https://doi.org/10.1007/s42979-023-02436-2

ORIGINAL RESEARCH

A Review and Benchmark on State-of-the-Art Steel Defects Detection

Anthony Ashwin Peter Chazhoor[1] · Edmond S. L. Ho[3] · Bin Gao[2] · Wai Lok Woo[1]

Received: 4 August 2022 / Accepted: 18 October 2023

Abstract

Steel, a critical material in construction, automobile, and railroad manufacturing industries, often presents defects that can lead to equipment failure, significant safety risks, and costly downtime. This research aims to evaluate the performance of state-of-the-art object detection models in detecting defects on steel surfaces, a critical task in industries such as railroad and automobile manufacturing. The study addresses the challenges of limited defect data and lengthy model training times. Five existing state-of-the-art object detection models (faster R-CNN, deformable DETR, double head R-CNN, Retinanet, and deformable convolutional network) were benchmarked on the Northeastern University (NEU) steel dataset. The selection of models covers a broad spectrum of methodologies, including two-stage detectors, single-stage detectors, transformers, and a model incorporating deformable convolutions. The deformable convolutional network achieved the highest accuracy of 77.28% on the NEU dataset following a fivefold cross-validation method. Other models also demonstrated notable performance, with accuracies within the 70–75% range. Certain models exhibited particular strengths in detecting specific defects, indicating potential areas for future research and model improvement. The findings provide a comprehensive foundation for future research in steel defect detection and have significant implications for practical applications. The research could improve quality control processes in the steel industry by automating the defect detection task, leading to safer and more reliable steel products and protecting workers by removing the human factor from hazardous environments.

Keywords Steel defect detection · Automated defect detection · Steel surface · Deep learning models

Introduction

Steel is crucial to our way of life and essential to a sustainable circular economy, since it is a permanent material that can be recycled indefinitely without losing its quality. Global crude steel output in 2019 was 1868.8 million tonnes, according to the World Steel Association [1], making it incredibly significant and lucrative in today's building, industrial, and construction industries and are utilized in many sectors, including automobile production, aviation components, tools, agricultural, mining, catering, shipping, medical, electronic parts, and construction, to mention a few. One of the most essential factors for determining the quality of hot-rolled steel sheets is their appearance. If the product does not meet the standards and criteria of clients due to defects, the end user will not accept it. There is a considerable motivation to detect surface defects early in the fabrication process, because undesirable or irreparable faults may result in the rejection of a completed fabrication product [2]. Detecting surface defects has become increasingly important as customer requirements have become more stringent [3].

✉ Anthony Ashwin Peter Chazhoor
anthony.chazhoor@northumbria.ac.uk

Edmond S. L. Ho
shu-lim.ho@glasgow.ac.uk

Bin Gao
bin_gao@uestc.edu.cn

Wai Lok Woo
wailok.woo@northumbria.ac.uk

[1] Computer and Information Science Department, Northumbria University, ELB 108, Ellison Pl, Newcastle upon Tyne NE1 8ST, UK

[2] School of Automation Engineering, University of Electronic Science and Technology of China, 4 1st Ring Rd East 2 Section, Chenghua District, Chengdu 610056, Sichuan, China

[3] School of Computing Science, University of Glasgow, 18 Lilybank Gardens, Glasgow G12 8RZ, UK

Published online: 18 December 2023

The typical surface defects of steel, as depicted in Fig. 1 from the Northeastern University (NEU) steel dataset, includes crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches [4]. The intraclass defects in the dataset result in considerable differences in appearance, such as the category scratches exhibiting horizontal, vertical, or diagonal scratch defects. Meanwhile, interclass defects such as rolled-in scale, crazing, and pitted surfaces have similar properties. Due to the lighting and material variances in the greyscale images, defects having interclass similarities are incredibly challenging to detect. The evolution in computer vision and image processing techniques has created novel ways, which include object detection with SWIFT clustering [5], SURF modelling [6], among others, to detect the defects [7]. However, they take a comparatively longer time and lower accuracy than modern deep learning techniques [8]. A wide range of cutting-edge deep learning models that can detect objects and defects in images with high accuracy has been researched and implemented in the industry [9]. The object detection algorithm consists of single- and two-stage detectors. The two-stage detector includes a convolutional neural network (CNN) that extracts image features for classification, region proposal networks, and bounding box regression at the end. In contrast, the single-stage detector does not include region proposal networks.

An efficient neural network model that can detect defects accurately must be created to detect the defects on the steel surfaces. A massive quantity of labelled data must be trained using a powerful graphical processing unit (GPU) to obtain optimum accuracy in detecting defects on steel surfaces. The introduction of transfer learning has resulted in a significant shift in the learning processes of deep neural networks. Since the dataset used in this study is relatively small, with a limited number of images in each class, transfer learning is integrated before training each model. While previous studies have made significant strides in steel surface defect detection, they have primarily focused on defect classification rather than localization and detection. The benchmark paper by the authors of [10] introduced a novel Xsteel surface defect dataset (X-SDD) and analyzed the performance of various models. However, their work was limited to defect classification and did not extend to the localization and detection of these defects. Similarly, the authors of [11] used transfer learning to analyse state-of-the-art classifiers on the NEU and Severstal datasets. However, their work was also confined to classifying the defects.

Therefore, there is a clear gap in the literature for research that not only classifies steel surface defects, but also localizes and detects them. This is a crucial area of research, as the ability to accurately localize and detect defects can significantly improve the quality control processes in the steel industry. This research investigates the existing algorithms for surface steel detection on the Northeastern University (NEU) steel database [12], which includes faster R-CNN [13], deformable DETR [14], double head R-CNN (DH R-CNN) [15], Retinanet [16] and deformable convolution network (DCN) [17]. The NEU steel database is a comprehensive and challenging database that contains images of

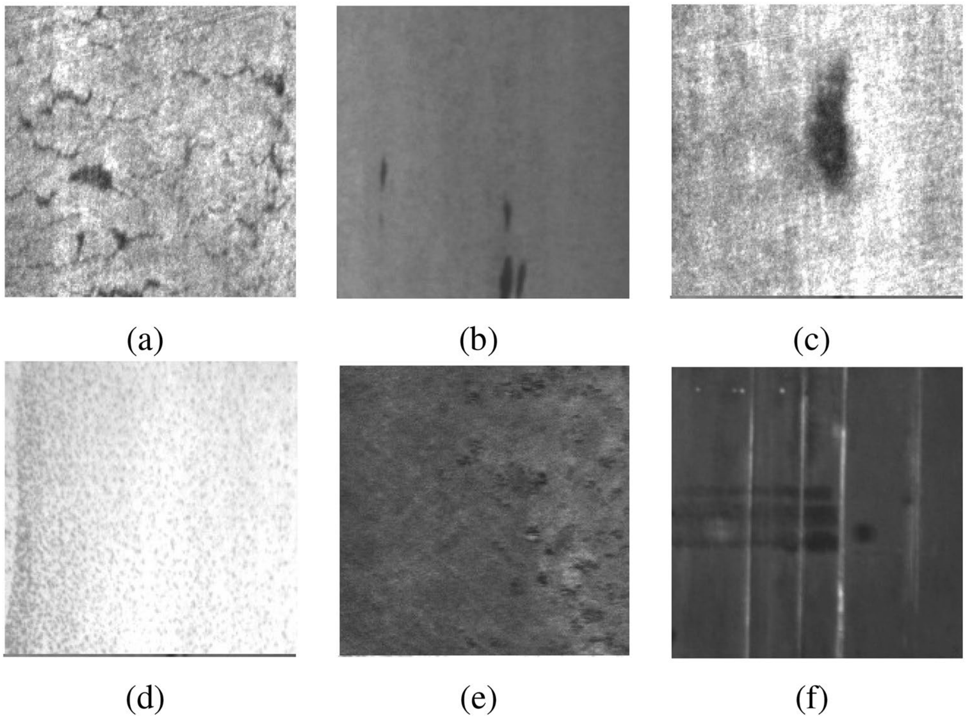

Fig. 1 Sample images from across all classes from the NEU steel dataset. **a** Crazing. **b** Inclusion. **c** Patches. **d** Rolled-in scale. **e** Pitted surface. **f** Scratches

hot-rolled steel strips with multiple defects, some of which are invisible to the naked eye. Experimentation revealed that the DCN model works best, with 77.28% average precision, followed by the DH R-CNN. The benchmarking work will offer an unbiased view of the existing deep learning models applied to the NEU steel dataset serving as a basis for future research. This effort will serve as a tool for evaluating and comparing performance to create continuous improvement to the future models leading to a more efficient model. ”Related work”, “Background” and “Methodology” consists of the related work, dataset and methodology, respectively, followed by “Results and discussions”, “Analysis” and “Conclusion”, consisting of results, analysis and conclusion.

