Intelligent Fusion Framework for Predicting Defect Type and Localization in Steel Manufacturing Processes

Mahmoud M. Ismail[1*],Mahmoud M.Ibrahim[2] , Heba R. Abdelhady [3]

[1,2,3] Decision support department, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Sharqiyah, Egypt.

Emails: mmsabe@zu.edu.eg; mmsba@zu.edu.eg.

*Correspondence: mmsabe@zu.edu.eg.

Abstract

The defect prediction in the manufacturing of steel is a critical challenge because it affects the quality and safety of the products. For this reason, intelligent image fusion approach is introduced in this research to enhance accurate prediction of defect types and locations in steel materials. By utilizing U-Net architecture and pretrained ResNet18 encoder layers, our method performs fusion of data from several imaging modalities thus supporting precise localization as well as classification of defects. In our model's learning curves as well as comparing predicted segmentation masks with ground truth images, extensive experimentation and visualization show that our model captures subtle defects very well. By so doing, it exhibits robust performance that mitigates risks associated with overfitting since it can accurately identify any flaw while still having the ability to accept unseen data from other sources. These results suggest that our approach can highly contribute to improving quality control and safety standards for steel production.

Keywords: Information Fusion; Steel Manufacturing; Defect Recognition; Defect Localization.

1. Introduction

Steel manufacturing is the backbone of industry that has the most significant influence in construction, infrastructure development, and motor vehicles. However, there is a problem ensuring the best standards of quality in steel manufacture caused by defects that can happen throughout production [1-3]. On the other hand, these defects may cause a loss of integrity as well as a general decline in quality of steel products resulting into huge economic losses and concerns on safety. The conventional techniques used to detect defects in steel production have been limited in their ability to predict types or location of such defects within the material [4].

To address this challenge, this paper puts forward an innovative approach based on smart image fusion techniques for accurate defect detection during steel manufacturing [5]. The main objective of this study is to forecast and locate precisely defects in steel materials as well as identify their specific kinds. On the other hand, this research seeks to improve significantly accuracy and efficiency of defect identification within the domain of steel production using progress made through artificial intelligence (AI) and image processing [6-9].

The core purpose of this study is the development of smart image fusion approach which combines various imaging modalities or sources in order to predict locations and types of defects in steel production [10]. By merging data from different sources such as X-ray, ultrasonic, and infrared imaging, this methodology aims at surpassing the limitations of single imaging modes towards a more holistic defect detection system. Moreover, integrating machine

Doi : https://doi.org/10.54216/JCHCI.080201
Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024

learning algorithms will help to develop an autonomous and accurate defect classification system and thus improve the accuracy and reliability of defect identification [11-12].

Not only will this research offer an improved way to detect flaws but also it will come up with a stronger system that accurately predicts both the kind and area of fault in steel materials [13]. Through intelligent image fusion approaches, which enhance fault detection capability, and machine learning techniques, this study looks forward to cutting-down on production losses, improving quality of products, as well as enhancing safety measures in the steel manufacturing industry.

2. Methodology

This section delineates the step-by-step procedure followed to amalgamate multiple imaging modalities and harness their collective power in identifying defect types and localizing their positions within steel materials.

