research attention. From Table 1, we can see that most strip surface defect detection is based on the NEU-DET and GC10-DET datasets. The YOLO version that was actively experimented with in this work was YOLOv7, which was the most recent version at the time this work was carried out. YOLOv7 is trained on COCO dataset that contains 80 classes. The current study

aims to provide a comprehensive solution to another steel strip dataset i.e., on the XSDD dataset, by improving the YOLOv7 baseline to balance accuracy and detection time. The weights used in this study are pre-trained weights. Based on the YOLOv7 method, a lightweight YOLOv7-CSF is introduced by including the CA module and SIoU loss function. An optimized model is then built by adding a vertical flip data augmentation technique to address the problems with small target detection. The experimental results show that the proposed strategy outperforms YOLOv7 on the XSDD dataset, highlighting its potential as a significant advancement in object detection methods.

The contributions of this study are enumerated as follows in the context of our experiments:

1) A CA mechanism is embedded to increase the detection accuracy. The CA mechanism allows the networks to focus more on errors by acting as a lightweight and cost-effective attention strategy.

2) In order to accelerate network convergence, increase detection effectiveness, address the issue of dataset imbalance, and lessen the detection of false detection in steel strips, an updated loss function for YOLOv7, SIoU is utilized rather than the standard CIoU. Penalty metrics were revised in SIoU to take the desired regression's vector angle into consideration.

3) Fine-tuning the hyper-parameters leads to better model performance, fewer errors, better results, and optimization. The model is trained with the vertical flipping data augmentation technique.

An enhanced YOLOv7-CSF is proposed by combining the CA mechanism, SIoU loss function with the YOLOv7 architecture and with the vertical flipping data augmentation technique for balancing the classes. The remaining sections of this study are organized as follows: the second section describes methodology with the background of YOLOv7 detection framework, coordinate attention mechanism, SIoU loss function, fine-tuning hyper-parameters and presented YOLOv7-CSF model. The third section then introduces the experimental dataset, evaluation indicators and metrics. Test results are described in the fourth section. Finally, conclusions are drawn and recommendations for further research are made.

II. METHODOLOGY

This section provides insights into the background of YOLOv7 network, the functioning of Coordinate attention module, SIoU loss function, and understanding of optimization with fine tuning of hyper-parameters.

A. BACKGROUND OF YOLOV7 NETWORK

Alexey Bochkovskiy created the latest YOLO object detection model, YOLOv7 [38]. This architecture is faster and more accurate than all previous iterations. The authors mainly made two contributions: (1) their ultimate aggregation layer, E-ELAN, which is an improved version of the ELAN computational block; and (2) an innovative method for scaling

models that allows for parallel scaling of model depth and width by concatenating layers.

The input, backbone, and head networks are the three components that make up the YOLOv7 network. The image was first preprocessed by the YOLOv7 network then scaled to $640 \times 640 \times 3$ and fed the backbone network as input. Similar to YOLOv5, SiLU was used as the activation function in YOLOv7. Fig. 2 shows the architecture of YOLOv7 with added CA module in the head part.

Fig. 3 shows the structure of the ELAN, RepConv and SPPCSPC modules. In YOLOv7, the ELAN module, which was made up of various convolutions, was added. Expand, shuffle, and merge cardinality of the image features are utilized to continuously improve the network's learning ability without erasing the original gradient route, thus boosting the network's accuracy. ELAN modules offer scalable and high-performance solutions for local area networks, incorporating features such as redundancy and centralized management. These capabilities elevate network reliability and adaptability to new levels.

The upper branch of the MP module reduced the length and width of the feature map in half by using the max-pooling operations and convolution. The channels were split in half by the first convolution on the lower branch, and the length and width were also cut in half by the second convolution on the feature map with 3 size kernel and a 2 stride. The upper and lower branches were joined and the final product was a feature map with half output length, half width, and equal input and output channels.

Convolution was applied with a kernel size of $1 (1 \times 1)$ and 3 layers. The main network consists of an SPPCSPC module, some CBS modules, an MP module, a CatConv module and three consecutive RepConv modules. Just as YOLOv5 uses the SPPF module, the SPPCSPC module extends the receiving field of the network. While preserving the integrity of feature map size, the SPPCSPC module may capture multi-scale object data. A more standardized model with a new parameterized structure, the RepConv structure, was developed in YOLOv7 [39]. The precision of computations is enhanced by RepConv without requiring additional computational resources. Furthermore, RepConv does not impose restrictions on the original convolution modules in terms of their type, quantity, numerical precision, and specific parameters.

The RepConv module updated the entire output channel to create a bbox prediction task, class, and objectivity results for image recognition. It enhanced the inference effect and increased the training time [40]. During training, the entire module is split into several identical or different module branches. Then 3×3 convolutional BN, 1×1 convolutional BN and BN layers were added to the training model. During inference, the three components were re-parameterized, using a 3×3 convolutional output to transform their parameters into another set of parameters equivalent to them. A fast single-branch inference model was built from the multi-branch training model. The SPPCSPC module is more

efficient than other methods as it reduces the number of parameters and calculations needed for feature extraction. One can boost the precision of detecting small objects by maintaining the sensing field of the model intact. This allows for accurate localization of various small targets, regardless of their sizes. It is lightweight and seamlessly integrates into existing models with minimal modifications needed. In addition to maintaining the high efficiency and other good features of the multi-branch model, YOLOv7 can improve network performance by balancing speed and accuracy.

B. COORDINATE ATTENTION MECHANISM

MODULE-PAYING SPECIFIC ATTENTION TO INFORMATION

Non-local/self-attention networks have recently received a lot of attention due to their ability to develop spatial or channel-wise attention. Examples of common networks that collect many types of spatial data via non-local techniques include NLNet [41], GCNet [42], A2Net [43], ScNet [44], GSoP-Net [45], or CCNet [46]. CIFAR-100 [47], and self-attention modules [48] are commonly used in big models, but are not suitable for mobile networks due to their computational complexity. In deep neural networks, the weights are the resources that are allocated as part of the attention mechanism. The following attention mechanisms are common in the area of vision: spatial domain [49], [50], [51], channel domain [52], [53], and mixed domain [54], [55], [56]. The SE channel and the dependencies between channels, that are moderately typical in the field of vision. The CBAM, an extension of SE, now includes a two-dimensional spatial attention matrix. CA transforms channel attention into two one-dimensional feature encoding processes that associate features in two spatial directions. One spatial direction can be used to store remote dependencies, while the other can be used to store precise location information. The created feature maps are then formulated as a pair of location-sensitive attention maps and direction-aware attention maps, respectively.

A 1×1 convolutional layer is added after the global average pooling layer and the fully connected layer is removed. ECA is an enhanced version of SE that successfully captures cross-channel interactions without dimensionality reduction. Convolutions can only model short-range dependencies, which are inadequate for vision tasks, and can only capture local connections. For mobile networks, the computational cost incurred by the majority of attention mechanisms is too expensive.