Related Work

The literature of vision-based approaches on defect detection is divided into two categories: traditional computer vision and deep learning approaches.

Traditional Techniques

The traditional image processing approach detects and segments defects using rudimentary properties indicated by local abnormalities. Jonker et al. [18] proposed a pattern recognition classifier based on support vector classifiers to distinguish the metallic defects. Jia et al. [19] proposed a method that thoroughly investigated the visual defect features and created a support vector machine learning algorithm that automatically learns complex decision limits in the presence of data noise. Pernkopf et al. [20] proposed a surface defect detection on steel blocks using the Bayesian network classifiers. Wu et al. [21] proposed an approach that can eliminate the problem of false alarms by scales and water markings using an undecimated wavelet transform and mathematical morphology in the surface defect detection of hot-rolled steel plates. Yazdchi et al. [22] eliminated the stationary background and improved the image detection by employing the temporal Fourier analysis. Borselli et al. [23] used the fuzzy inference method to figure out the flaws to be solved in a classification challenge. Li et al. [24] proposed a plan named regular bands which requires knowledge of the period length of a repeating pattern to detect defects with X-rays. Liu et al. [25] devised a defect identification technique based on the Haar–Weibull variance, created to describe stochastic texture distributions.

While these traditional techniques have achieved detection accuracies in the range of 90%, they have several limitations [26]. They often rely heavily on manual feature engineering, which requires domain knowledge and can be time-consuming. They might perform well on specific tasks or datasets they were designed for, but they may need to generalize better to new jobs or different data types [27]. Traditional methods might also be sensitive to variations in lighting, orientation, scale, etc. [28]. Deep learning methods have significantly outperformed conventional methods for complex tasks involving a high-level understanding of the scene, such as object detection, semantic segmentation, or image captioning. Furthermore, the performance of deep learning models typically improves with more data, making them a better choice in the era of big data [29]. Traditional techniques might benefit less from large datasets. Lastly, deep learning allows for end-to-end learning, where a model learns to map raw input data to output predictions in a single step. This can lead to better performance by enabling the model to learn complex patterns in the data. In contrast, traditional methods often involve multiple steps (e.g., feature extraction, feature selection, classification), which might not be optimally combined. Despite these limitations, traditional computer vision techniques have significantly progressed in defect detection. Still, the era of deep learning has surpassed these traditional techniques exponentially in a relatively shorter time and with higher accuracy [29].

Deep Learning Techniques

Since the early 2010, research into applying deep learning to tackle computer vision challenges has accelerated. Examples include object identification, object tracking, picture classification, semantic segmentation, feature extraction, and other vision challenges. Object detection is the critical interest in defect detection since defects are viewed as objects and must be located and categorized [30]. Deep learning has reduced the need for human interaction in categorization, making it more accessible, efficient, and cost-effective, with the CNNs supplementing the deep learning technique. CNNs are suitable for object classification, localization, and detection. CNNs, however need a massive amount of training data, which might be challenging due to ethical and social causes. Artificial neural networks have been frequently employed for defect identification using CNN because they have consistently demonstrated high performance in processing images to get the desired output [31].

He et al. [32] devised a new deep learning-based defect detection system with realistic commercial applicability by fusing multiple hierarchical features that can determine the class along with the detailed location of the defect. Lv et al. [33] contributed a new dataset GC10-DET and benchmarked state-of-the-art deep learning models including (single-stage detector) SSD, faster R-CNN, and YOLO (You Only Look Once) algorithms on the NEU-DET and GC10-DET dataset along with proposing a novel method to address the accuracy requirements for metallic defect detection. Cheng et al. [34] proposed an improvised single-stage detector called the DEA_retinanet having variable channel attention and remarkable feature fusion. Tang et al. [35] proposed a method for detecting surface defects on steel strips based on an attention mechanism and multi-scale max pooling. To compensate for the information loss caused by single max pooling, the study suggested an attention mechanism and a multi-scale max-pooling (MSMP) approach. Li et al. [36] proposed an improved YOLO_v5 and an optimized Inception Resnet_V2 model in a Two-Stage Industrial Defect Detection Framework to address the problem of complicated morphology, tiny size, and other comparable characteristics. Tian et al. [4] proposed DCC-CenterNet which locates centre points using a keypoint estimation technique using a centeredness function improving the overall accuracy.

Much time and research are devoted to detecting defects in steel surfaces with the help of modern and innovative deep learning algorithms. The deep learning models have a superior advantage over the traditional vision-based approaches as they attain a higher accuracy within a short training period. Numerous research papers published on steel defect detection claim to localize and detect defects, but many focus on classification [37–39].

Background

The background section discusses the Northeastern University (NEU) dataset, its composition, and the addition of a 'background' class for improved model performance. An outline of cross-validation and model training methodology, including dataset division, learning rate selection, and epoch counts for each model is then explained.

A crucial part of our methodology is the tuning of hyperparameters. Hyperparameters are parameters whose values are set before the learning process begins and can significantly influence the performance of a model. Our study used a consistent set of hyperparameters across all models to provide a fair and consistent comparison ground. However, we acknowledge that this approach might not bring out the best performance of each model. Therefore, we discuss our choice of hyperparameters and the potential impact of hyperparameter tuning on model performance. Therefore, we discuss our choice of hyperparameters and the potential impact of hyperparameter tuning on model performance in the 'Cross-Validation and Model Training Tuning' section of this paper.

An overview of the key concepts behind the models used in this study, namely faster R-CNN, DH R-CNN, RetinaNet, DETR, and DCN, is provided, along with a discussion on their relevance in deep learning. Lastly, we touch upon the concept and benefits of transfer learning and its application in our study. This section offers the necessary context to understand our research approach and findings.

Dataset

The Northeastern University (NEU) surface defect database record [12] contains 1800 greyscale images divided into six classes with 300 samples each, as displayed in Table 1. The six classes of hot-rolled steel strip surface defects, comprising crazing (Cr), inclusion (In), patches (Pa), pitted surface (PS), rolled-in scale (RS), and scratches (Sc), are illustrated in Fig. 1 [40].

In addition to these six classes, we have introduced a ’background’ class. This class represents areas in an image that do not contain any of the object classes the model has been trained to detect. Including a background class is crucial for the model to classify and avoid false positives correctly. The background class serves as a category for anything the model does not recognize as an object of interest. This strategy helps reduce the number of false positives, thereby improving the model's overall accuracy [41].

All the images in the dataset have a resolution of 200 × 200 pixels. The defect crazing, pitted surface, and rolled-in scale have interclass similarities making their detection strenuous. Due to the intraclass similarities like the horizontal, slanted, and vertical patterns, defects such as scratches are quite challenging to detect. The greyscale dataset can result in faulty detection of intraclass defects due to the lighting conditions. The intraclass as well as the interclass defects are the significant challenges in the NEU steel dataset [40]. Intraclass defects are variations or differences within the same type of defect, while interclass defects occur between different types of defects.

