Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2025.058361

ARTICLE

Steel Surface Defect Detection Using Learnable Memory Vision Transformer

Syed Tasnimul Karim Ayon[1,#], **Farhan Md. Siraj**[1,#] **and Jia Uddin**[2,*]

[1]Department of Computer Science and Engineering, BRAC University, Dhaka, 1212, Bangladesh

[2]Department of AI and Big Data, Endicott College, Woosong University, Daejeon, 34606, Republic of Korea

*Corresponding Author: Jia Uddin. Email: jia.uddin@wsu.ac.kr

#These authors contributed equally to this work

Received: 10 September 2024 Accepted: 17 December 2024 Published: 03 January 2025

ABSTRACT

This study investigates the application of Learnable Memory Vision Transformers (LMViT) for detecting metal surface flaws, comparing their performance with traditional CNNs, specifically ResNet18 and ResNet50, as well as other transformer-based models including Token to Token ViT, ViT without memory, and Parallel ViT. Leveraging a widely-used steel surface defect dataset, the research applies data augmentation and t-distributed stochastic neighbor embedding (t-SNE) to enhance feature extraction and understanding. These techniques mitigated overfitting, stabilized training, and improved generalization capabilities. The LMViT model achieved a test accuracy of 97.22%, significantly outperforming ResNet18 (88.89%) and ResNet50 (88.90%), as well as the Token to Token ViT (88.46%), ViT without memory (87.18), and Parallel ViT (91.03%). Furthermore, LMViT exhibited superior training and validation performance, attaining a validation accuracy of 98.2% compared to 91.0% for ResNet18, 96.0% for ResNet50, and 89.12%, 87.51%, and 91.21% for Token to Token ViT, ViT without memory, and Parallel ViT, respectively. The findings highlight the LMViT's ability to capture long-range dependencies in images, an area where CNNs struggle due to their reliance on local receptive fields and hierarchical feature extraction. The additional transformer-based models also demonstrate improved performance in capturing complex features over CNNs, with LMViT excelling particularly at detecting subtle and complex defects, which is critical for maintaining product quality and operational efficiency in industrial applications. For instance, the LMViT model successfully identified fine scratches and minor surface irregularities that CNNs often misclassify. This study not only demonstrates LMViT's potential for real-world defect detection but also underscores the promise of other transformer-based architectures like Token to Token ViT, ViT without memory, and Parallel ViT in industrial scenarios where complex spatial relationships are key. Future research may focus on enhancing LMViT's computational efficiency for deployment in real-time quality control systems.

KEYWORDS

Learnable Memory Vision Transformer (LMViT); Convolutional Neural Networks (CNN); metal surface defect detection; deep learning, computer vision; image classification; learnable memory; gradient clipping; label smoothing; t-SNE visualization

Nomenclature

LMViT	Learnable Memory Vision Transformer
ViT	Vision Transformer
CNN	Convolutional Neural Network
t-SNE	t-Distributed Stochastic Neighbor Embedding
Gradient Clipping	Technique to prevent exploding gradients by limiting gradient values during backpropagation
Label Smoothing	Regularization technique that prevents the model from being overconfident by softening the labels

1 Introduction

Object recognition, semantic segmentation, and image classification are fundamental tasks in computer vision, with deep learning driving significant advancements in these areas. The evolution of complex neural network architectures has markedly improved the speed and accuracy of these tasks. Among these advancements, Vision Transformers (ViT) have emerged as a notable breakthrough due to their innovative self-attention mechanisms, which excel at capturing long-range interactions in images; something traditional CNNs often struggle with.

Recent developments have positioned the transformer architecture as a powerful neural network paradigm, leveraging self-attention mechanisms to extract essential features from data [1–3]. Initially designed for natural language processing (NLP), where it has set new benchmarks [3–5], the transformer has demonstrated significant potential in a range of AI applications. Self-attention-based architectures, particularly transformers [3], have become the gold standard in NLP, benefiting from pre-training on large texts and fine-tuning on specific tasks [4]. In the context of defect detection, CNNs primarily excel in local feature extraction but struggle with long-range dependencies, which are critical for accurately identifying subtle defects. In contrast, ViTs utilize self-attention mechanisms to capture these long-range dependencies and contextual information effectively, making them particularly suitable for surface defect detection.

In computer vision, the application of transformers has shown promising results. Chen et al. [6] illustrated that sequence transformers could achieve competitive results with CNNs on image classification tasks by predicting pixels autoregressively. The ViT model, introduced by Dosovitskiy et al. [7], applies the transformer model directly to computer vision by treating images as sequences of patches, demonstrating state-of-the-art performance on various image recognition benchmarks. This capability to model complex relationships within the data addresses the limitations of CNNs in accurately detecting steel surface defects.

Recent advancements in fine-tuning have also introduced techniques such as learnable memory mechanisms to enhance their performance on specific tasks [8]. Incorporating learnable memory modules allows for more effective adaptation of ViTs to task-specific data, further improving their utility in practical applications. In addition to Learnable Memory Vision Transformers (LMViT), other models such as Token to Token ViT, ViT without memory, and Parallel ViT have shown promise in computer vision tasks and are included in this study as comparative models.

This study proposes an LMViT model that outperforms traditional CNNs in steel surface defect detection, achieving 97.22% accuracy compared to 88.90% for ResNet50. Additionally, it provides a comparative analysis of the Token to Token ViT, ViT without memory, and Parallel ViT models alongside conventional CNN architectures, specifically ResNet18 and ResNet50, for surface defect detection, which is a critical application in industrial settings where precision is crucial.

Key aspects of this study include:

- **Dataset:** We utilized the NEU surface defect detection dataset [9], which is commonly used for evaluating models on metal surface flaw detection tasks. This dataset contains six types of defects: rolled-in scale, patches, crazing, pitted surface, inclusion, and scratches. Further details about the dataset are described in Section 3.1, where we explain the dataset characteristics, preprocessing steps, and augmentation techniques employed.
- **Methodology:** Use of data augmentation, customized dataset adjustments, and t-SNE visualization for feature analysis.
- **Fine-Tuning:** Application of advanced techniques such as gradient clipping, label smoothing, and learnable memory mechanisms [8].
- **Learnable ViT:** The ViT model utilized in this research incorporates learnable memory mechanisms, as introduced in [8]. This advanced approach enhances the ViT's ability to adapt to specific tasks by integrating a memory module that facilitates more effective fine-tuning. By leveraging these learnable memory mechanisms, the ViT model captures long-range dependencies and contextual information, improving its adaptability to task-specific data and leading to superior performance in image classification and surface defect detection.

The study's findings reveal that the LMViT model achieved a test accuracy of **97.22%**, outperforming ResNet18 (**88.89%**), ResNet50 (**88.90%**), Token to Token ViT (88.46%), ViT without memory (87.19%), and Parallel ViT (91.03%). The LMViT also demonstrated superior performance in training and validation accuracy, with a validation accuracy of **98.2%** compared to ResNet18's **91.0%**, ResNet50's **96.0%**, and the results of the other ViT models (89.12%, 87.51%, and 91.21%, respectively). This research contributes to the ongoing discussion about the effectiveness of these models in practical computer vision tasks, highlighting their advantages and limitations compared to conventional CNNs and other transformer-based models in industrial surface defect detection.

This paper has been formulated as follows: The authors discuss the literature review in Section 2. Then, Section 3 introduces the architecture of the model with a proper explanation of each subsection. Additionally, the authors discuss the dataset in Section 4, the experimental setup, and the result analysis in Section 5. Finally, the authors conclude the paper with limitations and future scope in Section 6.

2 Related Works

Steel defect detection is a critical area of focus in manufacturing, where ensuring product quality is paramount. Traditional methods, often based on manual inspection or conventional image processing techniques, have limitations in terms of accuracy and efficiency [10,11]. Recent advancements in deep learning have led to the development of automated systems that address these shortcomings [12,13]. CNNs have been widely adopted for this purpose [14–16]. For example, Bouguettaya et al. [16] demonstrated that CNNs could effectively classify surface defects in steel, highlighting their potential in industrial applications. Similarly, another research conducted by Yang et al. [17] focuses on lightweight architectures and proposes a model based on YOLOV5, MobilenetV2, and convolutional block attention module that achieves high accuracy in detecting various types of steel defects.