A powerful new attention mechanism known as “coordinated attention” was first described in 2021. The CA mechanism outperforms existing lightweight attention approaches (e.g., SE, CBAM) by factoring the 2D global pooling operations into two 1D encoding processes. This allows for the capturing of long dependencies along with one spatial dimension while maintaining the accuracy of data localization information with other one. The two direction-aware and position-sensitive attention maps are then created separately from the output feature maps and can be used to improve

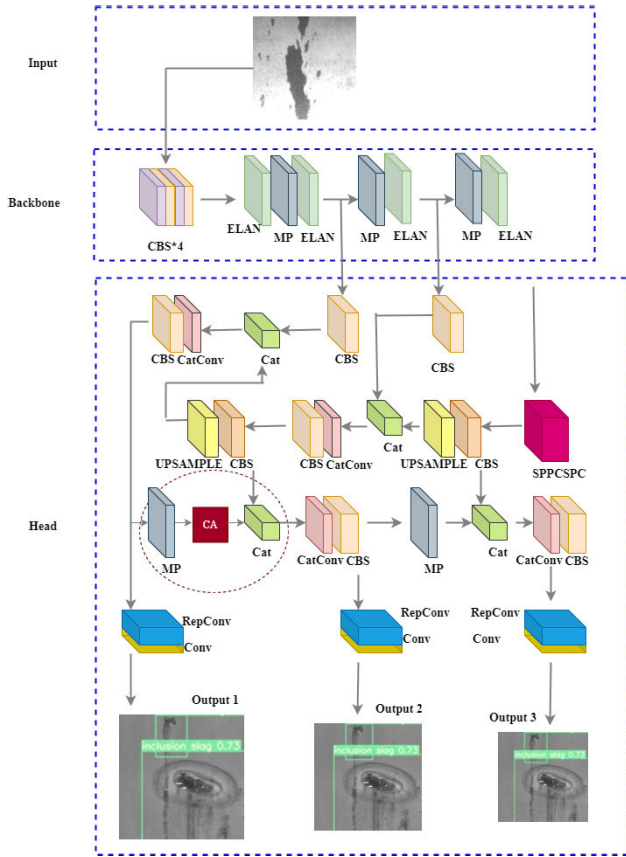


FIGURE 2. YOLOv7 network structure with added CA module.

the representations of the target objects in the input feature map. CA can be easily integrated into well-known mobile networks like MobileNetV2, MobileNeXt, and EfficientNet with almost no additional computational cost. The advantages of CA are as follows.

- Collects cross-channel data and direction- and position-sensitive data, which help models more accurately find and identify objects of interest.
- As a pre-trained model, CA can significantly improve task performance when using downstream cellular networks, especially for dense prediction problems (e.g., semantic segmentation).
- The composition block diagram of the CA mechanism is shown in Fig. 4. Coordinate information embedding and attention generation [52], are the two phases in which the CA mechanism encodes channel correlation and coordinates information.

The average value of each line and column of a channel is calculated using one of the two formulations below given a feature map X :

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \leq i \leq W} x_c(h, i) \quad (1)$$

$$Z_c^W(W) = \frac{1}{h} \sum_{0 \leq j \leq h} x_c(j, W) \quad (2)$$

where x_c designates the c -th channel of the feature map, z_h is the output of the transform at height h and has the shape (c, W) , and z_w is the output of the transform at width w and has the shape (c, h) and W and H stand for the feature map's individual width and height.

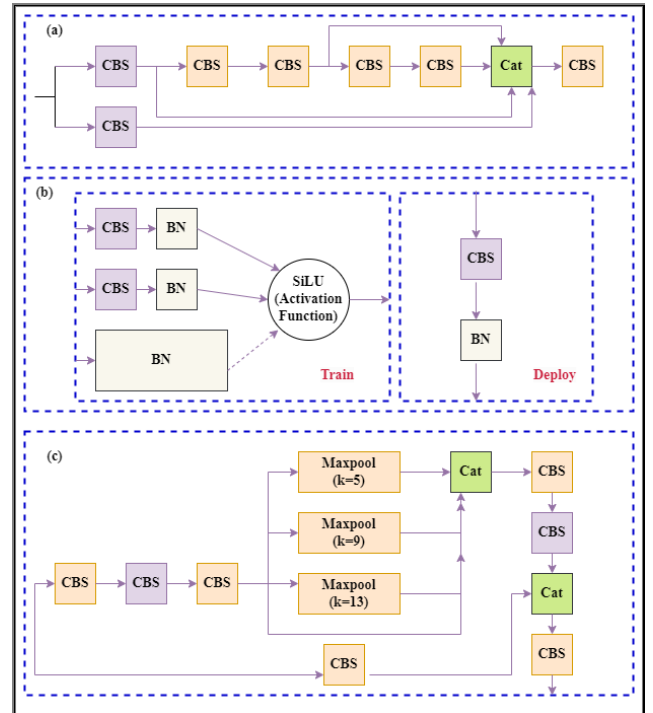


FIGURE 3. Structure of (a) ELAN module (b) RepConv module and (c) SPPCSPC module.

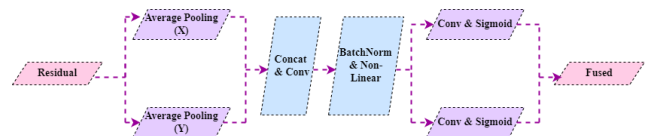


FIGURE 4. Coordinate attention mechanism composition block diagram.

The dimension of the combined feature map is compressed by a 1×1 convolution layer followed by a ReLU activation layer, which can be written as:

$$f = \text{ReLU}(\text{Conv}_{1 \times 1}(\text{concat}(Z^h, Z^W))) \quad (3)$$

where concat refers to the manipulation of concatenation and f is the shape $(C/r, h+W)$. After that, f is divided into the tensors $f^h \in \mathbb{R}^{C/r \times h}$ and $f^w \in \mathbb{R}^{C/r \times W}$. Two 1×1 convolution layers are defined as follows for f_h and f_w , respectively, in an effort to restore them into the same shape as z_h and z_w .

$$g^h = \sigma(F_h(f^h)) \quad (4)$$

$$g^w = \sigma(F_w(f^w)) \quad (5)$$

Here, F_h and F_w are the convolution manipulations for f_h and f_w independently, and σ is the sigmoid activation function. Expanded feature maps for the horizontal and

vertical coordinates, respectively, are employed as attention weights by first obtaining feature maps g_h and g_w .

Reweighting each value on the original feature map x is described as follows:

$$y_c(i, j) = y_c(i, j) X g_c^h(i) X g_c^w(j) \quad (6)$$

where y_c is the c^{th} channel feature map generated by the attention block.