Cross-Validation and Model Training

The dataset was then divided into fivefold for cross-validation. Cross-validation is a statistical method used to estimate the skill of machine learning models. It involves partitioning the dataset into subsets, training the model on some subsets, and testing the model on the remaining subsets. This process is repeated multiple times, and the results are averaged to provide a more robust estimate of model performance [42].

Table 1 Number of images in each class of the NEU dataset

Defect	Number of images
Crazing	300
Inclusion	300
Patches	300
Pitted surface	300
Rolled-in scale	300
Scratches	300
Total	1800

The entire dataset was tested to generate the mean average precision (mAP) for each of the models. Faster R-CNN, DH R-CNN, Retinanet, and DCN models were put through the training loop of 24 epochs, whereas the deformable DETR model had to run 50 epochs for the data to converge.

The learning rate, a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving towards a minimum of a loss function, was carefully adjusted during training [43]. The learning rate of 0.0025 was chosen for all models to ensure a fair comparison. This value is a reasonable choice because it is small enough to prevent the models from overshooting the optimal solution, but large enough to ensure that the models converge in a reasonable amount of time.

Models and Their Key Concepts

Faster R-CNN is the oldest model used in this study and many new models have been developed based on it. It remains the state-of-the-art two-stage used in a variety of applications. Retinanet is the only single-stage state-of-the-art detector compared in this study. Having at par accuracy with two-stage detectors and concise training and inference time, Retinanet is used in diverse applications. DH R-CNN is a model that challenges the conventional two-stage models by exchanging its heads for classification and localization tasks and gives interesting results on the steel dataset. Deformable DETR is a latest detector and is a faster version of DETR [44] which uses transformers for object detection. All the above models are the most used in the field of deep learning, forming a benchmark for the NEU steel dataset.

Transfer Learning

A large amount of data along with a considerably large training time using a strong graphical processing unit (GPU) is needed to achieve optimal output accuracy in classification and detection tasks after training a neural network. The learning mechanisms in deep neural networks have changed exponentially with the introduction of transfer learning. A pre-trained model is a deep learning model trained on a large benchmark dataset like the ImageNet and it excels in extracting image characteristics from a new dataset [45]. Transfer learning aids in the training of new data using previously learned data. It also aids in efficiently avoiding data overfitting [46]. For all the experiments, fivefold cross-validation was used to validate the benchmark models. The training loss and accuracy, validation loss and accuracy, and training duration were all collected with identical model parameters. The resulting average data was tallied, and graphs for visual representation were created.

Methodology

In this research, we adopted a systematic approach to ensure the accuracy and reliability of our results. Our methodology involved several crucial steps, each designed to address specific aspects of the research question. We began by sourcing our dataset from the NEU website, which provided a diverse range of images for our analysis. Recognizing the potential issues that could arise from poor image quality, imbalanced distribution, and mislabelling, we undertook a meticulous examination of the dataset. We manually verified the ground truth annotations on the images, ensuring that our dataset was robust and reliable. This step was critical as a dataset health check is a prerequisite for training a defect detection model. Following the initial preparation, we divided the dataset into fivefold for cross-validation. This process was designed to provide a comprehensive assessment of our models' performance across different subsets of the data. We then tested the entire dataset to generate the mean average precision (mAP) for each of the models. Our study employed several models, including faster R-CNN, DH R-CNN, Retinanet, and DCN. These models underwent a training loop of 24 epochs. However, the deformable DETR model required 50 epochs for the data to converge. In all our models, we used Resnet-50 as the backbone. This decision was based on Resnet-50's proven performance in various tasks. The subsequent sections provide a more detailed discussion of the backbone and the models used in this research.

Backbone

In our study, we have chosen to use ResNet-50 as the backbone architecture. This choice is motivated by several factors. ResNet-50, a convolutional neural network with skip connections and 11 million parameters, has been widely adopted in the field of computer vision due to its deep, yet computationally efficient architecture. This allows it to learn a rich hierarchy of features from input images, making it a robust choice for our application [47]. Furthermore, ResNet-50 has demonstrated superior performance in various tasks, including image classification, object detection, and semantic segmentation. Its innovative use of residual connections effectively mitigates the vanishing gradient problem, enabling it to learn effectively from large-scale datasets. Lastly, our choice of ResNet-50 aligns with established practices in the field, facilitating comparisons with other studies and promoting the reproducibility of our work [48].

Resnet-50

Residual networks (Resnet-50) are extremely deep convolutional neural networks possessing skip connections with 11 million parameters. By linking each block with a skip connection, the vanishing gradient problem is addressed. The skip connection skips over several network levels. The two 3 × 3 convolutions are used in each block to get the desired output utilizing batch normalization and rectified linear unit (ReLU) activation [49]. The architecture of Resnet-50 is displayed in Fig. 2.

Defect Detection Models

In this subsection, we will introduce and discuss five deep learning models that have been utilized in our study. Each of these models plays a crucial role in addressing specific aspects of our research problem. We will provide a concise overview of the model

Faster R-CNN

The region-based convolutional neural network (R-CNN) [13] is a popular two-stage neural network for object recognition. This network comprises three subsidiary networks: the feature network, the region proposal network (RPN), and the detection network [51]. Faster R-CNN replaces the selective search approach with an RPN enhancing the object detection accuracy and speed. The model thus has quite a good advantage over other models to detect smaller objects. The structure of faster R-CNN incorporates feature extraction, region proposal, bounding-box regression, and classification. The faster R-CNN architecture combines fast R-CNN and the RPN which replaces the selective search method increasing the speed and accuracy of detection by sharing the feature map extracted by the convolutional network [30]. With Resnet-50 as the backbone, the pre-trained faster R-CNN uses a feature pyramid network that extracts single-scaled images as inputs and returns proportionately scaled feature maps at several layers in a completely convolutional way. The batch size chosen is two images per GPU and the loss function used is the cross-entropy loss. The model incorporates a threshold value of 0.5 for non-maximum suppression (NMS). The architecture of the faster R-CNN model is displayed in Fig. 3.

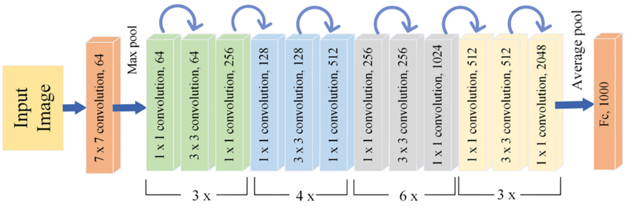

Fig. 2 Architecture of Resnet-50 [50]

Deformable Detection Transformer (DETR)

Carion et al. [52] proposed DETR, giving a new trajectory to the field of computer vision in 2020. An updated version of the DETR was released in 2021 by Zhu et al. [14] called the deformable DETR. It required ten times fewer epochs to train by giving comparatively better accuracies on smaller objects compared to the preceding models. Deformable DETR's attention modules only pay attention to a few essential sample points surrounding a reference point [53]. To sample local pixels rather than all pixels, deformable DETR replaces the multi-head self-attention layer with a deformable attention layer along with a cross-section module that integrates the multi-scale feature representation [54]. The backbone used is Resnet-50 with an encoder and decoder having six layers each. The type of loss function used is the Focal loss which allows practical training on all the samples in the dataset without simple negatives overpowering the loss function. The model includes a threshold value of 0.5 for Non-maximum Suppression (NMS). The optimizer used in the transformer model is ADAMW [55]. The architecture of the deformable DETR model is displayed in Fig. 4.

