Automated Detection of Defects on Metal Surfaces Using Vision Transformers

Toqa Alaa[1], Mostafa Kotb[1], Arwa Zakaria[1], Mariam Diab[1] and Walid Gomaa[1,2]
[1]*Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology, Alexandria, Egypt*
[2]*Faculty of Engineering, Alexandria University, Alexandria, Egypt*
{*toqa.alaa, mostafa.kotb, arwa.zakaria, mariam.diab, walid.gomaa*}*@ejust.edu.eg*

Keywords: Vision Transformers, Classification, Localization, Convolution Neural Networks, GC10-DET, NEU-DET, Multi-DET.

Abstract: Metal manufacturing often results in the production of defective products, leading to operational challenges. Since traditional manual inspection is time-consuming and resource-intensive, automatic solutions are needed. The study utilizes deep learning techniques to develop a model for detecting metal surface defects using Vision Transformers (ViTs). The proposed model focuses on the classification and localization of defects using a ViT for feature extraction. The architecture branches into two paths: classification and localization. The model must approach high classification accuracy while keeping the Mean Square Error (MSE) and Mean Absolute Error (MAE) as low as possible in the localization process. Experimental results show that it can be utilized in the process of automated defects detection, improve operational efficiency, and reduce errors in metal manufacturing.

1 INTRODUCTION

The manufacturing and reshaping of metal surfaces are critical processes in various industries, including automotive, aerospace, and construction. These processes often result in products with defects such as cracks, dents, scratches, and other surface irregularities. Such defects can compromise the structural integrity and performance of metal products, posing significant challenges to quality control and product usability. Detecting and addressing these defects is crucial to ensure the production of high-quality metal products and to prevent costly operational failures (Wang et al., 2021; Murakami, 2019; Leibfried and Breuer, 2006).

Traditionally, defect detection on metal surfaces has relied heavily on manual inspection, where human experts visually examine surfaces for abnormalities. This method is not only time-consuming and labor-intensive but also highly subjective and inconsistent. It is prone to errors and often fails to detect subtle defects that are not easily visible to the human eye. Consequently, there is a compelling need for automated defect detection systems that can accurately and efficiently identify and classify defects on metal surfaces (Li et al., 2022; Fang et al., 2020).

Significant progress has been made in developing automated defect detection techniques. Traditional computer vision methods, such as edge detection, thresholding, Hough transfrom, and image segmentation, have been extensively explored. These methods typically rely on handcrafted features and rule-based algorithms to identify defects based on predefined criteria. Although these approaches have achieved some success in detecting specific types of defects, they are limited in their ability to handle complex and varied defect patterns. They depend heavily on the expertise of domain-specific engineers and lack the adaptability to new or varied defect types (Sharifzadeh et al., 2009).

With the advent of deep learning techniques and the availability of large-scale annotated datasets, there has been a paradigm shift towards employing neural networks for automated defect detection. Convolutional Neural Networks (CNNs) have demonstrated remarkable performance in various computer vision tasks, including image classification, object detection, and semantic segmentation. CNNs can automatically learn discriminative features from raw input data, making them well-suited for defect detection on metal surfaces. Several studies have successfully applied CNNs to detect defects on metal surfaces, achieving high accuracy and demonstrating the

Alaa, T., Kotb, M., Zakaria, A., Diab, M. and Gomaa, W.
Automated Detection of Defects on Metal Surfaces Using Vision Transformers.
DOI: 10.5220/0012936300003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 36-45
ISBN: 978-989-758-717-7; ISSN: 2184-2809

potential of deep learning in this domain. For example, U-Net-based CNN architectures have been applied to metal surface defect detection, showing improved performance in capturing fine-grained details of surface defects (Konovalenko et al., 2022).

Despite their promise, CNNs have limitations in defect detection. They typically rely on local receptive fields and hierarchical feature extraction, which may not adequately capture long-range dependencies and global context in images. Studies have shown that varying lighting conditions can impact the performance of neural network models in detecting defects, which highlights the importance of surface illumination factor to ensure a reliable performance of models (Maruschak et al., 2024).

This limitation is particularly critical when dealing with complex defect patterns that span significant portions of metal surfaces. Additionally, CNNs require large amounts of labeled training data, which can be challenging and time-consuming to acquire for specific defect types or rare occurrences (Tao et al., 2018).

To address these limitations, we propose the use of Vision Transformers (ViTs) for automated defect detection on metal surfaces. ViTs, originally introduced for image classification, (Dosovitskiy et al., 2020) have gained attention for their ability to capture global context and long-range dependencies through self-attention mechanisms (Vaswani et al., 2017). This makes them well-suited for capturing intricate defect patterns on metal surfaces.