Despite their success, CNNs have certain limitations, particularly in capturing long-range dependencies within images [18–20]. This has led to exploring alternative architectures, such as ViTs, which have shown promise in overcoming these challenges [21]. Authors Vaswani et al. [3] introduced ViTs as a powerful tool for image classification tasks, and subsequent studies, such as those by Luo et al. [22], Zhou et al. [23], and Wang et al. [24], have explored their application in steel defect detection. Furthermore, different variations of ViTs have been used for steel defect detection. For instance, a study conducted by Üzen et al. [25] proposes a Swin Transformer-based model called Multi-Feature Integration Network that uses a pre-trained InceptionNet as a feature extractor of the encoder block and a Swin Transformer as a feature extractor of the decoder block. The proposed model has obtained a mean Intersection over Union (IoU) score of 81.37%, surpassing several CNN alternatives. Another study that suggests a different variation of the Swin Transformer, called the LSwin Transformer, was conducted by Zhu et al. [26]. It incorporates an effective window shift strategy and a depth multilayer perceptron module, achieving an average precision score of 81.2%. Moreover, a study by authors Feng et al. [27] proposes a hybrid ViT model that has the fully connected layers of VGGNet and achieves 5.64% higher accuracy than the original ViT. These studies suggest that ViTs can offer a more nuanced understanding of defect patterns by capturing local and global features in the images.

Furthermore, Qiu et al. [28] presented an enhanced YOLOv5 model incorporating deformable convolution, a bidirectional feature pyramid network, and Wise IoU loss function alongside optimized anchor box parameters via K-means clustering. This approach significantly improved detection accuracy to 78.1% on the NEU-DET dataset, surpassing the original YOLOv5 by 8.8% and the Faster R-CNN by 6.8%, with a minimal increase in inference time. However, the model's performance is still constrained by the complexity of real-world steel surface defects, indicating a need for further refinement to handle diverse and intricate defect patterns effectively. Another novel research by Yi et al. [29] proposes an improved YOLOX model for detecting defects on hot-rolled steel strips, addressing existing methods' low efficiency and accuracy issues. Enhancements include the MobileViT block for better feature extraction, Efficient Channel Attention (ECA) modules for emphasizing essential features and the efficient IOU loss function for improved accuracy. The model achieved 80.67 mean average precision on the NEU-DET dataset and 75.69 on the Xsteel dataset, showing significant improvements over the original YOLOX model.

As deep learning models like ViTs become more complex, the need for interpretability has grown. XAI techniques are increasingly used to make these models more transparent [30–32]. For instance, a study by Bordekar et al. [33] utilized Support Vector Machine and Density-Based Spatial Clustering to classify and cluster defects from CT images, achieving an impressive area under the curve score of 0.94. However, the method's reliance on high-quality photos could be improved, particularly in noisy data environments. Another notable study by Arreche et al. [34] presents a framework that integrates XAI with Intrusion Detection Systems (IDS). The authors aim to improve the transparency and interpretability of IDS models, making it easier for security professionals to understand and trust the system's decisions. Using XAI techniques, the framework provides explanations for detected intrusions, enhancing the effectiveness and reliability of network security measures.

Integrating XAI into steel defect detection has recently garnered significant interest among researchers. A comprehensive study by Aboulhosn et al. [35] explores the application of SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) for enhancing the interpretability of deep learning models used in steel surface defect detection. The results indicate that SHAP and LIME effectively highlight the features contributing to defect detection, with SHAP providing more consistent global explanations and LIME offering localized, instance-specific insights. Despite their effectiveness, the methods face computational complexity and scalability challenges. Future research directions include optimizing these techniques for real-time applications, developing hybrid approaches to combine global and local interpretability, and improving scalability and user interfaces for industrial applications. While previous studies have made notable progress in steel defect detection, most have focused on CNN architectures or standard ViT models. Our work introduces a novel approach by incorporating learnable memory mechanisms into the ViT architecture, specifically designed for steel defect detection. This enhancement overcomes the challenge of capturing long-range dependencies in images, which is critical for identifying subtle defect patterns across extensive surface areas. In another notable study, the authors Yi et al. [36] utilize saliency maps to enhance the defect detection process. Saliency maps highlight the most critical regions of the steel strip images, allowing the CNN to focus on these areas for more accurate defect classification. This approach improves the interpretability of the model by visually indicating which parts of the image contribute most to the detection of defects. The use of saliency maps, combined with CNNs, results in a robust and efficient system for identifying various types of steel strip defects, ultimately contributing to better quality control in steel production. Table 1 summarizes some of the works on steel defect detection.

Table 1: Comparison of studies on steel defect detection approaches

Study	Approach	Limitations
[10,11]	Traditional methods based on manual inspection and conventional image processing techniques for steel defect detection.	Limited accuracy and efficiency due to human errors and inability to handle complex defect patterns.
[12,13]	Developed automated steel defect detection systems using deep learning techniques.	Dependence on large datasets and high computational resources.
[14–16]	Used CNNs for classifying surface defects in steel, demonstrating their potential for industrial applications.	Inability to capture long-range dependencies within images; may not generalize well to diverse defect types.
[17]	Proposed a lightweight architecture combining YOLOv5, MobileNetV2, and Convolutional Block Attention Module for defect detection.	High accuracy achieved but struggles with long-range dependencies and computational efficiency.
[3,21]	Introduced ViTs as an alternative architecture to capture long-range dependencies in images for defect detection.	Requires extensive training data and computational power; potential overfitting on small datasets.
[22–24]	Applied ViTs in steel defect detection to capture both local and global features more effectively than CNNs.	High computational complexity and demands on resources.
[25]	Developed a Swin Transformer-based model (Multi-Feature Integration Network) with InceptionNet and Swin Transformer for encoding/decoding.	Computationally intensive, with limited interpretability.

(Continued)

Table 1 (continued)

Study	Approach	Limitations
[26]	Proposed LSwin Transformer with a window shift strategy and Depth Multilayer Perceptron Module for improved precision.	High computational complexity and moderate scalability issues.
[27]	Introduced a hybrid ViT model incorporating fully connected layers of VGGNet to enhance accuracy over the original ViT.	Higher model complexity and computational cost.
[28]	Enhanced YOLOv5 with deformable convolution, bidirectional feature pyramid network, and optimized anchor box parameters.	Still constrained by real-world complexity of diverse and intricate defect patterns.
[29]	Proposed an improved YOLOX model with MobileViT block, ECA module, and efficient IOU loss function for defect detection.	Potential limitations in generalizability and scalability to different datasets and real-world applications.
[33]	Used SVM and Density-Based Spatial Clustering for classifying and clustering defects in CT images.	Reliance on high-quality images; less effective in noisy data environments.
[34]	Integrated XAI with Intrusion Detection Systems (IDS) for enhanced transparency and interpretability.	Not specifically tailored for steel defect detection; focused on IDS applications.
[35]	Explored SHAP and LIME to improve the interpretability of deep learning models for steel surface defect detection.	Computational complexity and scalability challenges; requires optimization for real-time applications.
[36]	Employed saliency maps to enhance the defect detection process by highlighting critical image regions.	Limited generalization to diverse types of defects and challenges with real-world application scenarios.

3 Proposed Methodology

Our study builds on the Vision Transformer (ViT) by introducing a learnable memory module to enhance surface defect detection. The ViT model, as introduced in [7], divides an input image $x \in \mathbb{R}^{h \times w \times c}$ into fixed-size patches, linearly embeds these patches, adds position embeddings, and processes the resulting sequence with a standard Transformer encoder.