Double Head R-CNN

The double-head R-CNN (DH R-CNN) [56] was proposed as the model separates the sibling head of the R-CNN network into two distinct branches for classification and localization. DH R-CNN investigates the best architectures for classification and localization using a contrasting approach in comparison to the conventional R-CNN. Wu et al. [15] recognized that for the classification challenge, the fully connected head is more suitable whereas for the localization or bounding box regression task is suitable. The batch size of two images per GPU was chosen, and the loss function utilized was the cross-entropy loss. The model has a

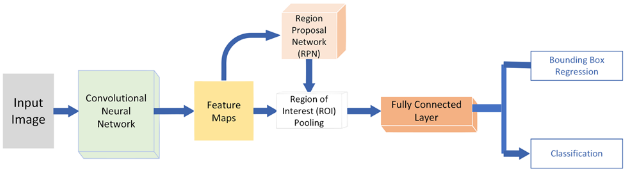
Input Image
Convolutional Neural Network
Feature Maps
Region Proposal Network (RPN)
Region of Interest (ROI) Pooling
Fully Connected Layer
Bounding Box Regression
Classification

Fig. 3 Architecture of faster R-CNN

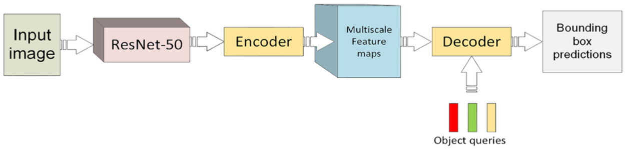

Fig. 4 Architecture of deformable DETR

non-maximum suppression (NMS) threshold of 0.5. The architecture of the deformable DETR model is displayed in Fig. 5.

Retinanet

Retinanet [34] is a single-stage detector made up of a Resnet-50 as the backbone network and two task-specific subnetworks that establish the focal loss in the training process and play an essential role in promoting object detection research in several aspects. Equivalent to the speed of previous single-stage detectors, the Retinanet proposed by Lin et al. [16] surpasses the accuracy of the two-stage detectors at a faster rate. The backbone, an off-the-shelf convolutional network, oversees constructing a convolutional feature map over an entire input picture. On the output of the backbone, the first subnetwork conducts convolutional object classification; the second subnetwork does convolutional bounding box regression. The two subnetworks have a straightforward architecture. The loss used is focal loss with Resnet-50 as the backbone. The architecture of the Retinanet model is displayed in Fig. 6.

Deformable Convolution Network (DCN)

DCNs [17] are used to learn more information and features about the geometric transformation that regular convolutional neural networks find challenging to understand. Deformable convolution utilizes a standard fixed sample grid and incorporates 2D offsets into the normal convolution procedure. To acquire fractional points not established places on the grid, bilinear interpolation is utilized to estimate pixel values, which are then added as offsets to the current sampling positions. Deformable convolution layers, which are more likely to include object-level semantic information, can be used in the final layers of the convolutional network [57]. The architecture of the DCN model is displayed in Fig. 7.

Experimental Setup

All the trials were carried out using the Ubuntu Linux operating system. The models were trained on an Intel i7 processor running at 3.60 GHz with 32 GB of RAM and the graphics processing unit was an Nvidia Quadro P5000. PyTorch was the deep learning framework employed in this study. The learning rate was set to 0.0025 for all the experiments. The value of momentum employed was 0.9, which is extensively used in the machine learning and neural network fields. The dataset was divided with an 80:20 split with 1440 images for training and 360 images for testing. Basic augmentation methods like random flip and normalization have been applied to the images for pre-processing. For better visualization, the final result is supplied together with graphs displaying the number of epochs vs accuracy using the fivefold cross-validation technique. The mean average precision (mAP) is reported with a 50% threshold value.

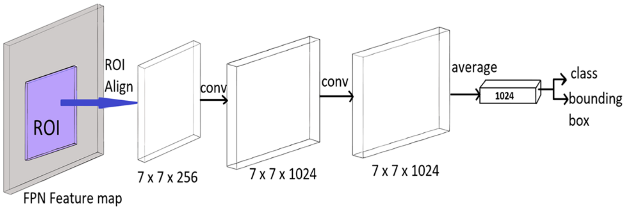

Fig. 5 Architecture of double-head R-CNN

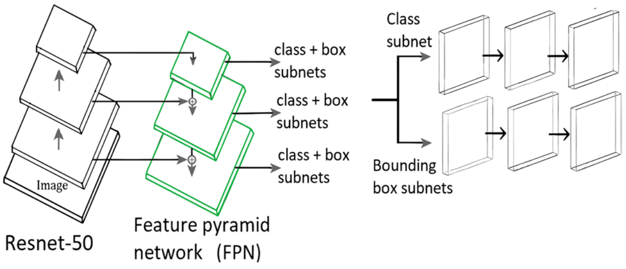

Fig. 6 Architecture of Retinanet

Results and Discussions

Experimental Results

From the experiments, it is observed that the DCN model achieves the highest mAP of 77.28%. The time taken by DCN for inference is lower than thatof other models which is 68.75 ms. Crazing, scratches, and rolled-in scale defects can be detected with a very high accuracy using the DCN model as compared to the other models. Crazing is exceptionally difficult to detect; however, DCN detects

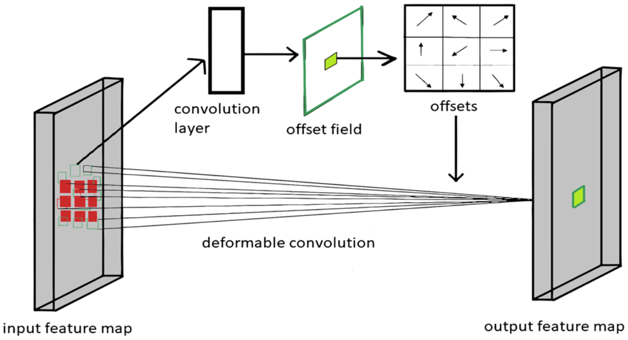

Fig. 7 Architecture of DCN

the defect with an mAP of 48.58%. It can be observed from Fig. 8 that at around the fifth epoch, the accuracy of the DCN model plummets to 60%. Having said that, after the tenth epoch, the DCN model's accuracy begins to flatten and no significant surge is seen. It can be observed from Table 2 that DH R-CNN achieves the second-highest precision with an mAP of 75.56%. The training time taken by the model is around 214 min and for inference it is around 161.1 ms, which is substantially large as compared with many other models used in this research, even after incorporating transfer learning. It attains the highest mAP for defect inclusion with an mAP of 80.48%. The model also attains considerably high accuracy for the defect patches and scratches. In Table 2, values in bold represent the best performance metrics achieved among the compared models. This includes the highest mean Average Precision (mAP) at 0.50, the best performance in detecting various defects such as crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches, as well as the shortest training and inference times, and the fewest number of parameters.

Retinanet takes the lowest inference time of 57.63 ms with an average mAP of 74.56%. Retinanet provides the best accuracy for the defect patches and pitted surface representing an mAP of 92.38% and 85.7% respectively. Retinanet is a single-stage detector and establishes a significant mAP of 74.56% within the least possible training time of 83 min. Faster R-CNN attains an mAP of 73.34% and the training time of the model is 106 min and the inference time is 91.66 ms.

The deformable DETR model takes 302 min of training time to attain an mAP of 71.98%. It takes 50 training epochs for the deformable DETR model to reach the current level of mAP, whereas all the other models in the study took only 24 epochs and considerably less training time to attain the maximum mAP achieved. However, it only took 91.66 ms for inference compared to DH R-CNN, which took 161.1 ms for inference.