To address the limitations of the current datasets used in metal surface detection, a new dataset called Multi-DET is built. Current datasets don't accurately simulate real-world conditions, as metal surfaces typically have more overlapping and higher number of defects per image. Our new dataset, Mult-DET, addresses these limitations by introducing diversity and increased density per image.

The primary objectives of this research are twofold: defect classification and defect localization. Defect classification aims to accurately identify the type and nature of each defect. Defect localization aims to precisely predict the boundaries of each defect, facilitating targeted treatment and repair. The proposed model should be able to detect multiple defects in the input image.

Leveraging the power of pre-trained ViTs on large-scale image datasets like Imagenet, the proposed model utilizes transfer learning to benefit from the learned representations of ViTs, which capture rich visual features. This pre-trained model is able to effectively extract meaningful defect-related features from raw metal surface images. So, we propose using Vision Transformers and deep learning techniques to automate defect detection on metal surfaces, aiming to enhance product quality and reduce costly errors in metal manufacturing.

The paper is organized as follows. Section 1 is an introduction, offering an overview of the problem and the undergoing research. Section 2 discusses related work. Section 3 explores the used datasets and our new dataset. Section 4 discusses the methodologies used for defect detection and localization. Section 5 gives the experimental work to validate our approach along with analyzing the results, illustrating as well the limitations of our approach. Section 6 summarizes our work and provides an outlook on future directions.

2 RELATED WORK

Metal defects detection is a critical task in industrial applications. Over the years, researchers have developed methods to identify and classify these defects using machine learning and computer vision (Wang et al., 2021). The advancement of these methods increases the efficiency and accuracy of the process to ensure quality and reliability of metal products in industries.

2.1 Traditional Approaches

Metal defects initially relied on manual inspection, including Magnetic Particle Inspection (MPI), Ultrasonic Testing (UT), and Dye Penetrant Inspection (DPI) (Lovejoy, 1993). While these approaches have been fundamental in ensuring the quality of metals, they come with limitations such as inconsistency and potential human error.

2.2 CNN

A compact Convolutional Neural Network was used alongside a cascaded autoencoder (CASAE) in the task of metal defects detection (Tao et al., 2018). The compact CNN architecture aimed to classify defects, while the CASAE was used to localize and segment defects. The usage of CASAE resulted in accurate and consistent results even under complex lighting conditions or with irregular defects. The pipeline of the architecture started with passing the input image to the CASAE, which outputs a segmented image of the defects. The segments are then cropped and fed to the compact CNN to obtain classification results. Nevertheless, the architecture had limitations, as the input data must be manually labelled as segments, not

as bounding boxes, which takes a lot of time and expense.

2.3 RepVGG

The authors of (Li et al., 2022) provided a reference for solving the problem of classifying aluminum profile surface defects. Defective images for training were obtained by means of digital image processing, such as rotation, flip, brightness, and contrast transformation. A novel model, RepVGG, with a convolutional block attention module (RepVGG-CBAM) was proposed. The model was used to classify ten types of aluminum profile surface defects.

2.4 Faster R-CNN

Another novel approach proposes a method combining a classification model with an object recognition model for metal surface defects detection (Wang et al., 2021). An improved, faster R-CNN model is used to detect multi-scale defects better by adding spatial pyramid pooling (SPP) (Mikołajczyk et al., 2017) and enhanced feature pyramid networks (FPN) modules (Girshick et al., 2017). The model aims to increase the detection accuracy of crazing defects by modifying the aspect ratio of the anchor. Non-maximum suppression is used to get the bounding box faster and better. Improved ResNet50-vd model is incorporated as the backbone of the classification model and object recognition model (He et al., 2019). Detection accuracy and robustness are increased by adding the deformable revolution network (DCN) (Zhu et al., 2018).

2.5 YOLOv5

Among the deep learning models, the You Only Look Once (YOLO) algorithm stands out for its capabilities for object detection (Redmon et al., 2016). YOLO reframes object detection as a single regression problem, predicting bounding boxes and class probabilities directly from full images in one evaluation. YOLOv5 (Bochkovskiy et al., 2020) is a famous version among the YOLO versions, as it was the first version with ultralytics support. YOLOv5 was used as a base model in metal defects detection (Wang et al., 2022), while adding a focus structure to the base network of the Darknet. Additionally, GIOU loss was chosen over L1 loss to focus on accuracy. The backbone feature extraction network of the YOLOv5 was retrained to improve the performance of the model. However, this model faced limitations due to its inability to detect small defects on metal surfaces.