The input to the ViT model is defined as:

$$z_{\text{vit}}^{0} := [x_{\text{cls}}, Ex_1, \ldots, Ex_N] + E_{\text{pos}} \tag{1}$$

where E is a learnable linear transformation into the encoder's embedding space, E_{pos} represents the positional embeddings, and x_{cls} is a special learnable token for classification.

To introduce memory into the ViT model, we concatenate m learnable embeddings $E_{\text{mem}} \in \mathbb{R}^{m\times D}$, where D is the dimensionality of the input tokens:

$$z^0_{\text{mem}} := [z^0_{\text{vit}}; E^0_{\text{mem}}] \tag{2}$$

The transformer then processes $N + 1 + m$ tokens, with only the original $N + 1$ tokens retained after the self-attention layer. This process is repeated across layers, with each layer output y_l acting as input for the next:

$$z^l_{\text{mem}} := [y_{l-1}; E^l_{\text{mem}}] \tag{3}$$

We modify the standard ViT by introducing learnable memory tokens E_{mem}. These tokens allow the model to store and update task-specific information throughout the network, enhancing its ability to capture complex defect patterns.

During fine-tuning, the memory tokens are randomly initialized, and gradient descent is used to learn the memory tokens, classifier head, and class token x_{cls}. We also implement attention masking to maintain model performance on the original task while fine-tuning for new tasks, enabling multi-task learning.

This method offers flexibility for multi-task learning by fine-tuning memory tokens and class tokens for different tasks without task interference.

Fig. 1 illustrates the proposed model architecture.

3.1 Dataset

This research utilizes the NEU-DET [9], a widely used dataset for detecting and classifying surface defects in steel materials. The NEU-DET dataset includes 1800 images distributed equally among six prevalent types of surface defects. Each defect category contains 300 images with a 200 × 200 pixels resolution. These images were captured under controlled conditions to focus on typical defect patterns in steel materials. The NEU-DET dataset includes images of hot-rolled steel strips with six types of defects:

- **Crazing:** Fine cracks appearing on the surface.
- **Inclusion:** Non-metallic particles embedded in the metal surface.
- **Patches:** Surface areas with an irregular texture.
- **Pitted Surface:** Small depressions or cavities on the surface.
- **Rolled-in Scale:** Oxide layers or scale rolled into the surface during milling.
- **Scratches:** Linear marks caused by abrasion.

The defect types are illustrated in Fig. 2.

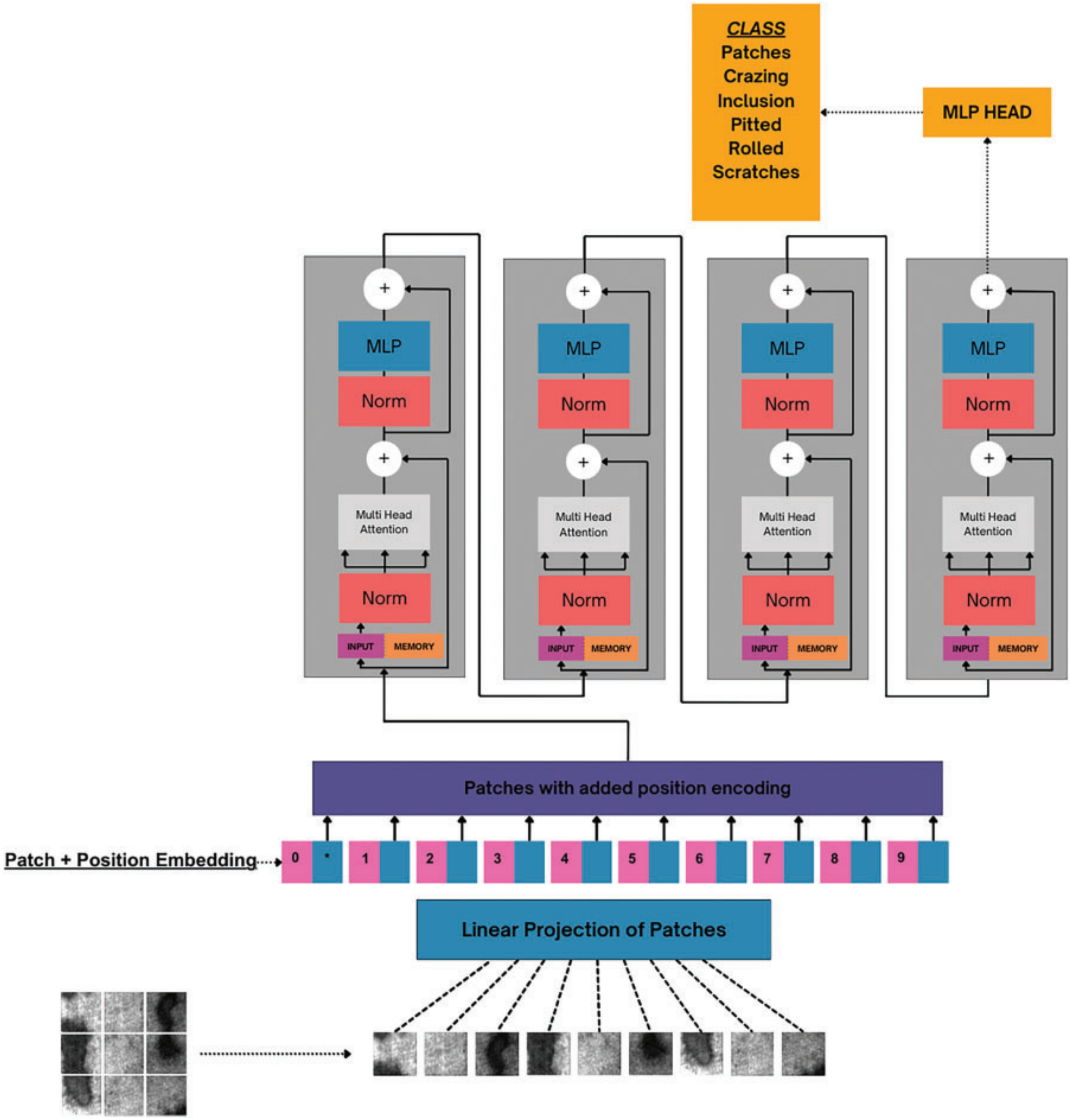


Figure 1: Visualization of learnable ViT

We divided the dataset into three subsets: training, validation, and testing, to maintain an equal representation of all defect classes and ensure class balance throughout the model development phases. The training set comprises 92% of the total data, while the validation and testing sets each account for 4%. Each subset includes images from all six defect types: Crazing, Inclusion, Patches, Pitted Surface, Rolled-in Scale, and Scratches. Specifically, the training set contains 276 images per defect type, totaling 1656. Both the validation and testing sets contain 12 images per defect type, resulting in 72 images each. This distribution was intended to provide adequate samples per class for the model to learn from, validate, and test.

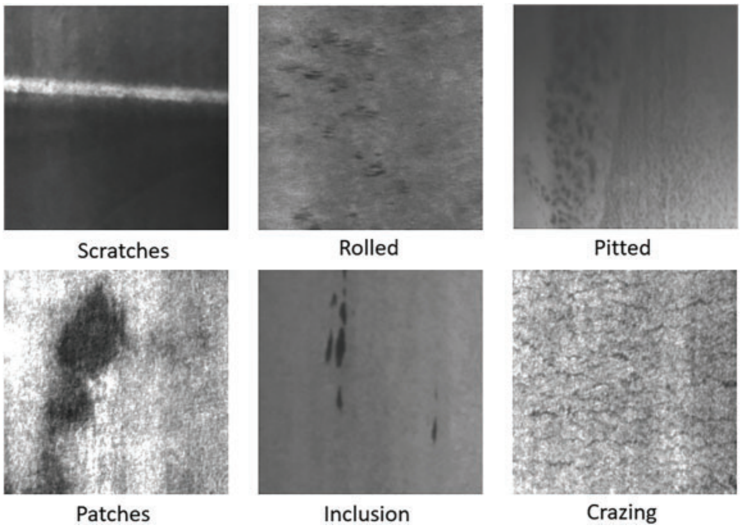


Figure 2: Types of defects

While the NEU-DET dataset is comprehensive, it is limited to six defect types and may not represent all possible steel surface defects. Additionally, all images are of the same size (200 × 200 pixels), which may not reflect the variety of scales encountered in real-world inspections. These limitations could affect the model's generalizability to other defect types or image scales. Models trained on this dataset may require further fine-tuning or adaptation when applied to datasets with more varied defect types and image resolutions.