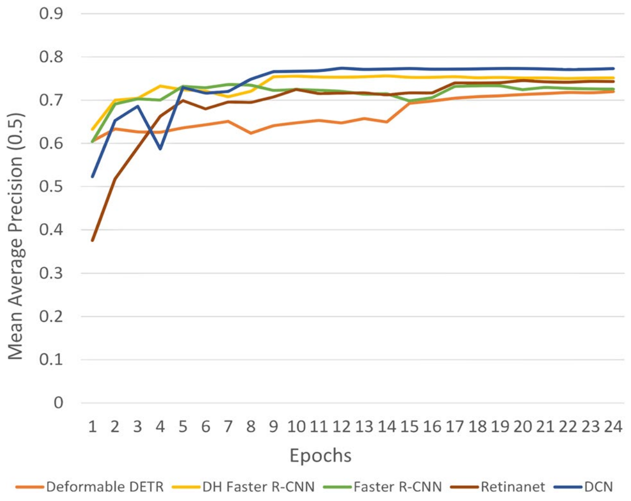

Fig. 8 Mean average precision (0.5) vs number of epochs curve for faster R-CNN, deformable DETR, DH R-CNN, Retinanet and DCN models

Observations

From Fig. 9, positive detection with the blue bounding box representing the ground truth and the orange bounding box representing the predicted outcome during the testing phase were visualized [58]. Similarly, in Fig. 10, the incorrect predictions of the models predicted for the defect can be observed. We can deduce substantial false positives (FP) by the visualization of the testing dataset. Figure 14 shows the graphical representation of class-wise accuracy for each of the tested models. The defects crazing and rolled-in scale have a noticeable drop in accuracy compared with the defects inclusion, patches, pitted surface, and scratches.

Figure 11 shows the confusion matrix [58] for each of the models summarizing the prediction across all six classes can be visualized. For individual classes, significant detection performance can be observed. Comparing the confusion matrices across all the models, it has been observed that the DCN model predicts the defects with better accuracy. Background class has been added to the confusion matrix, which is necessary for tracing FP and missing defections,

Table 2 Mean average precision and the class-wise average precision of the models used in the research along with the training time, inference time, number of parameters, and the total number of epochs

Pretrained	mAP (0.50)	Crazing	Inclusion	Patches	Pitted surface	Rolled-in scale	Scratches	Training time (min)	Inference time (ms)	Parameters (millions)	Epochs
Deformable Detr	71.98	39.52	76.98	89.7	80.74	56.5	77.54	302	91.66	39.82	50
Faster R-CNN	73.3	39.7	76.1	90.62	84.7	56.36	90.42	106	61.80	41.5	24
DH R-CNN	75.56	45.34	**80.48**	92.1	85.48	59.78	91.72	214	161.1	46.74	24
Retinanet	74.56	42.7	76.52	**92.38**	**85.7**	60.52	84.42	**83**	**57.63**	**36.21**	24
DCN	**77.28**	**48.58**	80.24	92.18	85.22	**65.74**	**93**	98	68.75	44.63	24

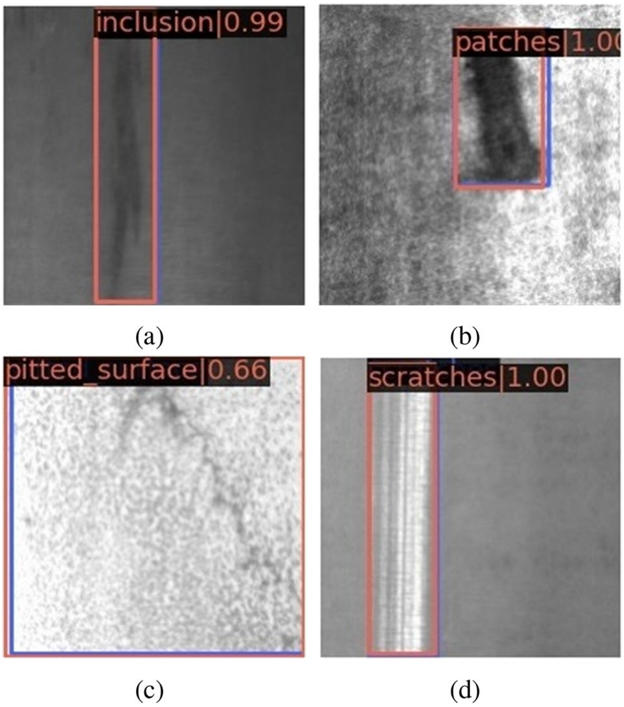

Fig. 9 Samples from the test data which were correctly identified. The predicted bounding boxes are represented in orange, whereas the ground truth is represented by the blue box. **a** Inclusion. **b** Patches. **c** Pitted surface. **d** Scratches

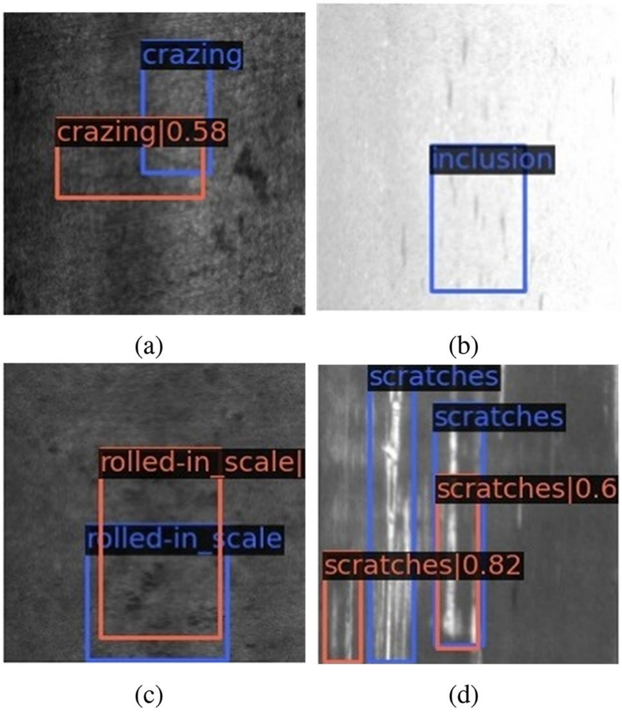

Fig. 10 Samples from the test data which were incorrectly identified. The predicted bounding boxes are represented in orange, whereas the ground truth is represented by the blue box. **a** Crazing. **b** Inclusion. **c** Rolled in scale. **d** Scratches

thereby covering all possible defections. From Fig. 12 [59], we observe the average precision–recall curve for all the models along with the detailed breakdown of FP. Comparing all the models in the study, it is observed that the DCN model has the highest area under the curve (AUC), followed by DH R-CNN. The PR curve gets strictly higher than the last curve as the evaluation criteria become lenient. For deformable DETR, the AP (average precision) at the intersection over union (IoU) at 0.50 is 0.728. The AP grows further to 0.882 after all the localization errors are ignored other than the duplicate detections termed as perfect localization. The AP will rise to 0.887 after all the class confusions are removed which happens when the defects are grouped under the same class label. Finally, once the background FP is removed, the AP further rises to 0.997. The background is a step function that has the value of one until the maximum recall is attained and then declines to zero for a single category. Figure 12 shows that the AP for faster R-CNN is 0.734, and it rises to 0.904 with perfect localization. The elimination of any class confusion does not affect AP. When the background FP is eliminated, the AP rises to 0.974. The AP for DH R-CNN is 0.751, which rises to 0.912 after complete localization and 0.987 after background removal. Retinanet has an AP of 0.74, which rises to 0.89 following perfect localization. After removing the class confusion, the AP would have a value of 0.896. After removing the background, the AP value will be close to 0.997. In the instance of the DCN model, overall AP is 0.771 at IoU = 0.50, and perfect localization would increase AP to 0.912. Removing all class confusion would just marginally increase AP to 0.913. By removing the background FP, the performance improves to 0.982. Figure 13 shows the precision–recall (PR) curve for all the individual models.