3 DATASETS

Several datasets have been created to provide research with standardized data for training and evaluating defects detection algorithms. This section focuses on three datasets - NEU-DET, GC10-DET and Multi-DET that were used for the work done in this research.

3.1 Training Datasets

GC10-DET is a dataset for surface defects collected in a real industry (Lv et al., 2020). The dataset contains 2300 high-resolution images of surfaces with 10 different classes of defects, which are punching, weld line, crescent gap, water spot, oil spot, silk spot, inclusion, rolled pit, crease, and waist folding.

NEU-DET is the Northeastern University (NEU) surface defect dataset (Dixit, 2020). It contains 1800 grayscale images with 6 different classes of surface defects and 300 images per defect type. The defects' classes include rolled-in scale, patches, inclusion, scratches, crazing, and pitted surface. Figure 1 shows different classes from the two datasets.

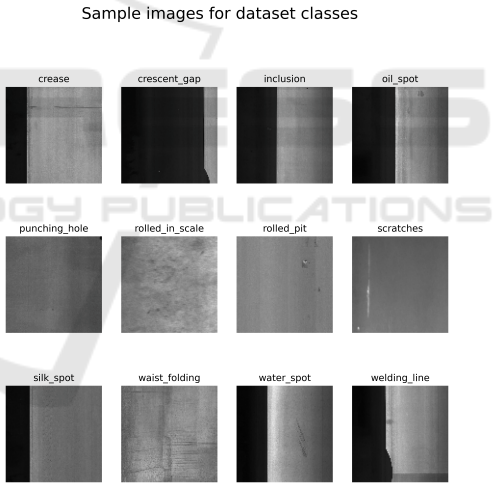

Figure 1: Defect Classes from GC10-DET and NEU-DET.

3.2 Multi-DET

We introduce a new dataset called Multi-DET in order to address the limitations of the current datasets. The proposed dataset surpasses current datasets by offering increased diversity and density of defects per photo.

Multi-DET contains 300 high-resolution images for 8 classes. Unlike traditional datasets that feature repetitive defect types per image, our dataset covers a wide range of defects, including scratches, welding line, inclusion, water spot, oil spot, crescent gap , and

variations in texture and color. Our approach mimics real-world conditions, where metal surfaces exhibit complex and overlapping defects. Figure 2 represents some samples of Multi-DET dataset.

3.2.1 Dataset Collection

The dataset collection process started with surface preparation, where metal samples were cleaned and smoothed to ensure a uniform pattern. Following this, different defects were introduced using various tools. Scratches were made using sharp instruments, welding lines were simulated using a welding machine, crescent gap were made using precise cutting tools, and inclusions were created using contaminant materials. Oil and water spots were applied using controlled droplets.

3.2.2 Dataset Pre-Processing

The pre-processing of photos in Multi-DET is a crucial step in order to ensure quality and the uniformity in terms of resolution, lighting, and color balance. Pre-processing involves adjusting the brightness and contrast to compensate for any variations. In addition, image denoising is used to remove any unwanted noise that may interfere with the detection process. Image cropping is performed to focus on relevant parts of the metal, excluding information that does not contribute to the analysis. Images are converted to grayscale to enhance the feature extraction process. Data augmentation techniques are utilized to increase the variability of the dataset. This includes rotating, flipping, and scaling. These transformations represent the diversity of real-world conditions, therefore it was avoided to reach extreme levels of these changes.

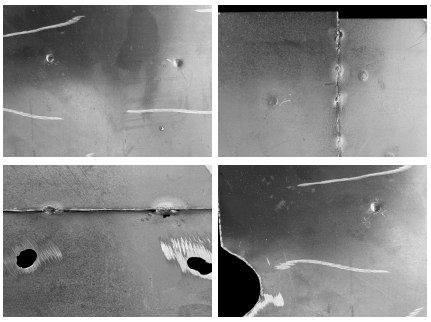

Figure 2: Sample photos of Multi-DET dataset.

3.2.3 Data Annotations

For annotating Multi-DET dataset, Roboflow serves as the primary tool for creating the annotations for our dataset. Figure 3 illustrates the process of annotations.