The utilization of CA holds great significance in defect detection as it empowers the model to concentrate on particular spatial regions within an image. The CA integration to the base model assists in identifying areas of focus that have a higher probability of containing defects. This ultimately enhances the accuracy of the model in detecting and localizing defects. By enabling the model to focus on the coordinates or locations where defects are more susceptible to manifest, CA improves the model's ability to accurately and swiftly identify defects in images. The structure of the embedded CA module in the neck of YOLOv7 is shown in Fig. 5. The CA module is inserted after the MP block instead of the Conv module, as seen in the Fig. 5b. The CA mechanism makes it possible for the neural network to focus on valid coordinates while suppressing invalid coordinates, thus improving the efficiency of information flow. This reduces the number of parameters and GFLOPS in our experiment.

C. SIOU LOSS FUNCTION

Estimated and ground truth box's aspect ratio, overlap area, distance, and other bounding box regression metrics, form the basis of conventional object identification loss functions (i.e. GIoU, CIoU, ICIoU etc). According to Gevorgyan et al. [57], the addition of SIOU (SCYLLA-IoU) greatly facilitates the training process because it makes the prediction box to drift to the nearest axes quite quickly, requiring only the regression of one coordinate, either X or Y.

Equation (7) depicts the loss function, which has three parts: Bounding box regression loss function L_{box} , classification loss function L_{cls} , and confidence loss function L_{obj} .

$$LOSS_{Fun} = L_{obj} + L_{cls} + L_{box} \quad (7)$$

The bounding box regression loss function in the YOLOv7 source code is CIoU. The only dimensions that CIoU can consider are the overlap area, centroid distance, and aspect ratio of the real frame and the predicted frame. Since SIOU can better reflect the variations in width, height, and confidence level, it was chosen over CIoU.

The SIOU [57] loss function mainly includes the following four parts: Angle cost Λ , Distance cost Δ , Shape cost Ω , and IoU cost. Angle cost is defined by equation (8).

$$\Lambda = 1 - 2^* \sin^2(\arcsin(x) - \frac{\pi}{4}) \quad (8)$$

where $x = \frac{c_h}{\sigma}$, σ is the distance between the ground truth box and the prediction box's center point, c_h is the vertical distance between the prediction box's and the ground truth

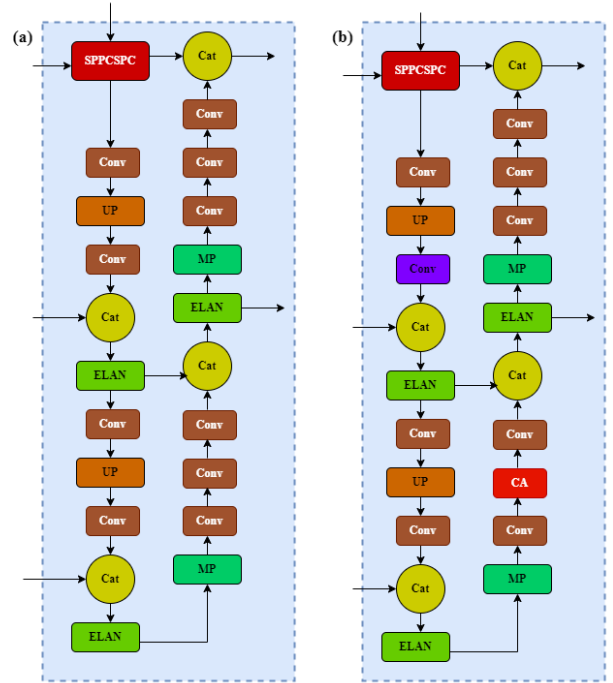


FIGURE 5. Structure of (a) YOLOv7 head, (b) YOLOv7+CA head.

box's center points. Equation(9) defines the distance cost based on the angle cost.

$$\Delta = 2 - e^{(\Lambda-2) \times \left(\frac{c_h}{c_{h1}}\right)^2} - e^{(\Lambda-2) \times \left(\frac{c_w}{c_{w1}}\right)^2} \quad (9)$$

$$x = \frac{\max(b_{cy}^{gt}, b_{cy}) - \min(b_{cy}^{gt}, b_{cy})}{\sqrt{(b_{cx}^{gt} - b_{cx})^2 + (b_{cy}^{gt} - b_{cy})^2}} \quad (10)$$

Equation (10) gives the average distance between the horizontal and vertical coordinates of the ground truth box and centre point of the prediction box. c_w is the horizontal distance between the ground truth box and the prediction box's center point, the prediction box's center point's horizontal and vertical coordinates are b_{cx} and b_{cy} ; the ground truth box's center point's horizontal and vertical coordinates are b_{cx}^{gt} and b_{cy}^{gt} ; Equation (11) defines shape cost as follows:

$$\Omega = \left(1 - e^{-\frac{|w-w^{gt}|}{\max(w, w^{gt})}}\right)^\theta + \left(1 - e^{-\frac{|h-h^{gt}|}{\max(h, h^{gt})}}\right)^\theta \quad (11)$$

h^{gt} and w^{gt} are the ground truth box's width and height, w and h are the prediction box's width and height.

The genetic algorithm calculates the value of θ , which establishes how much to focus on the shape's cost. The value varies between 2 and 4 in different data sets. Equation (12) defines the IoU cost as follows:

$$IoU = \frac{A \cap B}{A \cup B} \quad (12)$$

where A is the prediction box's area and B is the area of the ground truth box.

Equation (13) is obtained as the concluding loss function SIOU.

$$SIOU = 1 - IoU + \frac{\Omega + \Delta}{2} \tag{13}$$

SIOU significantly improves the model’s precision in identifying defects of varied sizes. This is especially valuable when dealing with defect detection tasks that involve varying scales of defects within images, ultimately enhancing the precision and robustness of the base model. In our investigation, it is observed that embedding the SIOU loss function improves training accuracy.

D. OPTIMIZATION WITH FINE-TUNING HYPER-PARAMETERS

An example of an optimization problem is the tuning of machine learning models. To tune the hyper-parameters of a learning algorithm, one must specify a set of ideal values and then apply the tuned method to any set of data. All parameters that the user can arbitrarily set before starting the training are called hyper-parameters. Hyper-parameters control the original structure of the model. Setting the hyper-parameters is necessary to determine the minimum (for example, loss) or maximum (for example, precision) of the function.

In this study, vertical flipping is taken as a hyper-parameter for our model. By completely inverting the rows and columns of pixels in an image, vertical flip augmentation is achieved. As a result, the image along the x-axis will be upside down. The input image is rotated vertically 180 degrees when it is vertically flipped. The total number of input images for the model increases to 2292 when the vertical flip is applied during the pre-processing stage on our dataset (total images = 1360). YOLOv7-CSF: An optimized YOLOv7-CSF is created by adding CA module, taking the SIOU loss function (S), and adapting the hyper-parameter vertical flip (F). The modified YOLOv7-CSF framework is shown in Fig. 6.