This work applies the U-Net architecture to serve as a cornerstone in our approach to precisely delineate defects within the steel manufacturing process. U-Net, renowned for its effectiveness in biomedical image segmentation, has gained widespread acclaim due to its distinctive architecture and design principles, making it an ideal candidate for our defect prediction framework. The fundamental architecture of U-Net consists of an encoder-decoder network, where the encoder extracts hierarchical features from the input image, while the decoder reconstructs the spatial information to generate high-resolution segmentation masks. The main components of U-Net encompass its contracting path (encoder) and expansive path (decoder), coupled with skip connections facilitating feature concatenation across mirrored layers, fostering precise localization of defects [14].

```python
import torch
import torch.nn as nn
import torchvision.models as models

# Helper function to create convolutional layers followed by ReLU activation
def convrelu(in_channels, out_channels, kernel, padding):
    return nn.Sequential(
        nn.Conv2d(in_channels, out_channels, kernel, padding=padding),
        nn.ReLU(inplace=True),
    )

# UNet model definition
class UNet(nn.Module):
    def __init__(self, n_class):
        super().__init__()

        # Loading the pre-trained ResNet18 model
        self.base_model = models.resnet18()
        # Loading the pre-trained weights (assuming the path is specified correctly)
        self.base_model.load_state_dict(torch.load("../input/resnet18/resnet18.pth"))
        self.base_layers = list(self.base_model.children())

        # Encoder layers
        self.layer0 = nn.Sequential(*self.base_layers[:3])
        self.layer0_1x1 = convrelu(64, 64, 1, 0)
        self.layer1 = nn.Sequential(*self.base_layers[3:5])
        self.layer1_1x1 = convrelu(64, 64, 1, 0)
        self.layer2 = self.base_layers[5]
        self.layer2_1x1 = convrelu(128, 128, 1, 0)
        self.layer3 = self.base_layers[6]
        self.layer3_1x1 = convrelu(256, 256, 1, 0)
        self.layer4 = self.base_layers[7]
        self.layer4_1x1 = convrelu(512, 512, 1, 0)
```


Doi : https://doi.org/10.54216/JCHCI.080201
Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024

```
        # Upsampling
        self.upsample = nn.Upsample(scale_factor=2, mode='bilinear', align_corners=True)

        # Decoder layers with skip connections
        self.conv_up3 = convrelu(256 + 512, 512, 3, 1)
        self.conv_up2 = convrelu(128 + 512, 256, 3, 1)
        self.conv_up1 = convrelu(64 + 256, 256, 3, 1)
        self.conv_up0 = convrelu(64 + 256, 128, 3, 1)

        # Layers for handling original size input
        self.conv_original_size0 = convrelu(3, 64, 3, 1)
        self.conv_original_size1 = convrelu(64, 64, 3, 1)
        self.conv_original_size2 = convrelu(64 + 128, 64, 3, 1)

        # Final convolutional layer
        self.conv_last = nn.Conv2d(64, n_class, 1)

    def forward(self, input):
        # Processing original input size
        x_original = self.conv_original_size0(input)
        x_original = self.conv_original_size1(x_original)

        # Encoder path
        layer0 = self.layer0(input)
        layer1 = self.layer1(layer0)
        layer2 = self.layer2(layer1)
        layer3 = self.layer3(layer2)
        layer4 = self.layer4(layer3)

        # Downsampling layers and skip connections
        layer4 = self.layer4_1x1(layer4)
        x = self.upsample(layer4)
        layer3 = self.layer3_1x1(layer3)
        x = torch.cat([x, layer3], dim=1)
        x = self.conv_up3(x)

        x = self.upsample(x)
        layer2 = self.layer2_1x1(layer2)
        x = torch.cat([x, layer2], dim=1)
        x = self.conv_up2(x)

        x = self.upsample(x)
        layer1 = self.layer1_1x1(layer1)
        x = torch.cat([x, layer1], dim=1)
        x = self.conv_up1(x)

        x = self.upsample(x)
        layer0 = self.layer0_1x1(layer0)
        x = torch.cat([x, layer0], dim=1)
        x = self.conv_up0(x)

        x = self.upsample(x)
        x = torch.cat([x, x_original], dim=1)
```


Doi : https://doi.org/10.54216/JCHCI.080201
Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024

```
        x = self.conv_original_size2(x)

        # Final output
        out = self.conv_last(x)

        return out
```

The design principles ingrained within U-Net contribute significantly to its effectiveness in segmentation tasks. The contracting path comprises a series of convolutional layers followed by max-pooling operations, enabling the extraction of high-level abstract features from the input images while reducing spatial dimensions [15]. Subsequently, the expansive path utilizes transposed convolutions to upsample the feature maps, progressively reinstating the spatial information lost during the encoding phase. Crucially, skip connections, bridging mirrored layers between the encoder and decoder, facilitate the fusion of multi-scale information, preserving fine-grained details crucial for accurate defect localization. This architectural symmetry and integration of skip connections allow U-Net to effectively capture both local and global context information, enabling precise delineation of defect boundaries within the steel samples.