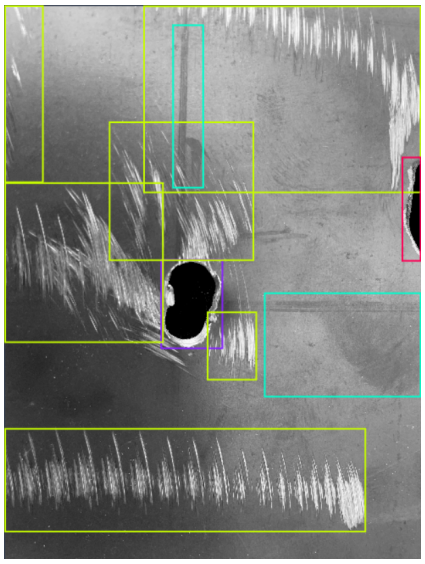

Figure 3: Defects bounding boxes annotation.

4 EXPERIMENTAL WORK

4.1 Methodology

This research presents an automated approach for detecting defects on metal surfaces utilizing the Vision Transformer (ViT) architecture. ViTs achieve enhanced accuracy via the self-attention mechanism inherent in transformer encoders. We utilize the ViT's encoder as a feature extractor, which is then forwarded to a CNN, followed by a couple of Multi-Layer Perceptron (MLP) models for classification and localization. For the detection mechanism, we implement the anchor boxes method in order to dynamically detect any number of defects within the input image. This repository includes the source code for our implementation *.

4.2 Vision Transformers

The architecture, Figure 4, of the ordinary transformer (Vaswani et al., 2017) was initially applied in Natural Language Processing (NLP). The main purpose of the transformer was to detect the relationships between the tokens through the self-attention mechanism. The architecture also added positional encoding to the input of the multi-head attention layers, which

*https://github.com/toqaalaa20/Metal-surface-defects-detection

allowed the transformer to keep track of the order of the tokens.

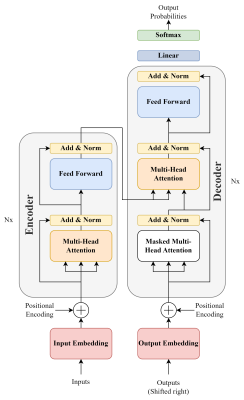

Figure 4: Transformer architecture.

The architecture in Figure 5 of the Vision Transformer (ViT) used the same basic structure of the transformer with the multi-head attention layers followed by the MLPs and the normalization layers (Dosovitskiy et al., 2020). However, the modification was on the input of the model, as the input was modified to take an image instead of a sequence of words. To use the same structure of the transformer, an input image is divided into patches and flattened. Positional embedding is added to the patches to keep track of the position of each patch. The encoded patches are fed to the ViT encoder. In order to perform any task of classification or localization, a learnable MLP is added to the output of the encoder.

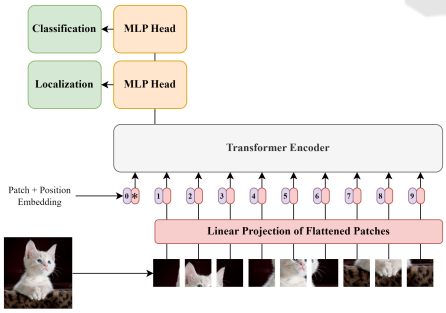

Figure 5: Vision Transformer Architecture.

4.3 Anchor Boxes

In order to dynamically detect any number of defects in an image without being limited to a fixed number of defects per image, the anchor boxes method was used. The mechanism of anchor boxes was first introduced as a part of the You Only Look Once (YOLO) model architecture (Redmon et al., 2016). At first, a set of pre-defined anchor boxes (Figure 6) is defined.

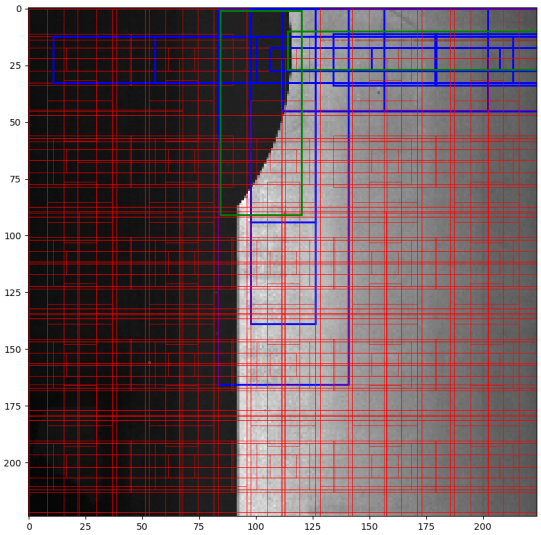

Figure 6: Initialized pre-defined anchor boxes. Red boxes are background boxes. Blue boxes are selected boxes before offset correction. Green boxes are the ground truth.