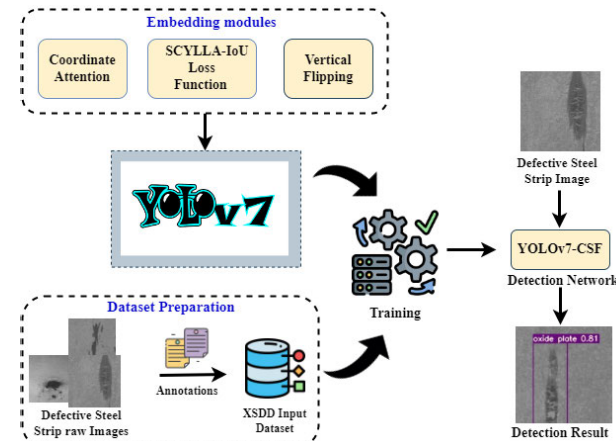


FIGURE 6. Modified YOLOv7-CSF framework.

III. INTRODUCTION TO DATASETS, EXPERIMENT CONFIGURATION AND METRICS

This section gives the overview of XSDD dataset, experimental configuration, metrics and indicators.

A. XSDD DATASET

The XSDD dataset, used in this study, is available at <https://github.com/Fighter20092392/X-SDD-A-New-benchmark>. The steel surface XSDD dataset has seven different types of defects, including ash sheet (As), inclusion slag (Is), iron red (Ir), oxide plate (Op), oxide temperature (Ot), roll printing (Rp), and scratch surface (Ss). Raw images of the steel strip can be found at the dataset link.

TABLE 2. Images distribution of XSDD dataset.

Category	Number
Ash sheet	122
Inclusion slag	238
Iron red	397
Oxide plate	63
Oxide temperature	203
Roll printing	203
Scrathces	134
Total	1360

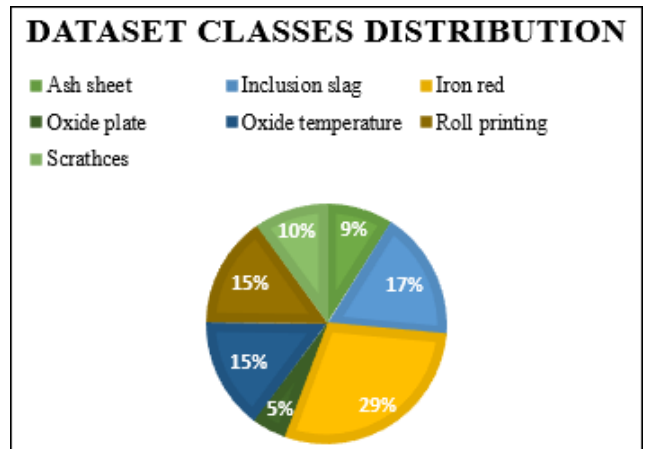


FIGURE 7. Dataset classes distribution.

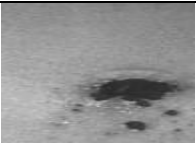
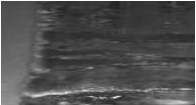
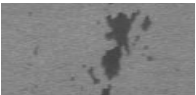
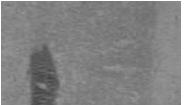

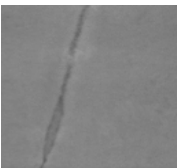

The annotation of the defective images with bounding boxes is done using the developer tool ROBOFLOW. Table 2 and Fig. 7 display the distribution of images in the XSDD dataset. The below Table 3 is a description of each XSDD dataset defect category in detail.

B. EXPERIMENTAL CONFIGURATION, EVALUATION METRICS AND INDICATORS.

The experimental evaluation setup is shown in Table 4 with the hardware environment and software versions and the evaluation index information is shown in Table 5. The processing platform is a desktop computer running the Windows 10 operating system. CPU: Intel(R) Core(TM) i5-1035G; memory: 12GB. The Google Colaboratory notebook with an NVIDIA T4 Tensor Core GPU, Python version 3.7.13, and the Torch framework version 1.11.0+cu113 was used as the implementation platform for this work.

Precision, recall, and mAP serve as the primary evaluation metrics employed to assess the model’s capability

TABLE 3. Elaboration of XSDD dataset defects.

Defect Type	Defect Image	Description
Ash sheet		Inclusions of ash or dust show up as missing areas of steel in the casting. This defect indicates that the mold may contain ash particles.
Inclusion slag		In the welding of steel, the flux coating leaves behind slag, which is mostly a de-oxidation byproduct of the interaction between the flux, air, and surface oxide.
Iron red		Steel strip, having this defect, cracks when being worked in the cold state. The presence of too much sulphur is the cause of this defect.
Oxide plate		This form of defect in steel strips results from the high temperature and high-speed rolling process, which is caused by the roller table's passive rotation.
Oxide temperature		This defect is brought on by poor stand water utilization and excessively high-temperature control during rough rolling.
Roll printing		A steel metal stock is rolled through rollers to thin out and uniformly distribute thickness. Cold rolling, the rolling is done at a temperature below the recrystallization point while hot rolling is used or processes carried out above the recrystallization point.
Scratch surface		This defect is a straight line cut parallel to the steel's rolling direction. It results from a foreign object coming into contact with the coil surface while in the pass line of a rolling mill, pickling line, etc., and leaving a scratch.

to accurately classify or detect objects/defects. These metrics ensure a balance between identifying true positives and avoiding false positives, thus enabling an all-encompassing quantification of the model's accuracy and resilience [58], [59]. The other evaluation metrics used in this study to show that the updated YOLOv7 model performs better include mAP, latency, FPS, parameters, GFLOPS, and others. The terms Fps, GFLOPS, latency, and parameters refer to the

TABLE 4. Experimental evaluation setup.

Hardware & Software	Configuration Parameter	
Computer	Operating System	Windows 10
	CPU	Intel(R) Core(TM) i5-1035G.
	GPU	Google Colaboratory GPU-NVIDIA Tesla T4
	RAM	32 GB
Software	Python-3.7.13 and torch framework-1.11.0+cu113.	

TABLE 5. Evaluation indices.