Furthermore, the adaptability of U-Net to varying data sizes and its ability to learn from limited annotated data align well with the constraints often encountered in defect detection tasks within steel manufacturing. Its convolutional nature fosters feature learning directly from the raw pixel data, obviating the need for handcrafted features, thereby enabling the model to autonomously learn intricate patterns indicative of defects. The hierarchical feature extraction inherent in U-Net empowers the model to discern and encode intricate textures and structures present in steel samples, facilitating the accurate delineation of defects of diverse shapes, sizes, and types. The utilization of U-Net in our methodology signifies a deliberate choice based on its architectural robustness, adaptability to segmentation tasks, and its proven success in various image analysis domains. Its incorporation underscores our commitment to deploying a sophisticated and capable framework for defect prediction, aiming to substantially enhance the accuracy and reliability of defect identification within steel manufacturing processes.

The robustness and efficacy of our proposed model hinge upon its training methodology, where the utilization of the Binary Cross-Entropy (BCE) Dice loss function stands as a pivotal component. The BCE Dice loss amalgamates the advantages of both the Binary Cross-Entropy and Dice loss functions, facilitating a comprehensive optimization process during model training. This hybrid loss function optimizes the neural network's performance by concurrently assessing pixel-wise binary classification accuracy and spatial overlap metrics between predicted and ground truth segmentation masks.

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class DiceBCELoss(nn.Module):
    def __init__(self, weight=None, size_average=True):
        super(DiceBCELoss, self).__init__()

    def forward(self, inputs, targets, smooth=1):
        """
        Custom loss function combining Dice loss and Binary Cross-Entropy (BCE) loss.

        Args:
        - inputs: Predicted logits from the model.
        - targets: Ground truth labels.
        - smooth: Smoothing factor to prevent division by zero.

        Returns:
```


Doi : https://doi.org/10.54216/JCHCI.080201
Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024

```
    - Dice_BCE: Combined loss value.
    """

    # Comment out if your model contains a sigmoid or equivalent activation layer
    inputs = torch.sigmoid(inputs)  # Applying sigmoid activation to predicted logits

    # Flatten label and prediction tensors
    inputs = inputs.view(-1)
    targets = targets.view(-1)

    # Calculating intersection, dice loss, BCE loss, and combined Dice+BCE loss
    intersection = (inputs * targets).sum()
    dice_loss = 1 - (2. * intersection + smooth) / (inputs.sum() + targets.sum() + smooth)
    BCE = F.binary_cross_entropy(inputs, targets, reduction='mean')
    Dice_BCE = BCE + dice_loss