During training, each anchor box is assigned to a ground truth label depending on the Intersection over Union (IOU) value. If the IoU value is higher than an upper threshold, the anchor box gets assigned to a ground truth label. If the IoU is lower than a lower threshold, it is marked as background. If it is between the two thresholds, it is marked as discarded.

After that, the offset in position from the ground truth is calculated for the assigned anchor boxes and set to zero for the background and discarded boxes. The class of the ground truth is also passed to the assigned anchor boxes. During prediction, a non-maximal suppression is applied on the predicted anchor boxes in order to choose the most suitable anchor box from the overlaying boxes with the same predicted class.

4.4 Data Pre-Processing

In order to fit our model architecture, all images were resized and normalized to fit the input of the ViT encoder. Then, they were passed to an image processor, which prepares the images to be a suitable input for the ViT. The image processor ensures that the input data is in the correct format suitable for the ViT. It handles image transformations such as resizing, normalization, and conversion to TensorFlow tensors.

After that, offsets are calculated for each image as the distance between the anchor box point and the ground truth point and divided by either the width or the height of the anchor box according to the orientation. This makes the offsets invariant to different scales of anchor boxes. Afterwards, the offsets are

normalized by passing on a min-max scaler which subtracts the minimum value from the offset and divides the result by the range. All these normalizations were made to ensure the stability of the model and help it predict the values without overshooting or overfitting.

4.5 Model

Our model architecture (Figure 7) consists mainly of 4 parts: ViT encoder, CNN, Classification MLP, Regression MLP.

4.5.1 Feature Extraction

After the image is pre-processed using the image processor, it is passed to the encoder of the ViT model. The output embeddings of the encoder are then passed on a CNN to process these embeddings and extract the features from them.

The output of the CNN is then flattened and shared to two different MLPs. The share mechanism is selected over using two different models for classification and localization to allow the model's CNN to learn the common features between classification and localization. This will allow the model to understand the image features better and connect the two parts of detection together.

4.5.2 Detection

Each MLP consists of multiple dense and dropout layers followed by an output layer. The output layer is then reshaped to (number of anchor boxes, number of classes) in the classification MLP and reshaped to (number of anchor boxes, 4) in the regression MLP. This is done to apply Softmax in classification and sigmoid in regression individually on each sample (anchor box). Sigmoid is used as the values of the offsets are limited between 0 and 1 after scaling. This reshape will yield more meaningful results as applying Softmax on the whole number of samples will yield false results.

The output of the Softmax layers and sigmoid layers is then concatenated and reshaped to match the output shape, which is [(number of anchor boxes, number of classes), (number of anchor boxes, 4)].

4.6 Loss Functions

The model is compiled with modified versions of Categorical Cross-Entropy and Mean Square Error (MSE) for classification and regression, respectively.

4.6.1 Classification Head

In categorical cross-entropy, the modification aimed to handle the fact that most anchor boxes will be assigned to the background class, as found in the training datasets, which will lead towards extreme bias towards the background class. In other words, it would be easier for the model to predict all classes as background than actually spotting features in the image. To eliminate this bias, the categorical cross-entropy is performed individually on each detection (instead of grouping by all defects in the image) only if the true value is not a background class. This is to mimic a two-level categorical cross-entropy, a lower level on each sample, and a higher level on the whole image with all samples. This modification aims to stop giving the model a positive score on detecting the background class, which prevents bias towards this class. Algorithm 1 describes how the modified function works.

Algorithm 1: Modified Categorical Cross-entropy.

Data: y_{true}, y_{pred}
Result: *Categorical Cross Entropy Value*
$N \leftarrow len(y_{\text{true}})$;
$M \leftarrow len(y_{\text{true}}[0])$;
$loss \leftarrow 0$;
for $i \leftarrow 0$ **to** N **do**
  for $j \leftarrow 0$ **to** M **do**
    if $argmax(y_{true}[i][j]) \neq background$ **then**
      $loss \leftarrow loss + \sum_{m=1}^{M} y_{\text{true}}[i][j][m] \log(y_{\text{pred}}[i][j][m])$;
    end
  end
end
return *loss*

4.6.2 Localization Head

In MSE, the modification aimed to handle the problem that most offsets will be 0, due to the fact that most anchor boxes are from the background class, as found in the training datasets. The modification was to iterate over all detections and calculate the MSE individually for each detection only if the true values are not zeros (background class, and then output the total MSE. This modification results in lowering the bias towards the background class of offset 0. Algorithm 2 describes how the modified function works.