3.2 Data Preprocessing and Augmentation

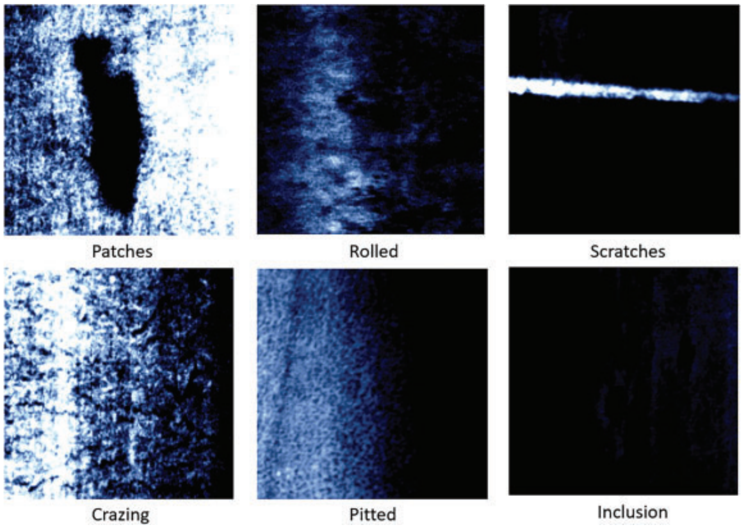
We applied a series of preprocessing and augmentation techniques using the PyTorch library to enhance the model's ability to generalize to unseen data. The transformations applied to the training set include:

1. **Resizing:** All images were resized to **224 × 224 pixels** to meet the input requirements of our deep learning model.
2. **Color Jittering:** A random adjustment of brightness and contrast, each with a magnitude of up to **0.08**, was applied to simulate variations in lighting conditions.
3. **Gaussian Blur:** A Gaussian blur with a kernel size of **3** was added to introduce noise and simulate surface irregularities.
4. **Random Rotation:** Each image was randomly rotated within a range of −**15 to +15 degrees** to ensure the model's robustness to orientation changes.
5. **Normalization:** The images were normalized using their mean and standard deviation values calculated for each channel. This standardization step ensures that the pixel values are centered around zero and scaled to a consistent range, which facilitates effective training of the deep learning model.

For testing, the images were transformed as follows:

- **Resizing:** Images were resized to **224 × 224 pixels**.
- **Normalization:** The images were normalized using the previously calculated mean and standard deviation values.

A denormalization function was employed during the visualization of images to convert them back to their original form during prediction and visualization. Fig. 3 illustrates the dataset after the preprocessing phase is completed.


Patches

Rolled

Scratches

Crazing

Pitted

Inclusion

Figure 3: Preprocessed dataset

3.3 t-SNE

In this study, we employ t-SNE to visualize high-dimensional representations of the dataset. t-SNE, a nonlinear dimensionality reduction technique developed by van der Maaten et al. [37], effectively embeds high-dimensional data into a lower-dimensional space (typically two or three dimensions), preserving local data structures to facilitate human interpretation.

We chose t-SNE because it effectively captures the complex relationships inherent in the data used for steel defect detection. Unlike traditional linear dimensionality reduction techniques such as Principal Component Analysis (PCA) [38], t-SNE is designed to focus on preserving the local structure of the data, making it particularly effective for visualizing clusters or groupings in data that are not linearly separable.

In steel defect detection, where different defects might share subtle similarities or differences, t-SNE allows for more precise separation of these types, thus aiding in more intuitive data analysis and better model understanding [39]. Moreover, t-SNE provides an intuitive visual tool that can help evaluate different models' performance, compare their outputs, and understand the underlying features that contribute most significantly to model decisions [40,41].

We have moved the detailed derivations of t-SNE's functions to the supplementary material, providing a concise explanation here. t-SNE begins by calculating a conditional probability $p_{j|i}$ that represents the similarity of data points in high-dimensional space. For each point x_i, a Gaussian distribution centered on x_i is computed, yielding a conditional probability:

$$p_{j|i} = \frac{exp(-\|x_i - x_j\|^2/2\sigma^2)}{\sum_{k\neq i} exp(-\|x_i - x_k\|^2/2\sigma^2)} \tag{4}$$

where σ is the bandwidth of the Gaussian distribution. The conditional probability $p_{j|i}$ measures the similarity of data point x_j to data point x_i.

Next, t-SNE constructs a joint probability distribution p_{ij} over the dataset by symmetrizing the conditional probabilities:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{5}$$

where N is the total number of data points.

In the lower-dimensional space, t-SNE aims to create a corresponding joint probability distribution q_{ij} based on a Student's t-distribution with one degree of freedom (Cauchy distribution):

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_i - y_k\|^2)^{-1}} \tag{6}$$

The t-SNE algorithm iteratively optimizes the Kullback-Leibler divergence between the two distributions p and q:

$$C = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \tag{7}$$

Minimizing this divergence ensures that the low-dimensional map accurately reflects the similarity relationships of the original high-dimensional data. The lower the KL divergence, the better the preservation of local data structures in the low-dimensional embedding.

To achieve this minimization, t-SNE employs gradient descent, an iterative optimization method that continuously adjusts the positions of data points in the low-dimensional space to reduce the divergence. This optimization is sensitive to several hyperparameters, including the perplexity, which controls the balance between local and global aspects of the data, and the learning rate, which affects the convergence speed and stability of the algorithm.

4 Experiments

This section presents our experiments comparing our LMViT, Token to Token ViT, ViT without memory, and Parallel ViT models to conventional CNNs like ResNet18 and ResNet50 for surface defect detection. The experiments evaluated the models' performance using confusion matrices, loss patterns, test accuracy, and training and validation accuracy. The models were trained for 100 epochs using a batch size of 16. We utilized the AdamW optimizer with a learning rate of 0.001, with adjustments for different ViT-based models to ensure optimal training. The learning rate was adjusted dynamically using the ReduceLROnPlateau scheduler to ensure optimal convergence. Gradient clipping was applied with a maximum norm of 1 to prevent gradient explosion.

The models were evaluated based on key metrics, including accuracy, precision, recall, and F1-score, ensuring a comprehensive assessment of their performance throughout the training duration of 100 epochs. The experiments were conducted on a system equipped with an NVIDIA TITAN X GPU and 64 GB of DDR4 RAM, running Ubuntu 20.04 LTS as the operating system. All models were implemented using PyTorch version 1.10.0 and trained on the above hardware configuration. The performance of the models was validated against a held-out test set to ensure the consistency and reliability of the results. We observed that LMViT achieved the highest test accuracy and F1-score, while the other ViT-based models also demonstrated improvements over ResNet18 and ResNet50.

4.1 Evaluation Metrics

Multiple evaluation metrics assess the proposed architecture's performance. Eqs. (8)–(11) present the calculations for accuracy, recall, precision, and F1-score. Additionally, the architecture's effectiveness is examined using a confusion matrix. In these equations, TP (true positive), TN (true negative), FP (false positive), and FN (false negative) represent correctly and incorrectly classified instances of positive and negative images.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$\text{F1-score} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{11}$$

4.2 Result Analysis with t-SNE

We monitored the training and validation accuracy of the models over 100 epochs. The LMViT, Token to Token ViT, ViT without memory, and Parallel ViT models showed a steady increase in validation accuracy, as shown in Fig. 4, eventually surpassing both ResNet18 and ResNet50. In contrast, the ResNet models exhibited clear signs of overfitting during the later epochs, with validation accuracy decreasing significantly or plateauing.