Parameter	Value
Batch	16
Epochs	100
Input Image Size	640*640 pixels
Optimizer	SGD
Weight Decay	$5*10^{-4}$
Momentum	$937*10^{-3}, 92*10^{-3}$
Weights	yolov7.pt
Initial, Final learning rate	$1*10^{-2}, 1*10^{-1}$

number of images processed per second, 1 billion floating point operations executed per second, the time required for inference, and the total number of parameters, respectively. By computing the region covered by the precision, equation (14) and recall, equation (15) curves (P-R curves) using the coordinate axes, AP values, and equation (16) for each category are obtained. The mAP, equation (17) is then calculated by averaging the AP scores of the seven different classes. The following is a list of performance metrics:

$$P = \frac{TP}{FP + TP} \quad (14)$$

$$R = \frac{TP}{FN + TP} \quad (15)$$

TP denotes that the model predicts positive cases and that the sample's actual class is positive; FN predicts negative results despite the fact that the example should be positively classified; Although the prediction is positive, FP indicates that the true class of the sample is negative.

$$AP = \int_0^1 p(R) dR \quad (16)$$

$$mAP = \frac{1}{k} \sum_{i=1}^k AP(i) \quad (17)$$

FPS is a metric used in real-time applications to measure how fast a model can process data. Equation (18) is used to estimate the FPS, while equation (19) determines the F1 Score.

$$FPS = \frac{\text{No of frames}}{\text{Total detection time(s)}} \quad (18)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (19)$$

IV. RESULTS

Table 6 compares the mAP, Precision, Recall, and F1 scores of several models. The detection Inference is represented as DI in Table 6 and 9. In YOLOv7-CA, the coordinate attention mechanism is integrated into the network structure. YOLOv7-SIoU uses the Siou loss function instead of the traditional Ciou, in YOLOv7-CA+Siou, CA and Siou are combined, and YOLOv7-CSF is formed by taking CA, Siou, hyper-parameter vertical flipping. As we can see our modified YOLOv7-CSF model increased the average accuracy by 4.09% compared to the original YOLOv7 model. This resulted in a 5.78% recall increase. The F1 score of YOLOv7-CSF is 67.88%. Although the precision value of YOLOv7-CA is higher at 75.1%, other indicators of YOLOv7-CSF were much better. When the detection inference is calculated with a confidence threshold of 0.25, it can be seen that the original YOLOv7 model, YOLOv7-SIoU, has detection inferences of 11.8ms and 11.9ms, respectively, while YOLOv7-CSF has a detection inference of 12.0ms. Although the inference has been slightly raised and the F1 score is little bit lowered, it can still detect steel strip defects in real-time engineering. The confusion matrix summarizes the results of a classifier. The number of accurate and incorrect predictions for each class is expressed as count values.

TABLE 6. Metrics comparison for different models.

Model Name	mAP @0.5 (%)	Precision (%)	Recall (%)	F1-Score (%)	DI (ms)
YOLOv7	63.5	74.9	62.2	67.9	11.8
YOLOv7-CA	63.6	75.1	57.1	64	12.1
YOLOv7-SIoU	64.1	74.2	58.5	65.4	11.9
YOLOv7-CA+Siou	63.6	65.4	62.4	63.8	12.2
YOLOv7-CSF	66.1	70.1	65.8	67.88	12.0

TABLE 7. Detection effect of YOLOV7-CSF on each category.

Class	Precision (%)	Recall (%)	mAP (%)
ash sheet	74.1	65.1	67.5
inclusion slag	85.2	93.9	97.4
iron red	52.4	61.3	58.1
oxide plate	99	54.2	57.4
oxide temperature	25.1	24.5	16.3
roll printing	80.2	61.6	69.1
scratch surface	74.5	100	96.9

Figures Fig.8a and Fig.8b show the confusion matrix of YOLOv7 and YOLOv7-CSF. The defect detection performance for every class of defect on the X-SDD dataset is shown in Table 7. According to Table 7, the scratch surface has a 100% recall and 96.9% mAP rate, while the oxide

TABLE 8. Comparison of model parameters.

Model	Parameters (M)	FPS (s)	GFLOPS (s)
YOLOv7	37.22	84.7	105.2
YOLOv7-CA	35.10	82.6	40.6
YOLOv7-SIoU	37.22	84.03	105.2
YOLOv7-CA+Siou	35.10	81.96	83.33
YOLOv7-CSF	35.10	83.33	40.6

plate has the highest precision value. The precision, recall, and mAP values for the oxide temperature class are lower-25.1, 24.5, and 16.3, respectively. This indicates that the model cannot accurately detect this particular class. Table 8 compares the parameters of the different models. By adding the CA mechanism, the parameters and GFLOPS are reduced by almost 5.69% and 61.4%, respectively. The FPS rate of YOLOv7-SIoU is 84.03s which is almost the same as of the original YOLOv7 84.7s.

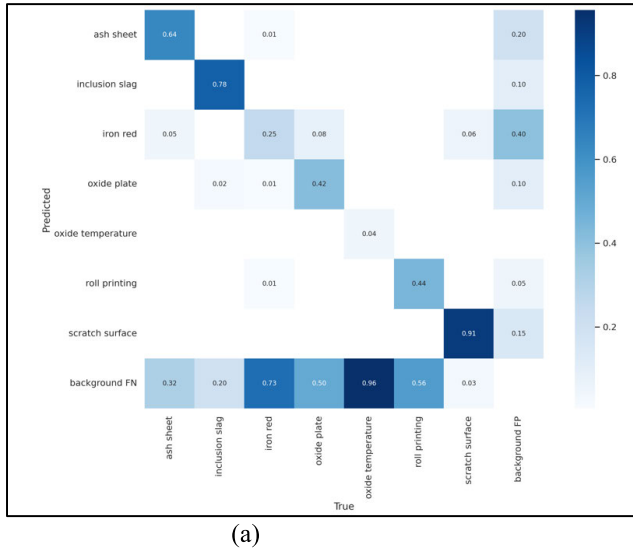
To exemplify the appropriateness of our model, a comparison of other hyper-parameters is made. Momentum and HYP are the hyper-parameters discussed in this work. The gradient descent optimization method has a characteristic called momentum which allows the search to navigate flat areas of the search space and avoid noisy gradient oscillations. Momentum in YOLOv7 has a default value of 0.937. Hyper-parameters like warmup epochs, class loss gain, object loss gain, and image copy-paste are denoted by the acronym HYP. The default settings in YOLOv7 for the above hyper-parameters are 3, 0.3, 0.7, and 0.0, respectively. We selected warmup epochs=2.5, class loss gain=0.25, object loss gain=0.6, and copy-paste=0.1 for adjusting YOLOv7 to YOLOv7-CSF+HYP.

A comparison of YOLOv7-CSF with different hyper-parameters is shown in Table 9. The YOLOv7-CSF+Momentum+HYP model has updated hyper-parameters and a momentum value of 0.92. The YOLOv7-CSF+HYP model is a CSF model with only updated hyper-parameters. Table 9 shows that, in terms of training accuracy, all of the YOLOv7-CSF+ Momentum+HYP, YOLOv7-CSF+HYP, and YOLOv7-CSF+Momentum models outperformed the basic YOLOv7 model by a range of 0.6% to 3.3%. The proposed YOLOv7-CSF outperforms all previous models with a detection accuracy of 66.1%.