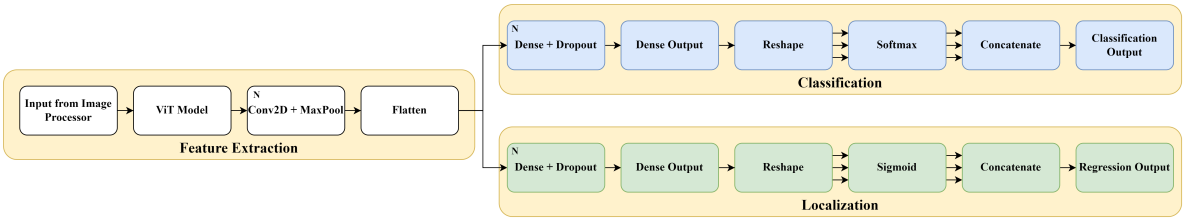

Figure 7: Our Model Architecture.

Algorithm 2: Modified Mean Squared Error.

Data: y_{true}, y_{pred}
Result: *Mean Squared Error Value*
$N \leftarrow len(y_{\text{true}})$;
$M \leftarrow len(y_{\text{true}}[0])$;
$loss \leftarrow 0$;
$count \leftarrow 0$;
for $i \leftarrow 0$ **to** N **do**
  if $\sum_{m=1}^{M}(y_{true}[i][m]) \neq 0$ **then**
    $loss \leftarrow loss +$
    $\sum_{m=1}^{M}(y_{\text{true}}[i][m] - y_{\text{pred}}[i][m])^2$;
    $count \leftarrow count + 1$;
  end
end
$loss \leftarrow \frac{loss}{count}$;
return $loss$

5 EVALUATION

This section discusses the evaluation metrics used to assess our model, the obtained results, the strengths, and the limitations of our approach.

5.1 Metrics

Evaluation metrics are essential for assessing the performance our model. In this study, we employ three key metrics: a modified version of accuracy for defect classification, a modified version of Mean Absolute Error for bounding box regression, and Mean Intersection over Union (Mean IOU) for bounding box localization.

5.1.1 Accuracy: Defect Classification

The modification on accuracy followed the same procedures mentioned in Algorithm 1. The accuracy measures the proportion of the correctly classified defect instances out of the total instances.

$$\text{Accuracy} = \frac{\text{Number of correctly classified defects}}{\text{Total number of defects}} \tag{1}$$

5.1.2 Mean Absolute Error: Bounding Box Regression

The modification on MAE followed the same procedures mentioned in Algorithm 2. MAE quantifies the average absolute distance between predicted bounding box coordinates $\hat{B}_i = (x_{\hat{B}_i}, y_{\hat{B}_i}, w_{\hat{B}_i}, h_{\hat{B}_i})$, and ground truth bounding box coordinates $B_i = (x_{B_i}, y_{B_i}, w_{B_i}, h_{B_i})$ for each instance i.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\Big(|x_{\hat{B}_i} - x_{B_i}| + |y_{\hat{B}_i} - y_{B_i}| + |w_{\hat{B}_i} - w_{B_i}| + |h_{\hat{B}_i} - h_{B_i}|\Big) \tag{2}$$

where n is the total number of instances.

5.1.3 Mean IOU

Mean IOU measures the spatial overlap between predicted and ground truth bounding boxes across all defect instances.

$$\text{IOU}(B_i, \hat{B}_i) = \frac{\text{Area of overlap}(B_i, \hat{B}_i)}{\text{Area of union}(B_i, \hat{B}_i)} \tag{3}$$

where:

- Area of overlap$(B_i, \hat{B}_i)$ is the area where the predicted and ground truth bounding boxes overlap.
- Area of union$(B_i, \hat{B}_i)$ is the area encompassed by both the predicted and ground truth bounding boxes.

Mean Intersection over Union (Mean IOU) is calculated as the average IOU across all bounding boxes:

$$\text{Mean IOU} = \frac{1}{n}\sum_{i=1}^{n}\text{IOU}(B_i, \hat{B}_i) \tag{4}$$

5.2 Results

This section emphasizes the results of the loss functions, and the evaluation metrics illustrated earlier. The figures compare the results obtained by training our data on our model with and without using the ViT

as our base feature extractor. Figure 8 shows the training versus validation accuracy and MAE without using ViT as a feature extractor. As shown in the figure, the model is overfitting our data as there is a significant gap between the training and the validation results. Figure 9 shows the training and the validation accuracy using the ViT. Figure 10 shows the loss using the ViT. Figure 11 shows the MAE using ViT. As the figures show, using ViT as the feature extractor has addressed the problem of over-fitting and achieved high performance.