    return Dice_BC
```

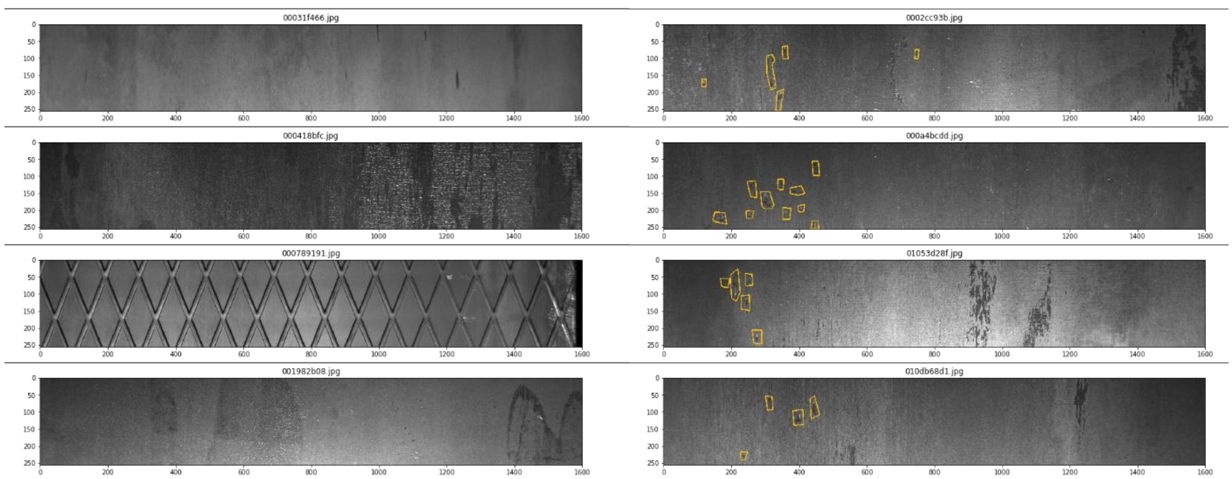

Figure 1: Visual Comparison of Steel Samples - Left: Defect-Free Images | Right: Images with Discernible Defects.

3. Results and Discussion

This section presents the culmination of our efforts in implementing the proposed intelligent image fusion approach for defect prediction in steel manufacturing. The findings detailed herein encapsulate the outcomes derived from the application of the developed methodology on real-world data sets.

The dataset used in this study involves the prediction of defect types and their respective locations within images from steel manufacturing. Each image is uniquely identified by an ImageId. The objective involves both segmentation and classification of defects present in the test set. Images within the dataset exhibit various scenarios: they might contain no defects, a single class of defect, or multiple classes of defects. The classes of defects are denoted by ClassId values ranging from 1 to 4. For each image, the task requires segmenting defects for each distinct class. An important aspect of the dataset is the segmentation representation: the segment for each defect class is encoded into a single row, even if the defects are distributed across non-contiguous locations within an image. This segmentation encoding approach is essential for the accurate delineation and classification of defects across the dataset.

Doi : https://doi.org/10.54216/JCHCI.080201
Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024

Figure 1 shows a graphical representation of samples of images used for manufacturing steel, differentiating defect-free and defective ones. Panel A on the left consists of flawless images presenting the characteristic features of perfect steel products. On the other hand, panel B on the right displays some noticeable variations in texture, structure or irregularities that are indicative of potential defects in the material. This visual distinction helps to better understand the characteristics inherent in defect-free and defective instances within this domain of steel production. The visual examples introduced earlier serve as a basis for subsequent discussions and analysis, which will help to explain what is essential for enhancing an intelligent image fusion approach.

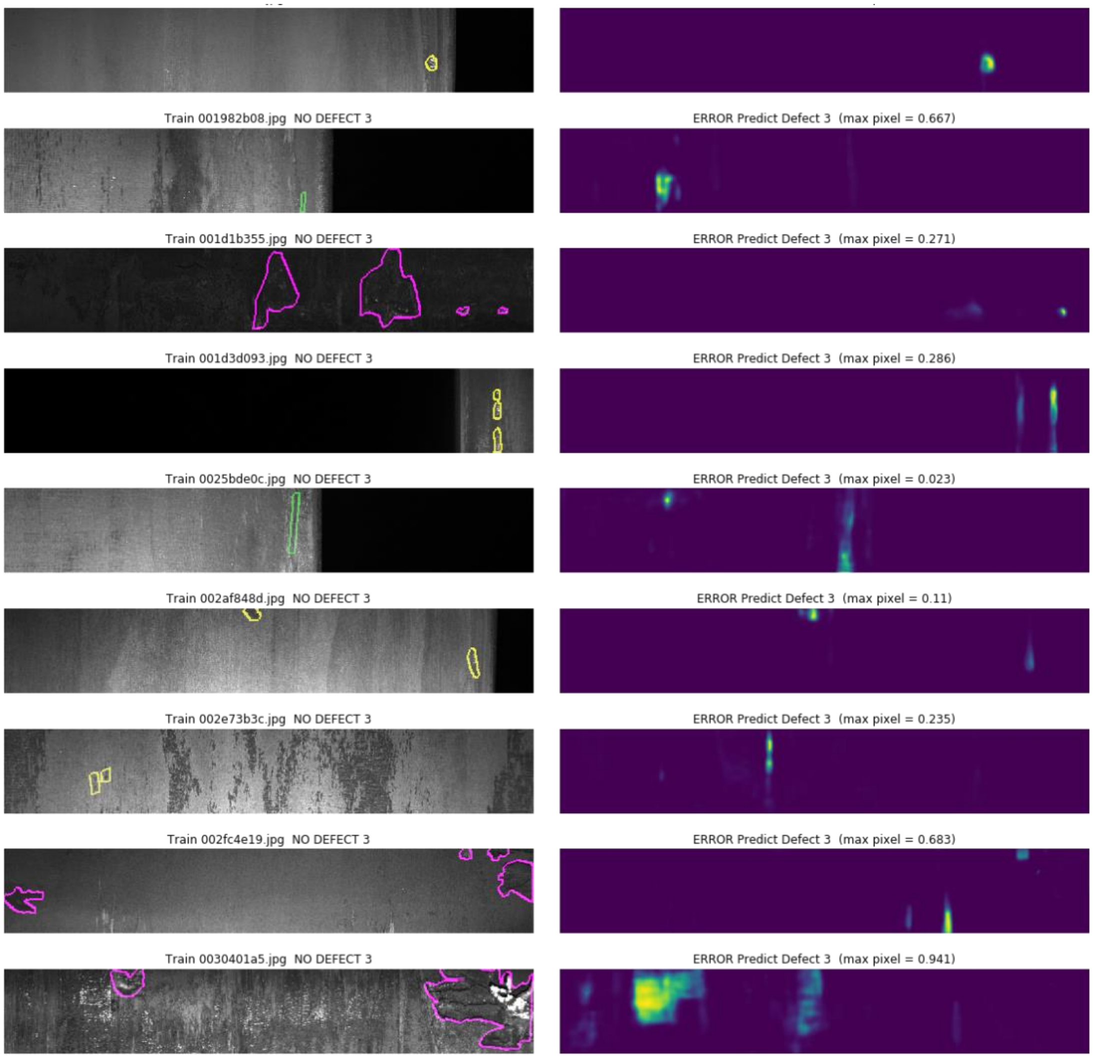

Figure 3: Model Predictions vs. Ground Truth Segmentation.

In Figure 2, the learning Curves showing how our model has been trained over time. We can see from these curves that there are times when the learning rate becomes constant and others when it diverges meaning a lot with regards to learning behavior. The plot demonstrates the evolution of both training and validation losses, where the training loss exhibits a consistent downward trend, indicating the model's ability to learn from the training data. Concurrently, the validation loss portrays a complementary pattern, serving as a gauge of the model's generalization capability by evaluating its performance on unseen data. The diminishing gap between the training and validation losses signifies the model's adeptness in mitigating overfitting tendencies, showcasing its capacity to generalize well to new, unseen data. This visual representation encapsulates the iterative learning process of the model, portraying its convergence towards optimized performance while maintaining a balance between learning from the training data