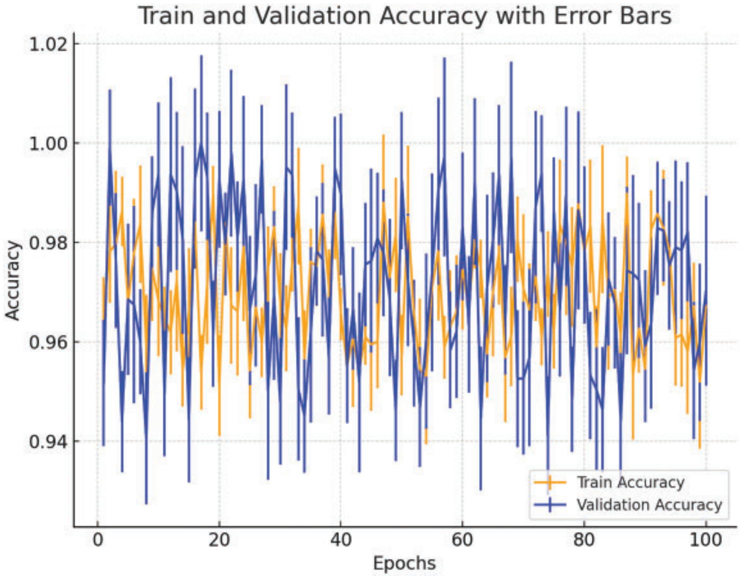


Figure 4: Training and validation accuracy

Table 2 compares the final training and validation accuracies of the models after 100 epochs.

Table 2: Training and validation accuracy

Model	Training accuracy (%)	Validation accuracy (%)
LMViT	99.4	98.2
Token to Token ViT	88.4	89.12
ViT without memory	87.18	87.51
Parallel ViT	91.03	91.21
ResNet18	85.0	91.0
ResNet50	89.0	96.0

Following the training phase, each model was assessed on the test set. The LMViT model demonstrated a test accuracy of 97.22%, surpassing both ResNet18 (88.89%) and ResNet50 (88.90%), as well as showing an improvement over the other ViT-based models. The test accuracies are summarized in Table 3.

Table 3: Test accuracy comparison

Model	Test accuracy (%)
LMViT	97.22
Token to Token ViT	88.46
ViT without memory	87.18
Parallel ViT	91.03
ResNet18	88.89
ResNet50	88.90

Confusion matrices were generated to understand the classification performance across different defect categories. The confusion matrix for the LMViT model is shown in Fig. 5, depicting fewer misclassifications compared to the CNN-based models.

The LMViT model showed particular improvement in distinguishing between ‘Rolled-in Scale’ and ‘Patches’ defects, which were often confused by the CNN models. However, it still occasionally misclassified ‘Crazing’ as ‘Scratches’, suggesting room for improvement in detecting fine-grained linear defects.

Tables 4–6 summarize the confusion matrix metrics for Token to Token ViT, ViT without memory, and Parallel ViT, respectively, providing precision and recall comparisons with LMViT and the CNN models. Table 7 below shows the different matrices with different ablation.

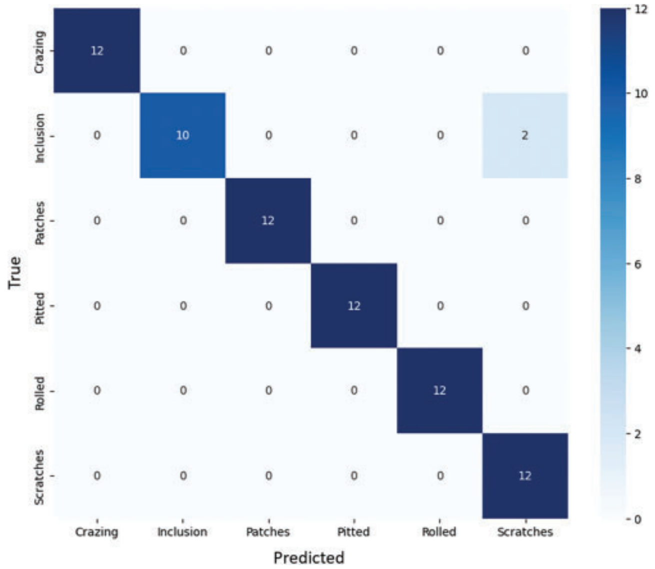


Figure 5: Confusion matrix for LMViT model

Table 4: Confusion matrix metrics—Token to Token ViT

Model	Precision (%)	Recall (%)
Token to Token ViT	89.12	88.14

Table 5: Confusion matrix metrics—ViT without memory

Model	Precision (%)	Recall (%)
ViT without memory	87.51	87.20

Table 6: Confusion matrix metrics—Parallel ViT

Model	Precision (%)	Recall (%)
Parallel ViT	91.21	90.49

Table 7: Evaluation performance with accuracy split ratio ablation

Model	Epochs	Batch size	Splitting ratio	Accuracy
Token to Token ViT	50	16	70:30	85.54
Token to Token ViT	100	16	80:20	88.46
ViT without memory	50	16	70:30	85.45
ViT without memory	100	16	80:20	87.18

(Continued)

Table 7 (continued)

Model	Epochs	Batch size	Splitting ratio	Accuracy
Parallel ViT	50	16	70:30	90.02
Parallel ViT	100	16	80:20	91.03
ResNet18	50	16	70:30	88.65
ResNet18	100	16	80:20	91.0
ResNet50	50	16	70:30	95.22
ResNet50	100	16	80:20	96.0
LMViT	50	16	70:30	89.22
LMViT	100	16	80:20	97.22

t-SNE visualizations were used to examine the feature representations learned by each model over multiple epochs. The ViT-based models produced more distinct clusters for each defect class compared to CNNs, indicating improved feature space separation. Fig. 6 illustrates t-SNE visualizations for the LMViT model at different epochs, showing the progressive refinement in clustering defect classes over time.

To further evaluate the performance of our models, randomly selected images from the test set were fed into each model to predict their corresponding labels. Fig. 7 shows a visual comparison of actual *vs*. predicted labels, providing insight into cases where the models performed correctly or encountered misclassifications.