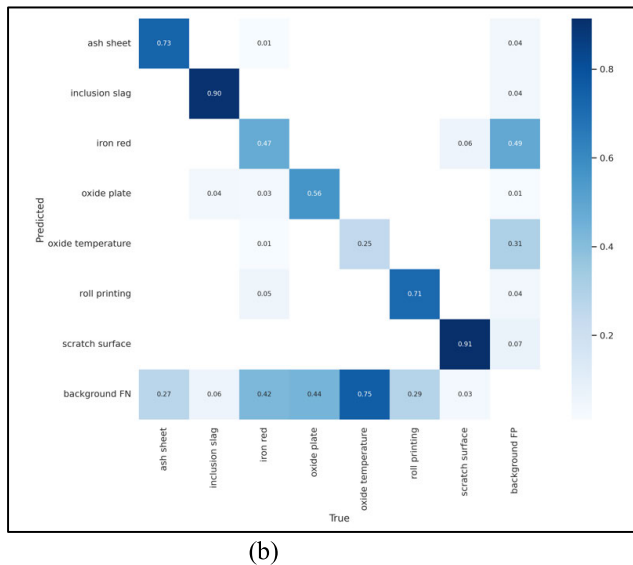
The visual detection results of XSDD dataset are shown in Fig. 9.

Fig. 10 show the results values graph for the YOLOv7-CA and YOLOv7-CSF on the XSDD dataset. Figs. 11 (a) and (b) show the P-R curves of YOLOv7-SIoU and YOLOv7-CSF, respectively.

Validation of the model can only show that the modified approach presented in this study is successful compared with the traditional image processing techniques Local Binary Pattern (LBP), Nearest Neighbor Classifier (NNC), Histogram Oriented Gradient (HOG), Support Vector Machine (SVM)



(a)



(b)

FIGURE 8. Confusion matrix of (a) YOLOv7, (b) YOLOv7-CSF.

TABLE 9. Comparison of yolov7-csf with other hyper-parameters.

Model	mAP @0.5 (%)	Precision (%)	Recall (%)	F1-Score (%)	DI (ms)
YOLOv7	63.5	74.9	62.2	67.9	11.8
YOLOv7-CSF+ Momentum + HYP	63.9	69.5	60.7	64.8	12.0
YOLOv7-CSF+HYP	65.6	66.2	64.5	65.3	12.0
YOLOv7-CSF+ Momentum	65.6	68.2	64.7	66.6	12.1
YOLOv7-CSF	66.1	70.1	65.8	67.88	12.0

and the recently published YOLOv8 model. From the table 10 we can see that the proposed YOLOv7-CSF attains high mAP, Recall when compared to other models on XSDS dataset.

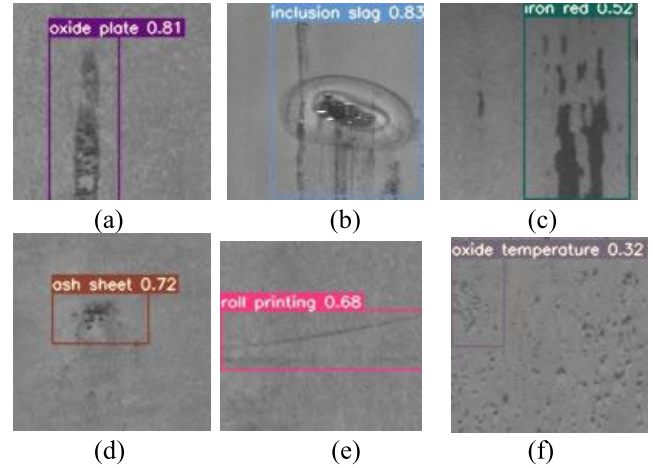


FIGURE 9. Visual detection results on XSDS Dataset. In sequence, the pictures are: (a) oxide plate(Op), (b) inclusion slag (Is), (c) iron red (Ir), (d) ash sheet (As), (e) roll printing(Rp) and (f) oxide temperature (Ot).

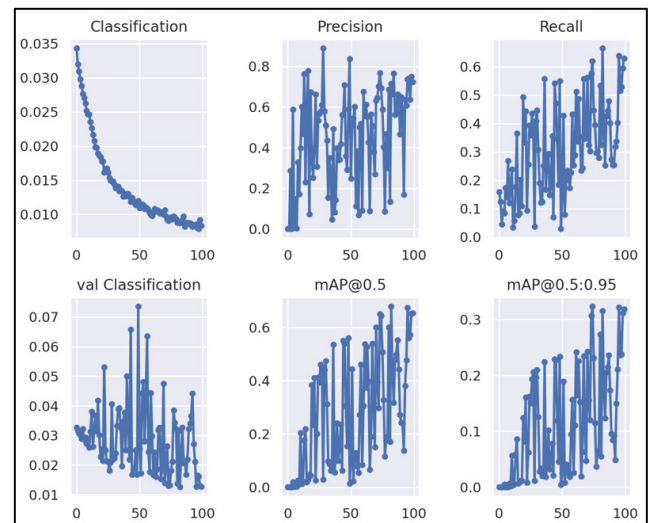


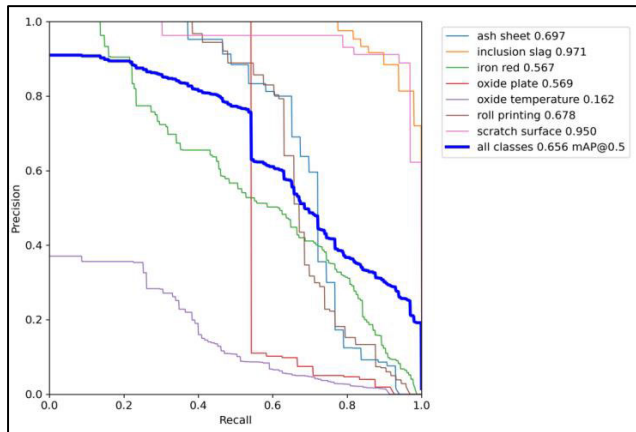
FIGURE 10. Results graph for YOLOv7-CSF.

TABLE 10. Metrics comparison between different models.

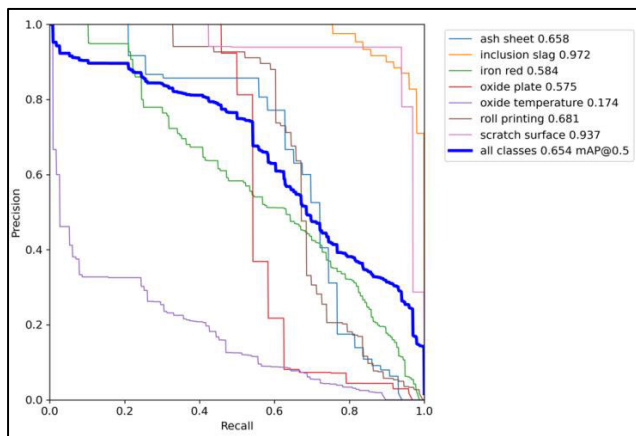
Model	mAP (%)	Precision (%)	Recall (%)	F1-Score (%)
LBP+NNC	43.2	40.0	41.2	40.5
HOG+SVM	57.4	58.0	55.8	56.8
YOLOv7	63.5	74.9	62.2	67.9
YOLOv7-CSF	66.1	70.1	65.8	67.8
YOLOv8	62.8	73.4	57.9	64.7

Also, comparing our proposed model to other datasets is crucial for evaluating its ability to generalize and perform well across various data sources.