Analysis

From all of the models utilized in this study, crazing is the least detectable defect, and it covers medium to extensive defect areas, and a substantial number of images in the training set exhibit more than one crazing defect in a single image. Figure 14 shows that DCN and DH R-CNN have more excellent mAP for defect crazing concerning other models. Both these models are based on R-CNN with a Resnet-50 backbone. The effect of compounded deformation is significant when the deformable convolutions are superimposed. This led to better detection of crazing defects. Retinanet employs thousands of region proposals with focal loss as the loss function, making it challenging to identify crazing, since crazing resembles its background, resulting in multiple false detections reducing the mAP of the entire model for the defect. After visualizing the test results, it is observed that the deformable DETR fails to

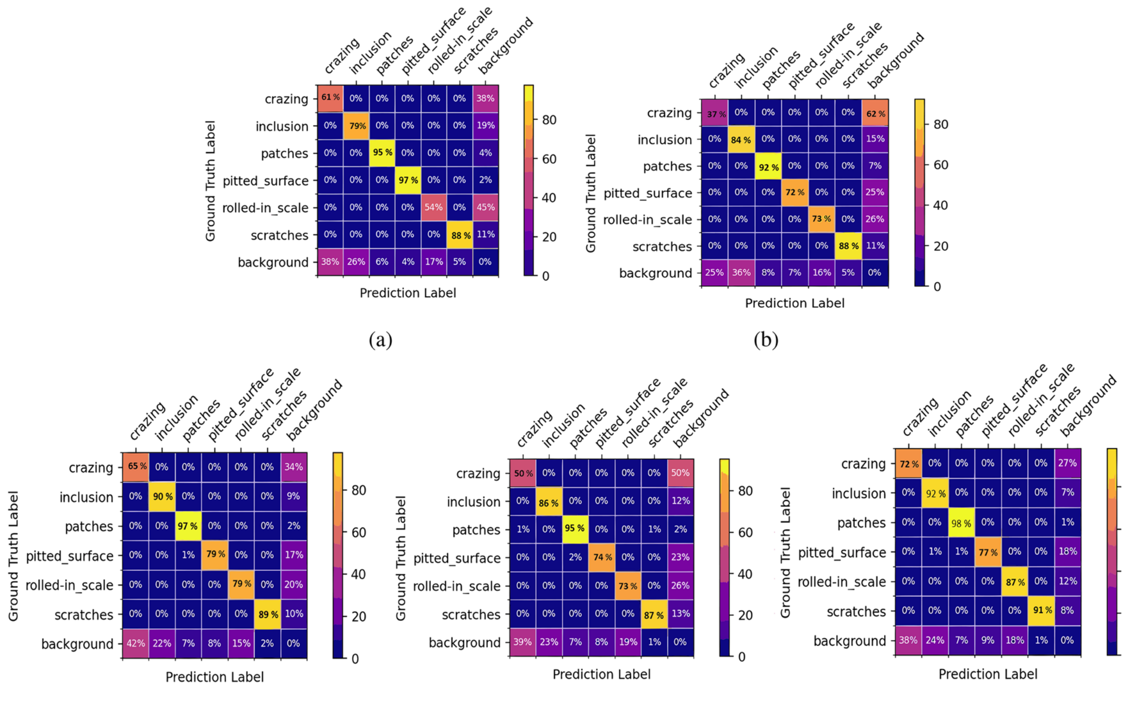

Fig. 11 Confusion matrix. **a** Faster R-CNN. **b** Deformable DETR. **c** DH R-CNN. **d** Retinanet. **e** DCN

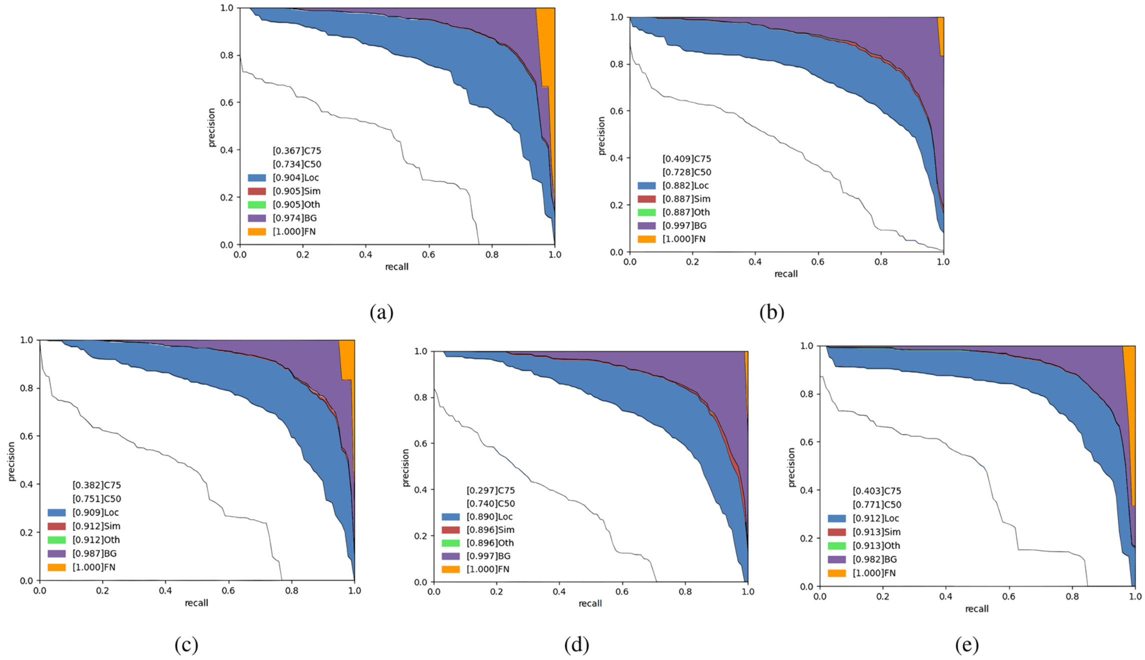

Fig. 12 Precision vs recall curve [59]. **a** Faster R-CNN. **b** Deformable DETR. **c** DH R-CNN. **d** Retinanet. **e** DCN. 1. C75: precision–recall (PR) at IoU = 0.75. 2. C50: PR at IoU = 0.50. 3. Loc: PR at IoU = 0.10. 4. Sim: PR after removal of false positives (FP). 5. Other: PR after removal of class confusions. 6. BG: PR after background removal. 7. PR after removal of all errors

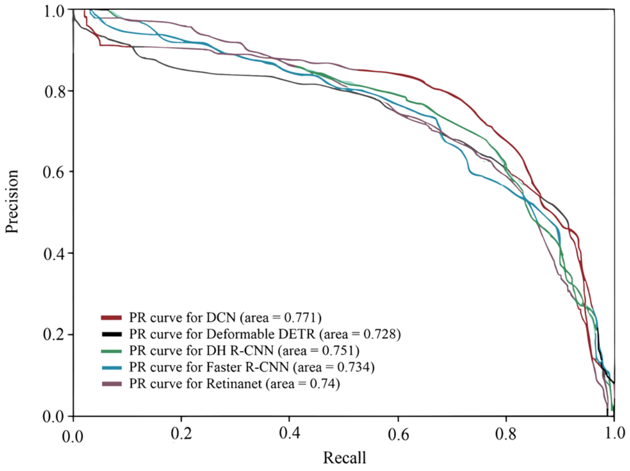

Fig. 13 Precision vs recall curve for faster R-CNN, deformable DETR, DH R-CNN, Retinanet, and DCN models