4.3 Ablation Study on Model Components

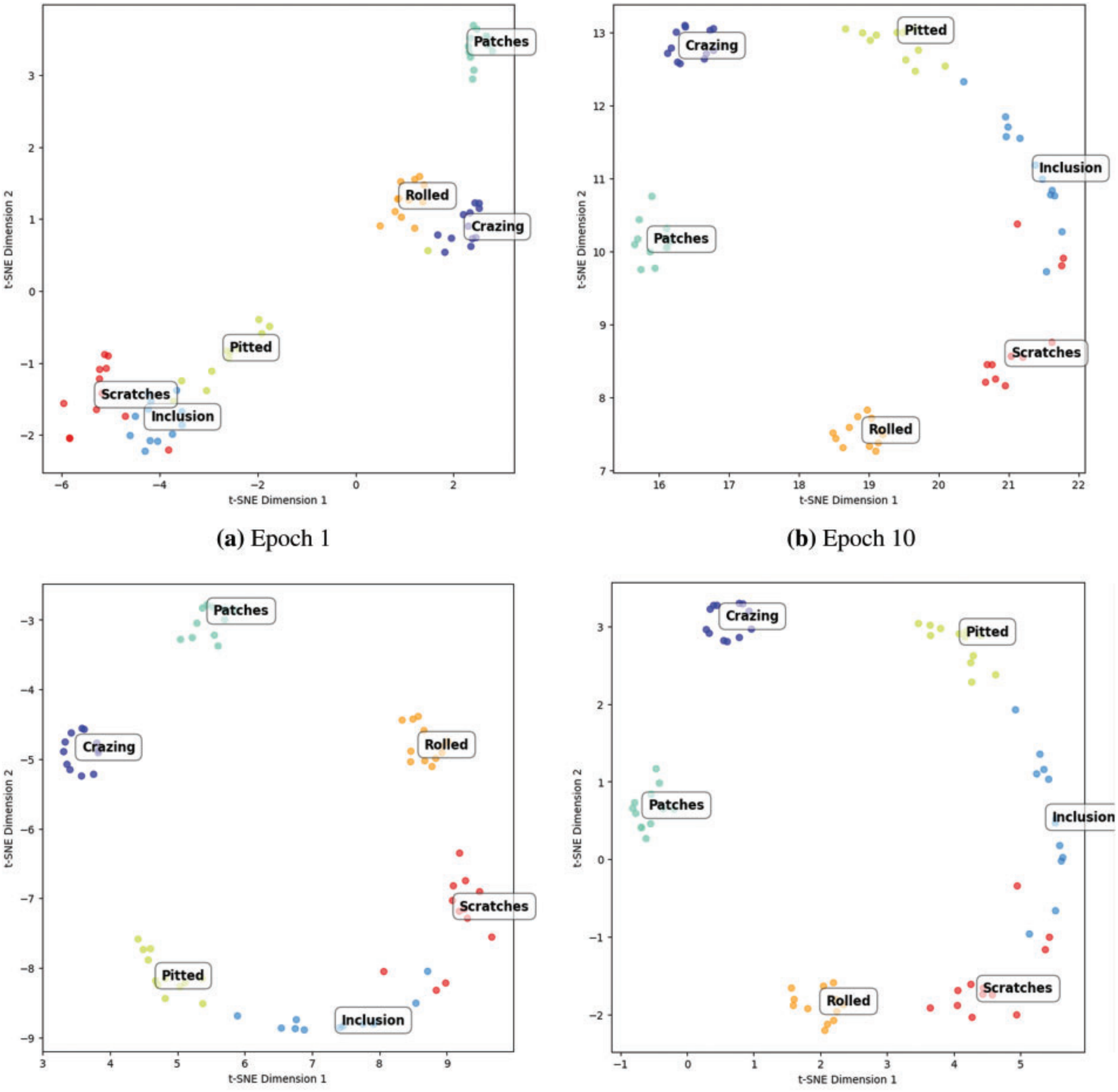
We also conducted additional experiments to analyze the impact of the learnable memory mechanism and other key modules. Specifically, we investigated how variations in memory configurations and architectural designs affect the validation accuracy of our proposed model.

Fig. 8 illustrates a comparative analysis of different model configurations, including baseline models (ResNet18, ResNet50, and variations of ViT without memory) and advanced configurations incorporating the learnable memory mechanism. The LMViT, our proposed lightweight design, achieves the highest validation accuracy of **97.22%**, outperforming standard Vision Transformers (ViTs) and other variants.

Key findings from this analysis are as follows:

- **Effect of Learnable Memory:** Models incorporating learnable memory (e.g., LMViT) demonstrate a significant improvement in accuracy compared to counterparts without memory tokens (e.g., ViT without Memory and Reduced Memory Tokens).
- **Lightweight Design:** Despite its reduced complexity, LMViT achieves superior performance, confirming its effectiveness in extracting meaningful representations while maintaining a lightweight architecture.
- **Comparative Performance:** LMViT surpasses state-of-the-art architectures like ResNet50 and Standard ViT, emphasizing the importance of the proposed memory mechanism and design optimizations.

These results provide a comprehensive ablation study, highlighting the critical role of the learnable memory mechanism and supporting the overall contributions of the proposed model.



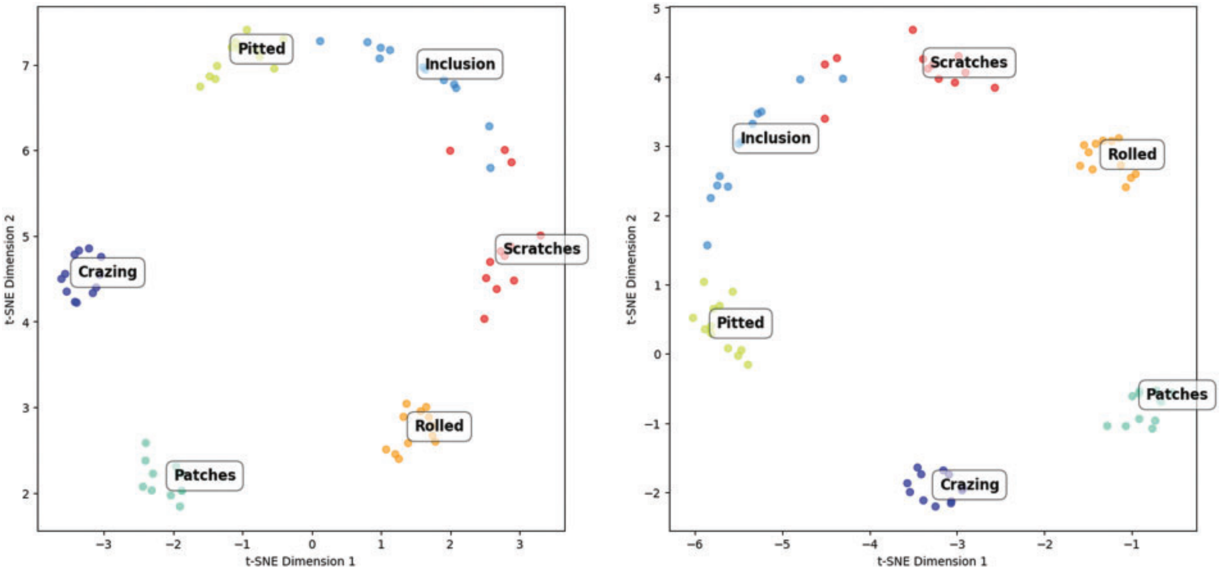
(a) Epoch 1

(b) Epoch 10

(c) Epoch 20

(d) Epoch 50

Figure 6: (Continued)



(e) Epoch 90

(f) Epoch 100

Figure 6: t-SNE visualization of feature representations at various epochs

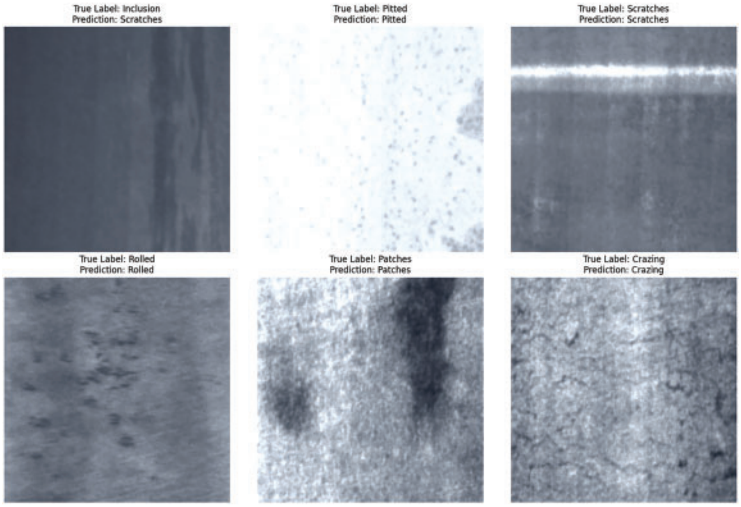


Figure 7: Visualization of predicted and true labels for sample images from the test set. Green boxes indicate correct predictions, while red boxes show misclassifications