The effectiveness and versatility of our proposed model can be observed in the results showcased in table 11. Regardless of the data environment, it has consistently proven to perform successfully on different datasets, highlighting its robustness.



(a): P-R Curve of YOLOv7-CSF+Momentum



(b): P-R Curve of YOLOv7-CSF

FIGURE 11. (a): P-R Curve of YOLOv7-CSF+Momentum (b): P-R Curve of YOLOv7-CSF.

TABLE 11. Comparison between different datasets.

Model	mAP(%) of Datasets		
	XSSD	NEU-DET	GC10-DET
YOLOv7	63.5	60	62
YOLOv7-CSF	66.1	63.8	62.9

Manufacturers benefit from defect detection and predictive maintenance by maintaining product quality, reducing costs, improving productivity, meeting regulatory requirements, and improving customer satisfaction. Reducing manufacturing costs, eliminating defective products, reducing recalls and complaints, streamlining procedures, providing suggestions for process improvement, and ensuring regulatory compliance are added advantages. Defect detection ultimately helps build a good brand reputation and competitive advantage, leading to sustainable manufacturing. Over the last few years, advanced deep learning-based computer vision algorithms are revolutionizing the manufacturing field. Thus, several industry-related hard problems can be solved by training these algorithms, including defect detection in various

materials. Therefore, identifying steel surface defects is considered one of the most important tasks in the steel industry. According to the comparison and analysis of the set of experiments in this paper, the improved YOLOv7 algorithm proposed in this study shows remarkable advantages in detection accuracy. Autonomous Vehicles, Surveillance & Security, Retail & Inventory Control, Manufacturing & Quality Control, Healthcare, Agriculture, Robotics & Drones, Sports Analytics, etc. Are some of the industries where our proposed YOLOv7-CSF can be used.

V. CONCLUSION

This study proposes YOLOv7-CSF, which is an improved version of the basic YOLOv7 model, to identify steel strip defects with complex backgrounds in the XSSD dataset. CA mechanism is combined in the head region of YOLOv7 to improve the ability of the feature graph to express itself and SIOU loss function is used to determine the gap involving the actual box and the predicted box for speeding up the network’s convergence. In addition, the vertical flipping augmentation technique is added to fine-tune the model. Experimental results show that by incorporating the above tactics, the updated model, known as YOLOv7-CSF, increases recall and mAP@0.5 by 5.78% and 4.09%, respectively, compared to the original network. For a particular defect category, scratch surface, there is 100% recall and 96.9% mAP indicating that the improved model is better than other models. To achieve industry standards in defect detection, the model strikes a reasonable balance between detection accuracy and speed. The newly released YOLOv8 is also compared with the model, showing that the YOLOv7-CSF leads in detection on the XSSD dataset. Furthermore, the model performance is evaluated with benchmark steel strip datasets namely, NEU-DET and GC10-DET. This model can be used in many future industrial and agricultural small-target detection scenarios. The head portion of the network is where YOLOv7 model improvement mostly takes place—adding a coordinate attention mechanism. The model fusion characteristics also depend on the backbone structure. Thus, in the future, our research plan will be extended to better understand the network model and pay more attention to the backbone. Grayscale images of the dataset are used in this study. Therefore, our investigation will further look into the use of hyper-spectral images, real time industrial images and evaluate the power of the model. Also, the images diversity and quality can be further improved by employing Generative Adversarial Network, which in turn, enhance the model’s defect detection accuracy.

REFERENCES

[1] T. Hoese, F. Bachofer, and C. Kuenzer, “Object detection and image segmentation with deep learning on Earth observation data: A review—Part II: Applications,” *Remote Sens.*, vol. 12, no. 18, p. 3053, Sep. 2020.

[2] S. Yu, J. Xiao, B. Zhang, E. G. Lim, and Y. Zhao, “Fast pixel-matching for video object segmentation,” *Signal Process., Image Commun.*, vol. 98, Oct. 2021, Art. no. 116373.