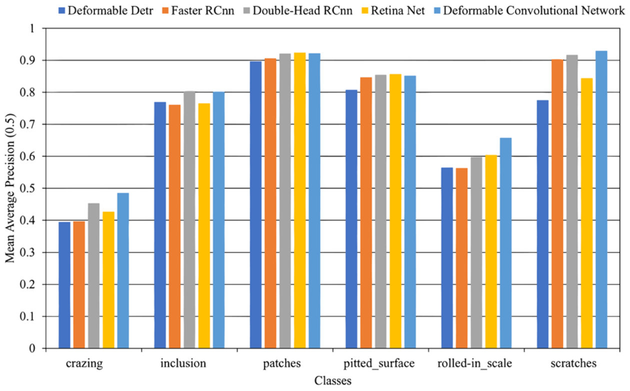

Fig. 14 Classwise mAP for the six defects on faster R-CNN, deformable DETR, DH R-CNN, Retinanet, and DCN models

detect the defect crazing in multiple images. Deformable DETR uses fewer significant reference points in its spatial feature maps, reducing defect identification rates. The defect inclusion is extensive, and some of the samples in the dataset have a distinct appearance, while others blend in with the background. There are multiple inclusion defects in a single image, the highest being six defects in a single image. DCN and DH R-CNN can detect the defect with higher precision than other models having better detection success rates as seen from Fig. 14. The defect patches are medium to large in size having a strong contrast, making them immediately distinguishable from their background. Inclusion defects are contained with patches in some of the images in the dataset, which helps the models in learning the difference between the defect patches and inclusion. The defect patches have high accuracy due to their immediately recognisable appearance. Pitted surface defects are significant, and most of the defects cover the entire dimension of the images, leading to a lower generalization of the defects. DCN, DH R-CNN, Retinanet, and faster R-CNN attain similar accuracies, whereas deformable DETR fails to achieve the average accuracy other models achieve. This could be due to the defect characteristics surrounding the reference point extracted using the multi-scale deformable attention module. The initial estimation of the bounding box centre is made using the reference point predicted by the detecting head as relative offsets from the reference point, resulting in improper defect detection.

Besides crazing, another complex defect to identify is the rolled-in scale, which includes defects of all sizes. The rolled-in scale defect has little resemblance to the defect inclusion and is blended in with its background, making detection challenging. Compared to other models, DCN achieves a high level of precision by adding new offsets to the spatial sampling sites in the deformable convolutional modules and learning the balances during the training. The highest precision is achieved for scratches from all the defects by the DCN model. DH R-CNN and faster R-CNN model also attain perfect accuracy. However, Retinanet and deformable DETR underperforms while detecting scratches. Unlike crazing and rolled-in scale, scratches have a strong contrast to their background and are immediately visible, yet they contain many patterns and defects scattered over the images. The focal loss, which helps distinguish the experience from the object, is the loss function of Retinenet and deformable DETR. Although the defect scratches are more minor, the bounding boxes of the slanting and horizontal scratch defects encompass most of the image, reducing the focus loss effectiveness.

Conclusion

In this research paper, we addressed the critical problem of steel defect detection and benchmarked five state-of-the-art models on the NEU steel dataset, providing an in-depth analysis of each. Our findings indicate that the deformable convolutional network (DCN) model achieves the highest accuracy compared to other models, while RetinaNet, a single-stage detector, attains perfect precision with the least inference time.

Each model has its strengths and weaknesses, both overall and within each class. Identifying these strengths and addressing the models' weaknesses could pave the way for improved models, significantly benefiting the defect detection industry. Deep learning models, particularly convolutional neural networks (CNNs), have the distinct advantage of scalability and adaptability. They can handle more extensive and complex datasets, learning directly and adaptively from raw data without manual feature engineering. This ability to understand intricate patterns and representations is a significant advantage over traditional machine learning models, making them highly efficient for tasks such as image classification and object detection. Furthermore, the performance of deep learning models can improve with more data or sophisticated architectures.

We acknowledge that our study might not bring out the best performance of each model due to the consistent set of hyperparameters used across all models. In future work, we plan to include a comprehensive hyperparameter tuning process for each model to optimize their performance further. We also plan to extend our experiments to other datasets to validate the generalizability of our findings and further improve the models' performance in steel defect detection.

Further improvements could also be achieved by adding more layers and trainable parameters to the models. Future research could explore unsupervised learning approaches for defect detection and enhance the detection of crazing defects by incorporating synthetic training data through data augmentation. Integrating active learning and human-in-the-loop systems could also be beneficial. In addition, we plan to include "true negative" examples in our future work and evaluate the models' performance in a more realistic scenario. This will provide valuable insights into the practical applicability of these models for detecting steel defects. It's important to note that our study was limited to the NEU dataset. In future work, we plan to consider more datasets to validate the generalizability of our findings and further to improve the models' performance in steel defect detection.

Data availability The dataset used in this study is available at the following website: https://github.com/siddhartamukherjee/NEU-DET-Steel-Surface-Defect-Detection/tree/master/IMAGES. Researchers interested in accessing the dataset can do so by visiting the provided link.

Declarations

Conflict of interest No funds, grants, or other support was received.

References

1. Association WS. 2020 world steel in figures. Steel statistical yearbook 2020. 2020.
2. Hendy C, Iles D. Guidance Notes on Best Practice in Steel Bridge Construction. 2015. SCI P185. ISBN: 978-1-85942-217-5. Steel Construction Institute, Silwood Park, Ascot, Berkshire, SL5 7QN, UK
3. Yu H-L, Tieu K, Lu C, Deng G-Y, Liu X-H. Occurrence of surface defects on strips during hot rolling process by fem. Int J Adv Manuf Technol. 2013;67(5):1161–70.
4. Tian R, Jia M. Dcc-centernet: a rapid detection method for steel surface defects. Measurement. 2022;187: 110211.
5. Piccinini P, Prati A, Cucchiara R. Real-time object detection and localization with sift-based clustering. Image Vis Comput. 2012;30(8):573–87 (**Special Section: Opinion Papers. [Online]**).
6. Farooq J. Object detection and identification using SURF and BoW model. In: 2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube); 2016 Apr 11-12; Quetta, Pakistan. Piscataway, NJ: IEEE; 2016. p. 318-323. Available from: https://doi.org/10.1109/ICECUBE.2016.7495245
7. Campos M, Martins T, Ferreira M, Santos C. Detection of defects in automotive metal components through computer vision. IEEE Int Symp Ind Electron. 2008;2008:860–5.
8. Weber I, Bongartz J, Roscher R. Artifive-potsdam: a benchmark for learning with artificial objects for improved aerial vehicle detection. IEEE Int Geosci Remote Sens Symp IGARSS. 2021;2021:1214–7.
9. Chen P, Elangovan V. Object sorting using faster r-cnn. 2020. arXiv preprint: arXiv:2012.14840.
10. Feng X, Gao X, Luo L. X-SDD: A New Benchmark for Hot Rolled Steel Strip Surface Defects Detection. Symmetry. 2021;13(4):706. https://doi.org/10.3390/sym13040706
11. Abu M, Binti Zahri NAH, Amir A, Azemi S. The performance analysis of transfer learning for steel defect detection by using deep learning. J Phys Conf Ser. 2021;1755:02.
12. Bao Y, Song K, Liu J, Wang Y, Yan Y, Yu H, Li X. Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. IEEE Trans Instrum Meas. 2021;70:1–11.
13. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst. 2015;28:91–9.
14. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable detr: deformable transformers for end-to-end object detection. 2020. arXiv preprint: arXiv:2010.04159
15. Wu Y, Chen Y, Yuan L, Liu Z, Wang L, Li H, Fu Y. Rethinking Classification and Localization for Object Detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13-19; Seattle, WA, USA. Piscataway, NJ: IEEE; 2020. p. 10183–10192. Available from: https://doi.org/10.1109/CVPR42600.2020.01020
16. Lin T, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell. 2020;42(02):318–27.
17. Ma P, Ma J, Wang X, Yang L, Wang N. Deformable convolutional networks for multi-view 3d shape classification. Electron Lett. 2018;54(24):1373–5.
18. Jonker PP, Duin RP, de Ridder D. Pattern recognition for metal defect detection. Steel Grips. 2003;1(1):20–3.
19. Jia H, Murphey YL, Shi J, Chang TS. An intelligent real-time vision system for surface defect detection. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004); 2004 Aug 23-26; Cambridge, UK. Vol 3. Piscataway, NJ: IEEE; 2004. p. 239–242. Available from: https://doi.org/10.1109/ICPR.2004.1334512
20. Pernkopf F. Detection of surface defects on raw steel blocks using bayesian network classifiers. Pattern Anal Appl. 2004;7(3):333–42.
21. Wu XY, Xu K, Xu JW. Application of Undecimated Wavelet Transform to Surface Defect Detection of Hot Rolled Steel Plates.