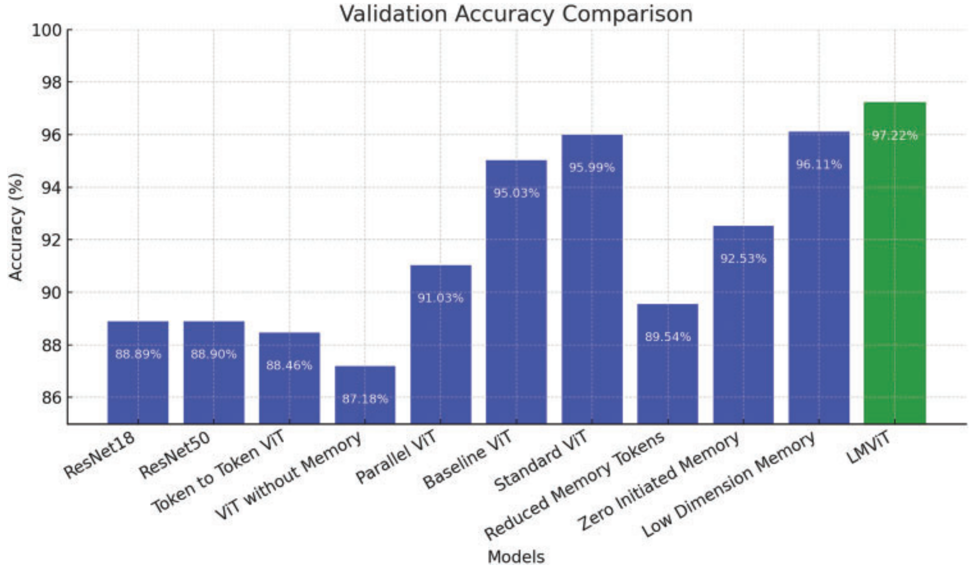


Figure 8: Validation accuracy comparison across various model configurations

4.4 Computational Efficiency

While the LMViT model achieved higher accuracy, it required 1.5× the training time of ResNet50 on the same hardware. However, inference time was comparable, with LMViT processing 100 images in 2.3 s compared to ResNet50's 2.1 s.

5 Discussion

The results of this study highlight the impressive capabilities of the LMViT model in comparison to traditional CNNs. The LMViTmodel achieved 8.33% and 8.32% higher test accuracies than ResNet18 and ResNet50, respectively. Moreover, it outperformed these CNNs in validation accuracy, achieving 7.2% higher than ResNet18 and 2.2% higher than ResNet50. These findings underscore the effectiveness of transformer-based models in complex tasks requiring nuanced understanding and contextual awareness.

In addition to LMViT, this study also examined the performance of three other transformer-based models: Token-to-Token ViT, ViT without memory, and Parallel ViT. Preliminary results indicate that these models demonstrated competitive performance, with Token-to-Token ViT achieving a test accuracy of 88.46%, ViT without memory achieving 87.18%, and Parallel ViT achieving 91.03%. While each model displayed distinct strengths, LMViT generally outperformed the other transformer-based models, showcasing the benefit of a learnable memory mechanism in this context.

The advantages of transformer-based models, particularly LMViT, over traditional CNN architectures suggest that transformer-based models possess significant potential to advance image analysis applications, especially in industrial settings where precision is crucial. For instance, the 8.32% accuracy improvement of the highest-performing transformer model, LMViT, over ResNet50 could significantly reduce false positives in a production environment. For a steel mill producing 500,000 tons annually, this could potentially prevent misclassification of 41,600 tons of product, saving considerable reprocessing costs and improving overall product quality.

However, while the results are promising, deploying transformer-based models like LMViT, Token-to-Token ViT, ViT without memory, and Parallel ViT in real-world settings may face challenges such as integrating with existing inspection systems, handling variable lighting conditions, and processing high-speed production lines. Further research is needed to assess the models' performance under these dynamic conditions.

Overall, the results of this study are inspiring and open up new possibilities in the field, emphasizing the need for continued exploration of transformer-based models in diverse applications and environments.

6 Conclusion

Looking forward, several avenues for future research are suggested. One potential direction is exploring hybrid architectures that combine the strengths of CNNs and transformers to balance computational costs and accuracy. Additionally, applying the ViT model, along with Token-to-Token ViT, ViT without memory, and Parallel ViT, to a broader range of industrial tasks could test their versatility and robustness. Future research should investigate the LMViT's performance on a wider range of steel defects, including less common types not present in the NEU-DET dataset. Improving data efficiency through self-supervised learning or advanced data augmentation techniques may reduce reliance on large labeled datasets. Optimizing ViT models, as well as the newly introduced Token-to-Token ViT, ViT without memory, and Parallel ViT, for deployment on edge devices with limited computational resources is another critical area, which could involve model pruning, quantization, and knowledge distillation. Scaling these approaches to larger datasets or real-time defect detection may require optimizing the model architectures for speed. Techniques such as model pruning and knowledge distillation could be explored to create more lightweight versions of these transformer models suitable for edge deployment in manufacturing environments. Enhancing the interpretability of ViT models, particularly by developing methods to visualize and understand their self-attention mechanisms, is also critical for fostering trust and adoption in safety-critical applications.

Despite these advances, the study has some limitations. The ViT model's significant computational requirements pose challenges for real-time or resource-constrained deployments. Its dependency on large labeled datasets can limit its utility in scenarios where data is scarce or expensive to obtain. Overfitting remains a risk, particularly with small or imbalanced datasets, necessitating careful regularization and hyperparameter tuning. Moreover, the complexity of self-attention mechanisms can reduce model interpretability, which is essential in industrial decision-making. Finally, while the study focused on surface defect detection, further validation is needed to generalize ViT's performance, as well as Token-to-Token ViT, ViT without memory, and Parallel ViT, across different industrial tasks and datasets. Addressing these limitations and pursuing the proposed research directions will be essential for further advancing the application and effectiveness of transformer-based models in various domains. The need for further research is urgent, and the potential benefits are significant.

Acknowledgement: We extend our gratitude to Northeastern University for their invaluable contribution by providing the dataset utilized in this research. The dataset is accessible at http://hdl.handle.net/2047/D20621538 (accessed on 16 December 2024).

Funding Statement: This research is funded by Woosong University Academic Research 2024.

Author Contributions: Syed Tasnimul Karim Ayon took responsibility for collecting data, conducting detailed research to define the study's scope, performing an in-depth literature review of the research

area, labeling and preparing the dataset, and contributing to the development of the research code. Farhan Md. Siraj was in charge of data processing and augmentation, designing the methodology, choosing the suitable codebase approach, selecting relevant evaluation metrics, setting up t-SNE for result visualization, and contributing to the development of the research code. Jia Uddin provided essential supervision throughout the project, ensuring alignment with the research objectives and goals. He thoroughly reviewed the manuscript, offering constructive feedback and suggestions to enhance its quality and clarity. Furthermore, Jia Uddin played a key role in securing the necessary funding for the research, ensuring the availability of resources essential for the research's success. The authors contributed equally to the research and the overall execution of the project. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The NEU-DET dataset used in this research is publicly available and can be accessed through the Kaggle platform. The 1800 photos in this dataset are divided into six main surface defect categories frequently seen in hot-rolled steel materials: crazing, inclusion, patches, pitted surfaces, rolled-in scale, and scratches. There are 300 photos, each with a 200 by 200 pixels resolution, for each fault category, which is evenly represented. The dataset can be accessed via the link: http://hdl.handle.net/2047/D20621538 (accessed on 16 December 2024).

Ethics Approval: This research does not involve human participants, animals, or sensitive data requiring formal ethical approval.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015, *arXiv:1409.0473*.

[2] A. Parikh, O. Tackstrom, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," 2016, *arXiv:1606.01933*.

[3] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Red Hook, NY: Curran Associates, Inc., 2017.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019, *arXiv:1810.04805*.

[5] T. B. Brown *et al.*, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.

[6] M. Chen *et al.*, "Generative pretraining from pixels," in *Proc. 37th Int. Conf. Mach. Learn.*, PMLR, 2020, vol. 119, pp. 1691–1703.

[7] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[8] M. Sandler, A. Zhmoginov, M. Vladymyrov, and A. Jackson, "Fine-tuning image transformers using learnable memory," 2024, *arXiv:2203.15243*.