- [3] C. Sun, C. Li, J. Zhang, F. Kulwa, and X. Li, "Hierarchical conditional random field model for multi-object segmentation in gastric histopathology images," *Electron. Lett.*, vol. 56, no. 15, pp. 750–753, Jul. 2020.
- [4] R. Yang and Y. Yu, "Artificial convolutional neural network in object detection and semantic segmentation for medical imaging analysis," *Frontiers Oncol.*, vol. 11, Mar. 2021, Art. no. 638182.
- [5] O. Wosner, G. Farjon, and A. Bar-Hillel, "Object detection in agricultural contexts: A multiple resolution benchmark and comparison to human," *Comput. Electron. Agricult.*, vol. 189, Oct. 2021, Art. no. 106404.
- [6] N. Gengeç, O. Eker, H. Çevikalp, A. Yazıcı, and H. S. Yavuz, "Visual object detection for autonomous transport vehicles in smart factories," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 29, no. 4, pp. 2101–2115, Jul. 2021.
- [7] Z. Wei, S. Dong, and X. Wang, "Petrochemical equipment detection by improved YOLOv5 with multiscale deep feature fusion and attention mechanism," *J. Electr. Comput. Eng.*, vol. 2022, pp. 1–13, Dec. 2022.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [9] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [14] M. Hussain, H. Al-Aqrabi, M. Munawar, R. Hill, and T. Alsoufi, "Domain feature mapping with YOLOv7 for automated edge-based pallet racking inspections," *Sensors*, vol. 22, no. 18, p. 6927, Sep. 2022.
- [15] X. Feng, X. Gao, and L. Luo, "X-SDD: A new benchmark for hot rolled steel strip surface defects detection," *Symmetry*, vol. 13, no. 4, p. 706, Apr. 2021.
- [16] A. Kumar and A. K. Das, "Evolution of microstructure and mechanical properties of Co-SiC tungsten inert gas clad coating on 304 stainless steel," *Eng. Sci. Technol., Int. J.*, vol. 24, no. 3, pp. 591–604, Jun. 2021.
- [17] S. Karthikeyan, M. C. Pravin, B. Sathyabama, and M. Mareeswari, "DWT based LCP features for the classification of steel surface defects in SEM images with KNN classifier," *Austral. J. Basic Appl. Sci.*, vol. 10, no. 5, p. 7, 2016.
- [18] R. Zaghdoudi, H. Seridi, and A. Boudiaf, "Multiple classifier combination for steel surface inspection," in *Proc. Conf. 2nd Conf. Inform. Appl. Math.*, 2019, pp. 1–6.
- [19] F.-M. Schleich and P. Tino, "Indefinite core vector machine," *Pattern Recognit.*, vol. 71, pp. 187–195, Nov. 2017.
- [20] R. Gong, C. Wu, and M. Chu, "Steel surface defect classification using multiple hyper-spheres support vector machine with additional information," *Chemometric Intell. Lab. Syst.*, vol. 172, pp. 109–117, Jan. 2018.
- [21] Y. Liu, K. Xu, and J. Xu, "An improved MB-LBP defect recognition approach for the surface of steel plates," *Appl. Sci.*, vol. 9, no. 20, p. 4222, Oct. 2019.
- [22] L. Wu, X. Lin, Z. Chen, P. Lin, and S. Cheng, "Surface crack detection based on image stitching and transfer learning with pretrained convolutional neural network," *Struct. Control Health Monitor.*, vol. 28, no. 8, Aug. 2021, Art. no. e2766.
- [23] Q. Luo, X. Fang, Y. Sun, L. Liu, J. Ai, C. Yang, and O. Simpson, "Surface defect classification for hot-rolled steel strips by selectively dominant local binary patterns," *IEEE Access*, vol. 7, pp. 23488–23499, 2019.
- [24] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic sign detection via improved sparse R-CNN for autonomous vehicles," *J. Adv. Transp.*, vol. 2022, pp. 1–16, Mar. 2022.
- [25] T. Liang, H. Bao, W. Pan, X. Fan, and H. Li, "DetectFormer: Category-assisted transformer for traffic scene object detection," *Sensors*, vol. 22, no. 13, p. 4833, Jun. 2022.
- [26] T. Liang, H. Bao, W. Pan, and F. Pan, "ALODAD: An anchor-free lightweight object detector for autonomous driving," *IEEE Access*, vol. 10, pp. 40701–40714, 2022.
- [27] Y. Li, H. Huang, Q. Xie, L. Yao, and Q. Chen, "Research on a surface defect detection algorithm based on MobileNet-SSD," *Appl. Sci.*, vol. 8, no. 9, p. 1678, Sep. 2018.
- [28] X. Zheng, S. Zheng, Y. Kong, and J. Chen, "Recent advances in surface defect inspection of industrial products using deep learning techniques," *Int. J. Adv. Manuf. Technol.*, vol. 113, nos. 1–2, pp. 35–58, Mar. 2021.
- [29] V. Nasir and F. Sassani, "A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges," *Int. J. Adv. Manuf. Technol.*, vol. 115, nos. 9–10, pp. 2683–2709, Aug. 2021.
- [30] X. Liu, L. Wu, X. Guo, D. Andriukaitis, G. Królczyk, and Z. Li, "A novel approach for surface defect detection of lithium battery based on improved K-nearest neighbor and Euclidean clustering segmentation," *Int. J. Adv. Manuf. Technol.*, vol. 127, nos. 1–2, pp. 971–985, Jul. 2023.
- [31] M. Li, H. Wang, and Z. Wan, "Surface defect detection of steel strips based on improved YOLOv4," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108208.
- [32] H. Deng, J. Cheng, T. Liu, B. Cheng, and Z. Sun, "Research on iron surface crack detection algorithm based on improved YOLOv4 network," *J. Phys., Conf. Ser.*, vol. 1631, no. 1, Sep. 2020, Art. no. 01208.
- [33] S. Wang, H. Wang, F. Yang, F. Liu, and L. Zeng, "Attention-based deep learning for chip-surface-defect detection," *Int. J. Adv. Manuf. Technol.*, vol. 121, nos. 3–4, pp. 1957–1971, Jul. 2022.
- [34] S. Li and X. Wang, "YOLOv5-based defect detection model for hot rolled strip steel," *J. Phys., Conf. Ser.*, vol. 2171, no. 1, Jan. 2022, Art. no. 012040.
- [35] J.-T. Huang and C.-H. Ting, "Deep learning object detection applied to defect recognition of memory modules," *Int. J. Adv. Manuf. Technol.*, vol. 121, nos. 11–12, pp. 8433–8445, Aug. 2022.
- [36] J. Shi, J. Yang, and Y. Zhang, "Research on steel surface defect detection based on YOLOv5 with attention mechanism," *Electronics*, vol. 11, no. 22, p. 3735, Nov. 2022.
- [37] Y. Wang, H. Wang, and Z. Xin, "Efficient detection model of steel strip surface defects based on YOLO-V7," *IEEE Access*, vol. 10, pp. 133936–133944, 2022.
- [38] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.
- [39] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13728–13737.
- [40] X. Ding, T. Hao, J. Tan, J. Liu, J. Han, Y. Guo, and G. Ding, "ResRep: Lossless CNN pruning via decoupling remembering and forgetting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4490–4500.
- [41] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
- [42] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.
- [43] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A²-nets: Double attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 350–359.
- [44] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10093–10102.
- [45] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3019–3028.
- [46] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.
- [47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

- [48] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "OCNet: Object context network for scene parsing," 2018, *arXiv:1809.00916*.
- [49] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K., Aug. 2020, pp. 213–229.
- [50] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [51] Z. Qin, P. Zhang, F. Wu, and X. Li, "FcaNet: Frequency channel attention networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 763–772.
- [52] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13713–13722.
- [53] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 11531–11539.
- [54] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Conf. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 3–19.
- [55] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3141–3149.
- [56] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 11863–11874.
- [57] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
- [58] L. H. Reeker *Performance Metrics for Intelligent System*. Accessed: Dec. 2007. [Online]. Available: http://www.isd.me1.nist.gov/research_areas/research_engineering/PerMIS_Workshop/22
- [59] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-Score, with implication for evaluation," in *Proc. Eur. Conf. Inf. Retr.* Berlin, Germany: Springer, Mar. 2005, pp. 345–359.



G. DEEPTI RAJ received the bachelor's degree from KSRMCE, in 2009, and the master's degree from GPREC, Andhra Pradesh, in 2011. She is currently a Research Scholar with the School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, Tamil Nadu, India. She has over three years of teaching experience. She has published review articles in international conferences. Her research interests include artificial intelligence, computer vision, the Internet of Things, and human–computer interaction.



B. PRABADEVI received the bachelor's and master's degrees from Anna University, Chennai, in 2010 and 2012, respectively, and the Ph.D. degree in information technology with networking as a specialization from the Vellore Institute Technology, Vellore, Tamil Nadu, India, in 2018. She is currently an Associate Professor with the School of Computer Science Engineering and Information Systems, Vellore Institute of Technology. She is having ten years of teaching experience. She has published around 30 research articles in journals of international repute and some in high-impact factor journals. In addition, she has around 20 international conference papers, five book chapters, and an edited book. Her research interests include decision support systems, artificial intelligence, blockchain, and edge computing. She received the Active Researcher Award from the Vellore Institute of Technology for six consecutive years. She has applied for three Indian patents and three funded projects in trans-disciplinary research.

• • •