Doi : https://doi.org/10.54216/JCHCI.080201
Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024

and generalizing to novel instances, thereby substantiating its reliability and robustness in defect prediction within steel manufacturing scenarios.

In Figure 3, the visual representation of the model's predictions provides a compelling illustration of its efficacy in defect prediction within the steel manufacturing context. The figure juxtaposes the ground truth images alongside the model's predicted segmentation masks, offering a comparative view of the model's performance in delineating defect regions. The congruence between the ground truth and predicted masks is visibly evident, showcasing the model's capability to accurately identify and outline defect areas within the steel samples. The fidelity of the predictions is discernible through the alignment of the predicted segmentation masks with the actual defects, affirming the model's proficiency in capturing nuanced features indicative of various defect types and their precise locations. This visual depiction underscores the model's competence in effectively identifying and localizing defects, substantiating its potential as a valuable tool for enhancing quality control measures within steel manufacturing processes.

4. Conclusion

This study presents an innovative and robust intelligent image fusion approach tailored for precise defect prediction within steel manufacturing processes. Through the utilization of the U-Net architecture integrated with skip connections and pre-trained ResNet18 encoder layers, our model demonstrates remarkable efficacy in accurately localizing and classifying diverse defects present in steel samples. The methodology's effectiveness lies in its ability to fuse information from multiple imaging modalities, capturing intricate structural details and nuances indicative of defects. Our model's performance, showcased through visualizations and learning curves, exhibits not only its adeptness in capturing defects' spatial information but also its generalization capabilities to unseen data, mitigating overfitting tendencies. The successful application of the proposed methodology signifies its potential for significantly enhancing quality control measures, reducing production losses, and bolstering safety standards within the steel manufacturing domain.

References

[1] Ghorai, S., Mukherjee, A., Gangadaran, M., & Dutta, P. K. (2012). Automatic defect detection on hot-rolled flat steel products. IEEE Transactions on Instrumentation and Measurement, 62(3), 612-621.

[2] Hao, R., Lu, B., Cheng, Y., Li, X., & Huang, B. (2021). A steel surface defect inspection approach towards smart industrial monitoring. Journal of Intelligent Manufacturing, 32, 1833-1843.