[9] M. Zabin, A. N. B. Kabir, M. K. Kabir, H. -J. Choi, and J. Uddin, "Contrastive self-supervised representation learning framework for metal surface defect detection," *J. Big Data*, vol. 10, no. 1, 2023, Art. no. 145. doi: 10.1186/s40537-023-00827-z.

[10] Y. Chen, Y. Ding, F. Zhao, E. Zhang, Z. Wu and L. Shao, "Surface defect detection methods for industrial products: A review," *Appl. Sci.*, vol. 11, no. 16, 2021, Art. no. 7657. doi: 10.3390/app11167657.

[11] W. B. Rashid and S. Goel, "Advances in the surface defect machining (SDM) of hard steels," *J. Manuf. Process.*, vol. 23, no. 2, pp. 37–46, 2016. doi: 10.1016/j.jmapro.2016.05.007.

[12] X. Zheng, S. Zheng, Y. Kong, and J. Chen, "Recent advances in surface defect inspection of industrial products using deep learning techniques," *Int. J. Adv. Manuf. Technol.*, vol. 113, pp. 35–58, 2021. doi: 10.1007/s00170-021-06592-8.

[13] X. Sun, J. Gu, S. Tang, and J. Li, "Research progress of visual inspection technology of steel products-a review," *Appl. Sci.*, vol. 8, no. 11, 2018, Art. no. 2195. doi: 10.3390/app8112195.

[14] Q. Luo, X. Fang, L. Liu, C. Yang, and Y. Sun, "Automated visual defect detection for flat steel surface: A survey," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 626–644, 2020. doi: 10.1109/TIM.2019.2963555.

[15] S. Qi, J. Yang, and Z. Zhong, "A review on industrial surface defect detection based on deep learning technology," in *Proc. 2020 3rd Int. Conf. Mach. Learn. Mach. Intell.*, 2020, pp. 24–30.

[16] A. Bouguettaya and H. Zarzour, "CNN-based hot-rolled steel strip surface defects classification: A comparative study between different pre-trained CNN models," *Int. J. Adv. Manuf. Technol.*, vol. 132, no. 1, pp. 399–419, 2024. doi: 10.1007/s00170-024-13341-0.

[17] L. Yang, X. Huang, Y. Ren, and Y. Huang, "Steel plate surface defect detection based on dataset enhancement and lightweight convolution neural network," *Machines*, vol. 10, no. 7, 2022, Art. no. 523. doi: 10.3390/machines10070523.

[18] L. Alzubaidi *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, pp. 1–74, 2021. doi: 10.1186/s40537-021-00444-8.

[19] A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique and J. Henkel, "A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends," *IEEE Access*, vol. 12, no. 2, pp. 41180–41218, 2024. doi: 10.1109/ACCESS.2024.3376441.

[20] G. Rangel, J. C. Cuevas-Tello, J. Nunez-Varela, C. Puente, and A. G. Silva-Trujillo, "A survey on convolutional neural networks and their performance limitations in image recognition tasks," *J. Sens.*, vol. 2024, no. 1, 2024, Art. no. 2797320. doi: 10.1155/2024/2797320.

[21] K. Han *et al.*, "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, 2022. doi: 10.1109/TPAMI.2022.3152247.

[22] Q. Luo, J. Su, C. Yang, W. Gui, O. Silven and L. Liu, "CAT-EDNet: Cross-attention transformer-based encoder-decoder network for salient defect detection of strip steel surface," *IEEE Trans. Instrum. Meas.*, vol. 71, no. 2, pp. 1–13, 2022. doi: 10.1109/TIM.2022.3165270.

[23] H. Zhou, R. Yang, R. Hu, C. Shu, X. Tang and X. Li, "ETDNet: Efficient transformer-based detection network for surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–14, 2023. doi: 10.1109/TIM.2023.3307753.

[24] J. Wang, G. Xu, F. Yan, J. Wang, and Z. Wang, "Defect transformer: An efficient hybrid transformer architecture for surface defect detection," *Measurement*, vol. 211, no. 3, 2023, Art. no. 112614. doi: 10.1016/j.measurement.2023.112614.

[25] H. Üzen, M. Türkoğlu, B. Yanikoglu, and D. Hanbay, "Swin-MFINet: Swin transformer based multi-feature integration network for detection of pixel-level surface defects," *Expert. Syst. Appl.*, vol. 209, 2022, Art. no. 118269. doi: 10.1016/j.eswa.2022.118269.

[26] W. Zhu, H. Zhang, C. Zhang, X. Zhu, Z. Guan and J. Jia, "Surface defect detection and classification of steel using an efficient swin transformer," *Adv. Eng. Inform.*, vol. 57, 2023, Art. no. 102061. doi: 10.1016/j.aei.2023.102061.

[27] X. Feng, X. Gao, and L. Luo, "An improved vision transformer-based method for classifying surface defects in hot-rolled strip steel," *J. Phys.: Conf. Ser.*, vol. 2082, no. 1, 2021, Art. no. 012016.

[28] K. Qiu and C. Wang, "Improved YOLOv5 based on the mobilevit backbone for the detection of steel surface defects," in *Proc. 2024 3rd Int. Conf. Cyber Secur., Artif. Intell. Digital Econ.*, 2024, pp. 305–309.

[29] C. Yi, B. Xu, J. Chen, Q. Chen, and L. Zhang, "An improved yolox model for detecting strip surface defects," *Steel Res. Int.*, vol. 93, no. 11, 2022, Art. no. 2200505. doi: 10.1002/srin.202200505.

[30] A. B. Arrieta *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020. doi: 10.1016/j.inffus.2019.12.012.

[31] D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen, "Explainable artificial intelligence: A comprehensive review," *Artif. Intell. Rev.*, pp. 1–66, 2022. doi: 10.1007/s10462-021-10088-y.

[32] T. Speith, B. Crook, S. Mann, A. Schomäcker, and M. Langer, "Conceptualizing understanding in explainable artificial intelligence (XAI): An abilities-based approach," *Ethics Inf. Technol.*, vol. 26, no. 2, p. 40, 2024. doi: 10.1007/s10676-024-09769-3.

[33] H. Bordekar, N. Cersullo, M. Brysch, J. Philipp, and C. Hühne, "Explainable artificial intelligence for automatic defect detection in additively manufactured parts using CT scan analysis," *J. Intell. Manuf.*, pp. 1–18, 2023. doi: 10.1007/s10845-023-02272-4.

[34] O. Arreche, T. Guntur, and M. Abdallah, "XAI-IDS: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems," *Appl. Sci.*, vol. 14, no. 10, 2024, Art. no. 4170. doi: 10.3390/app14104170.

[35] Z. Aboulhosn, A. Musamih, K. Salah, R. Jayaraman, M. Omar and Z. Aung, "Detection of manufacturing defects in steel using deep learning with explainable artificial intelligence," *IEEE Access*, vol. 12, pp. 99240–99257, 2024. doi: 10.1109/ACCESS.2024.3430113.

[36] L. Yi, G. Li, and M. Jiang, "An end-to-end steel strip surface defects recognition system based on convolutional neural networks," *Steel Res. Int.*, vol. 88, no. 2, 2017, Art. no. 1600068. doi: 10.1002/srin.201600068.

[37] L. Van der Maaten and G. Hinton, "Visualizing data using t-CNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[38] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Comput. Geosci.*, vol. 19, no. 3, pp. 303–342, 1993. doi: 10.1016/0098-3004(93)90090-R.

[39] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-CNE effectively," *Distill*, vol. 1, no. 10, 2016, Art. no. e2. doi: 10.23915/distill.00002.

[40] A. Gisbrecht and B. Hammer, "Data visualization by nonlinear dimensionality reduction," *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.*, vol. 5, no. 2, pp. 51–73, 2015. doi: 10.1002/widm.1147.

[41] S. Liu, D. Maljovec, B. Wang, P. -T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 3, pp. 1249–1268, 2016. doi: 10.1109/TVCG.2016.2640960.