In: 2008 Congress on Image and Signal Processing; 2008 May 27-30; Sanya, China. Piscataway, NJ: IEEE; 2008. p. 528–532. Available from: https://doi.org/10.1109/CISP.2008.278

22. Yazdchi M, Yazdi M, Mahyari AG. Steel surface defect detection using texture segmentation based on multifractal dimension. Int Conf Digit Image Process. 2009;2009:346–50.
23. Borselli A, Colla V, Vannucci M, Veroli M. A fuzzy inference system applied to defect detection in flat steel production. In: FUZZ-IEEE 2010, IEEE International Conference on Fuzzy Systems; 2010 Jul 18-23; Barcelona, Spain. Piscataway, NJ: IEEE; 2010. p. 1–6. Available from: https://doi.org/10.1109/FUZZY.2010.5584036
24. Li XG, Miao CY, Wang J, Zhang Y. Automatic Defect Detection Method for the Steel Cord Conveyor Belt Based on Its X-Ray Images. In: 2011 International Conference on Control, Automation and Systems Engineering (CASE); 2011; Singapore. Piscataway, NJ: IEEE; 2011. p. 1–4. Available from: https://doi.org/10.1109/ICCASE.2011.5997624
25. Liu K, Wang H, Chen H, Qu E, Tian Y, Sun H. Steel surface defect detection using a new haar-weibull-variance model in unsupervised manner. IEEE Trans Instrum Meas. 2017;66(10):2585–96.
26. Neogi N, Mohanta DK, Dutta PK. Review of vision-based steel surface inspection systems. EURASIP J Image Video Process. 2014;2014(1):1–19.
27. Tang B, Chen L, Sun W, Lin Z-K. Review of surface defect detection of steel products based on machine vision. IET Image Process. 2023;17(2):303–22.
28. Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M. Deep learning for generic object detection: a survey. Int J Comput Vis. 2020;128:261–318.
29. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27-30; Las Vegas, NV, USA. Piscataway, NJ: IEEE; 2016. p. 770–778. Available from: https://doi.org/10.1109/CVPR.2016.90
30. Tulbure A-A, Tulbure A-A, Dulf E-H. A review on modern defect detection models using dcnns-deep convolutional neural networks. J Adv Res. 2022;35:33–48.
31. Kim MS, Park T, Park P. Classification of Steel Surface Defect Using Convolutional Neural Network with Few Images. In: 2019 12th Asian Control Conference (ASCC); 2019; Kitakyushu, Japan. Piscataway, NJ: IEEE; 2019. p. 1398–1401.
32. He Y, Song K, Meng Q, Yan Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. IEEE Trans Instrum Meas. 2020;69(4):1493–504.
33. Lv X, Duan F, Jiang J-J, Fu X, Gan L. Deep metallic surface defect detection: the new benchmark and detection network. Sensors. 2020;20(6):1562.
34. Cheng X, Yu J. Retinanet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. IEEE Trans Instrum Meas. 2021;70:1–11.
35. Tang M, Li Y, Yao W, Hou L, Sun Q, Chen J. A strip steel surface defect detection method based on attention mechanism and multi-scale maxpooling. Meas Sci Technol. 2021;32(11): 115401.
36. Li Z, Tian X, Liu X, Liu Y, Shi X. A two-stage industrial defect detection framework based on improved-yolov5 and optimized-inception-resnetv2 models. Appl Sci. 2022;12(2):834.
37. Singh, S.A., Desai, K.A. Automated surface defect detection framework using machine vision and convolutional neural networks. J Intell Manuf 34, 1995–2011 (2023). https://doi.org/10.1007/s10845-021-01878-w.
38. Konovalenko I, Maruschak P, Brevus V. Steel surface defect detection using an ensemble of deep residual neural networks. J Comput Inf Sci Eng. 2021;22: 014501. https://doi.org/10.1115/1.4051435.
39. Zhu L, Baolin D, Xiaomeng Z, Shaoliang F, Zhen C, Junjie Z, Shumin C. Surface defect detection method based on improved semisupervised multitask generative adversarial network. Sci Program. 2022;2022:1–17.
40. Song K, Yan Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci. 2013;285:858–64.
41. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal Loss for Dense Object Detection. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):318–327. https://doi.org/10.1109/TPAMI.2018.2858826.
42. R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai, vol. 14(2). Montreal. 1995. p. 1137–45.
43. Bottou L, Stochastic gradient descent tricks. In: Neural networks: tricks of the trade, Second Edition. Springer. 2012. p. 421–36.
44. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European conference on computer vision. Springer. 2020. p. 213–29.
45. Chazhoor AAP, Ho ES, Gao B, Woo WL. Deep transfer learning benchmark for plastic waste classification. Intell Robot. 2022;2:1–19.
46. Chazhoor AAP, Zhu M, Ho ESL, Gao B, Woo WL. Intelligent Classification of Different Types of Plastics using Deep Transfer Learning. Proceedings of the 2nd International Conference on Robotics, Computer Vision and Intelligent Systems, ROBOVIS 2021; 2021 Oct 27-28; Online Streaming. SCITEPRESS; 2021. p. 190–195. Available from: https://doi.org/10.5220/0010716500003061
47. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J et al. Learning transferable visual models from natural language supervision. In: International conference on machine learning. PMLR. 2021. p. 8748–63.
48. Arafin P, Issa A, Billah AM. Performance comparison of multiple convolutional neural networks for concrete defects classification. Sensors. 2022;22(22):8714.
49. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proc IEEE Conf Comput Vis Pattern Recognit. 2016;10:770–8.
50. Yang J, Wang W, Lin G, Li Q, Sun Y, Sun Y. Infrared thermal imaging-based crack detection using deep learning. IEEE Access. 2019;7:182060–77.
51. Chiu M-C, Tsai H-Y, Chiu J-E. A novel directional object detection method for piled objects using a hybrid region-based convolutional neural network. Adv Eng Inform. 2022;51: 101448.
52. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. In: European conference on computer vision. Springer. 2020. p. 213–29.
53. Jin J, Feng W, Lei Q, Gui G, Wang W. PCB defect inspection via Deformable DETR. In: 2021 7th International Conference on Computer and Communications (ICCC 2021); 2021 Dec 10-13; Chengdu, China. Piscataway, NJ: IEEE; 2021. p. 646–651. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9674579
54. Zhou Q, Li X, He L, Yang Y, Cheng G, Tong Y, Ma L, Tao D. Transvod: end-to-end video object detection with spatial-temporal transformers. 2022. arXiv preprint: arXiv:2201.05047.
55. Yao Z, Gholami A, Shen S, Mustafa M, Keutzer K, Mahoney MW. Adahessian: an adaptive second order optimizer for machine learning. 2020. arXiv preprint: arXiv:2006.00719.

56. Song G, Liu Y, Wang X. Revisiting the Sibling Head in Object Detector. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020; Seattle, WA, USA. Piscataway, NJ: IEEE; 2020. p. 11560–11569. Available from: https://api.semanticscholar.org/CorpusID:212737107
57. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y. Deformable convolutional networks. IEEE Int Conf Comput Vis (ICCV). 2017;2017:764–73.
58. Chen K, Wang J, Pang J, Cao Y, Xiong Y, Li X, Sun S, Feng W, Liu Z, Xu J, Zhang Z, Cheng D, Zhu C, Cheng T, Zhao Q, Li B, Lu X, Zhu R, Wu Y, Dai J, Wang J, Shi J, Ouyang W, Loy CC, Lin D. Mmdetection: Open mmlab detection toolbox and benchmark. CoRR. 2019. arXiv:1906.07155
59. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer vision-ECCV 2014. Cham: Springer International Publishing; 2014. p. 740–55.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.