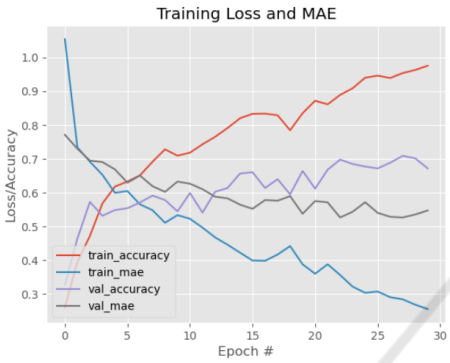

Figure 8: Evaluation metrics without using ViT.

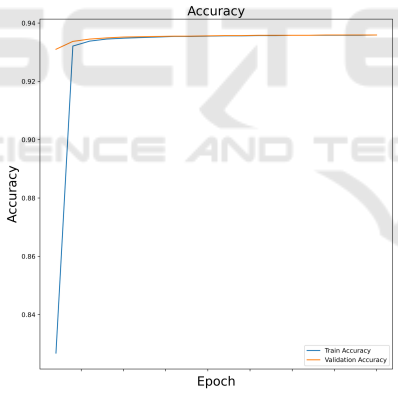

Figure 9: Model accuracy using ViT.

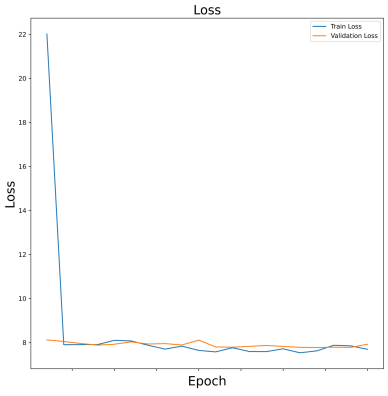

Figure 10: Model loss using ViT.

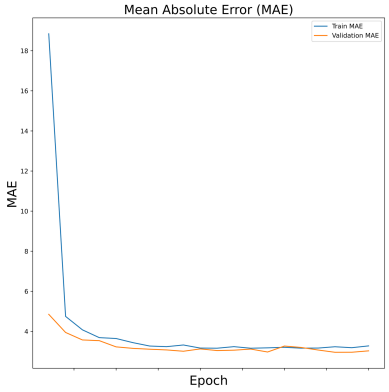

Figure 11: MAE using ViT.

5.3 Discussion

In this study, we evaluated a metal surface detection algorithm using classification accuracy, MAE for bounding box regression, and Mean IOU for bounding box localization. Our findings indicate significant achievements and some areas for improvement.

5.3.1 Performance Evaluation

Our model has achieved a high accuracy of 93.5% in classifying metal surface defects. This high accuracy emphasizes the effectiveness of our classification approach in detecting and distinguishing between different defects.

For bounding box regression, we calculated MAE to assess the accuracy of predicted bounding box coordinates compared to ground truth. The calculated MAE of 3.2 pixels suggests that our model can reasonably predict the dimensions and positions of the bounding boxes representing the defects.

Mean Intersection over Union (Mean IOU) was used to evaluate the spatial overlap between predicted and ground truth bounding boxes. Our model achieved a Mean IOU score of 0.72, indicating strong performance in accurately localizing metal defects within the bounding boxes.

The model has achieved State-of-the-Art results, as shown in Table 1. Our proposed methodology achieved a mean average precision (mAP) score of 0.732, which indicates the strong ability of our model to classify different types of defects. It has beaten the YOLO-backbone methods as indicated in the table.

5.3.2 Strengths and Limitations

Our study demonstrates several strengths, including robust classification accuracy and effective bounding box localization capabilities. These strengths highlight the potential of our approach to contribute to au-

Table 1: Comparison with State-of-the-Art Methods.

Model	mAP Score
SSD	0.634
Faster-RCNN	0.627
YOLOv5	0.573
YOLOv6	0.666
Proposed Model	0.732

tomated quality control processes in metal manufacturing industries.

However, our approach also has limitations. For instance, the current model may struggle with detecting highly irregular defects due to limitations in the training data. In addition, the localization and the classification processes are not fast enough due to the complex details of the transformer architecture. The multi-head attention mechanism and the numerous layers in the transformer architecture significantly increase computational overhead, which further slows down the processing speed and decreases computational efficiency.

Addressing these limitations could involve adding more samples to Multi-DET dataset to introduce these variations and incorporating further development on the vision transformer to optimize the detection and classification processes.