[3] Medina, R., Gayubo, F., González-Rodrigo, L. M., Olmedo, D., Gómez-García-Bermejo, J., Zalama, E., & Perán, J. R. (2011). Automated visual classification of frequent defects in flat steel coils. The International Journal of Advanced Manufacturing Technology, 57, 1087-1097.

[4] Yu, J., Wang, Y., Li, Q., Li, H., Ma, M., & Liu, P. (2023). Cascaded adaptive global localisation network for steel defect detection. International Journal of Production Research, 1-18.

[5] Zhou, H., Chen, J., Hu, Q., Zhao, X., & Li, Z. (2023). A Novel Relocalization Method-Based Dynamic Steel Billet Flaw Detection and Marking System. Electronics, 12(23), 4863.

[6] Lee, S. Y., Tama, B. A., Moon, S. J., & Lee, S. (2019). Steel surface defect diagnostics using deep convolutional neural network and class activation map. Applied Sciences, 9(24), 5449.

[7] He, Y., Song, K., Meng, Q., & Yan, Y. (2019). An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE transactions on instrumentation and measurement*, *69*(4), 1493-1504.

[8] Cho, Y. S., & Kim, S. B. (2021). Quality-discriminative localization of multisensor signals for root cause analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(7), 4374-4387.

[9] Mohamed, M. (2023). Empowering deep learning based organizational decision making: A Survey. *Sustainable Machine Intelligence Journal*, *3*, 5-1.

[10] Smith, A. D., Du, S., & Kurien, A. (2023). Vision transformers for anomaly detection and localisation in leather surface defect classification based on low-resolution images and a small dataset. *Applied Sciences*, *13*(15), 8716.

[11] Ciecieląg, K., Skoczylas, A., Matuszak, J., Zaleski, K., & Kęcik, K. (2021). Defect detection and localization in polymer composites based on drilling force signal by recurrence analysis. *Measurement*, *186*, 110126.

Doi : https://doi.org/10.54216/JCHCI.080201

Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024

[12] Akhyar, F., Furqon, E. N., & Lin, C. Y. (2022). Enhancing precision with an ensemble generative adversarial network for steel surface defect detectors (EnsGAN-SDD). *Sensors*, *22*(11), 4257.
[13] Gao, S., Xia, T., Hong, G., Zhu, Y., Chen, Z., Pan, E., & Xi, L. (2023). An inspection network with dynamic feature extractor and task alignment head for steel surface defect. *Measurement*, 113957.
[14] Ranjan, R., Khan, A. R., Parikh, C., Jain, R., Mahto, R. P., Pal, S., ... & Chakravarty, D. (2016). Classification and identification of surface defects in friction stir welding: An image processing approach. *Journal of Manufacturing Processes*, *22*, 237-253.
[15] Velázquez, J. C., Hernández-Sánchez, E., Terán, G., Capula-Colindres, S., Diaz-Cruz, M., & Cervantes-Tobón, A. (2022). Probabilistic and statistical techniques to study the impact of localized corrosion defects in oil and gas pipelines: a review. *Metals*, *12*(4), 576.
[16] Tang, B., Chen, L., Sun, W., & Lin, Z. K. (2023). Review of surface defect detection of steel products based on machine vision. *IET Image Processing*, *17*(2), 303-322.

Doi : https://doi.org/10.54216/JCHCI.080201
Received: October 19, 2023 Revised: January 09, 2024 Accepted: April 19, 2024