6 CONCLUSIONS

Automated defect detection on metal surfaces is a crucial research area as it contributes to various industries, like automotive and construction. Manual inspection methods are slow and subjective, calling for automated systems. This study proposes using Vision Transformers to overcome the limitations of traditional methods. ViTs, with their attention mechanisms, can capture complex defect patterns effectively. The research focuses on defect classification and localization, using pre-trained ViTs and transfer learning. By automating defect detection, the approach aims to improve product quality and reduce errors in metal manufacturing. The study addresses a research gap in applying ViTs to metal surface defect detection, contributing to the field. The promising results demonstrate accurate defect classification and precise defect localization. The proposed model achieved 93.5% accuracy in defect detection outperforming YOLO-based methods with a mean average precision of 0.732. These results demonstrate the model's performance and its potential impact across multiple industries.

Our methodology offers a promising approach for addressing the challenges posed by metal defects in manufacturing and reshaping industries. However, there is still room for improvement, particularly in addressing the model's capability for detecting extremely overlapping and irregular shapes of defects. This can be done by adding degrees of freedom to the model while augmenting the training dataset. In addition, optimizing the model to work in real-time will levitate the model's performance. This limitation is due to the complexity of the ViT. Despite the effectiveness, ViTs are known for their high computational demands in terms of memory and processing power. This can be challenging when deploying the model in real-time industrial settings.

Ultimately, this research paves the way for more effective defect detection, ensuring the production of high-quality metal products, and reducing operational challenges in various industries.

ACKNOWLEDGEMENTS

We would like to extend our sincere gratitude to Eng. Fatma Youssef, for her invaluable help and guidance throughout this project. Her expertise and thoughtful advice have played a crucial role in shaping the path and achievements of this research.

REFERENCES

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Dixit, K. (2020). Neu-det neu surface defect database.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

Fang, X., Luo, Q., Zhou, B., Li, C., and Tian, L. (2020). Research progress of automated visual surface defect detection for industrial metal planar materials. *Sensors*, 20(18):5136.

Girshick, R., Lin, T., Dollar, P., Belongie, S., et al. (2017). Feature pyramid networks for object detection. *Facebook AI Research (FAIR)(19 April 2017)*.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. (2019). Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567.

Konovalenko, I., Maruschak, P., Brezinová, J., Prentkovskis, O., and Brezina, J. (2022). Research of u-net-based cnn architectures for metal surface defect detection. *Machines*, 10(5).

Leibfried, G. and Breuer, N. (2006). *Point defects in metals I: introduction to the theory*, volume 81. Springer.

Li, Z., Li, B., Ni, H., Ren, F., Lv, S., and Kang, X. (2022). An effective surface defect classification method based on repvgg with cbam attention mechanism (repvgg-cbam) for aluminum profiles. *Metals*, 12(11):1809.

Lovejoy, M. (1993). *Magnetic particle inspection: a practical guide*. Springer Science & Business Media.

Lv, X., Duan, F., Jiang, J.-j., Fu, X., and Gan, L. (2020). Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6).

Maruschak, P., Konovalenko, I., Osadtsa, Y., Medvid, V., Shovkun, O., Baran, D., Kozbur, H., and Mykhailyshyn, R. (2024). Surface illumination as a factor influencing the efficacy of defect recognition on a rolled metal surface using a deep neural network. *Applied Sciences*, 14(6).

Mikołajczyk, T., Nowicki, K., Kłodowski, A., and Pimenov, D. Y. (2017). Neural network approach for automatic image analysis of cutting edge wear. *Mechanical Systems and Signal Processing*, 88:100–110.

Murakami, Y. (2019). *Metal fatigue: effects of small defects and nonmetallic inclusions*. Academic Press.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Sharifzadeh, M., Alirezaee, S., Amirfattahi, R., and Sadri, S. (2009). Detection of steel defect using the image processing algorithms. pages 125 – 127.

Tao, X., Zhang, D., Ma, W., Liu, X., and Xu, D. (2018). Automatic metallic surface defect detection and recognition with convolutional neural networks. *Applied Sciences*, 8(9):1575.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, K., Teng, Z., and Zou, T. (2022). Metal defect detection based on yolov5. *Journal of Physics: Conference Series*, 2218(1):012050.

Wang, S., Xia, X., Ye, L., and Yang, B. (2021). Automatic detection and classification of steel surface defect using deep convolutional neural networks. *Metals*, 11(3):388.

Zhu, X., Hu, H., Lin, S., and Dai, J. (2018). Deformable convnets v2: More deformable, better results. 2019 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9300–9308.