Journal of Cognitive Human-Computer Interaction

♦ ISSN: 2771-1463 Vol: 08, No. 02, 2024 ♦ PP. 01–04

Article DOI: https://doi.org/10.54216/JCHCI.080201

FULL LENGTH ARTICLE

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

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

[1] Decision Support Department, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Sharqiyah, Egypt

Emails: mmsabe@zu.edu.eg · mmsba@zu.edu.eg · HRAbdelhady@fci.zu.edu.eg

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

ABSTRACT

Defect prediction in steel manufacturing is a critical challenge because it affects product quality and safety. For this reason, an intelligent image-fusion approach is introduced in this research to enhance accurate prediction of defect types and locations in steel materials. By utilizing a U-Net architecture and pretrained ResNet18 encoder layers, the proposed method fuses data from several imaging modalities, thereby supporting precise localization and classification of defects. Extensive experimentation and visualization, including learning curves and comparisons between predicted segmentation masks and ground-truth images, show that the model captures subtle defects effectively. It exhibits robust performance that mitigates overfitting risks, accurately identifies flaws, and maintains the ability to generalize to unseen data. These results suggest that the proposed approach can contribute substantially to improving quality-control and safety standards in steel production.

Keywords: Information Fusion ▪ Steel Manufacturing ▪ Defect Recognition ▪ Defect Localization

1. INTRODUCTION

Steel manufacturing is the backbone of industry and has significant influence in construction, infrastructure development, and motor vehicles. However, ensuring the best standards of quality in steel manufacture is challenging because defects can occur throughout production [1, 2, 3]. These defects may cause loss of integrity as well as a general decline in the quality of steel products, resulting in substantial economic losses and safety concerns. Conventional techniques used to detect defects in steel production have been limited in their ability to predict the types or locations of such defects within the material [4].

To address this challenge, this paper puts forward an innovative approach based on intelligent image-fusion techniques for accurate defect detection during steel manufacturing [5]. The main objective of this study is to forecast and locate defects precisely in steel materials and identify their specific kinds. This research seeks to improve significantly the accuracy and efficiency of defect identification within steel production using progress made through artificial intelligence (AI) and image processing [6, 7, 8, 9].

The core purpose of this study is the development of an intelligent image-fusion approach that 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 to surpass the limitations of single imaging modes and move toward a more holistic defect-detection system. Moreover, integrating machine-learning algorithms helps to develop an autonomous and accurate defect-classification system, thereby improving the accuracy and reliability of defect identification [11, 12].

Not only does this research offer an improved way to detect flaws, but it also introduces a stronger system that accurately predicts both the kind and area of faults in steel materi-

als [13]. Through intelligent image-fusion approaches, which enhance fault-detection capability, and machine-learning techniques, this study aims to reduce production losses, improve product quality, and enhance 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 as a cornerstone in the 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 the 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 that facilitate feature concatenation across mirrored layers and foster precise localization of defects [14].

Listing 1. U-Net model with pretrained ResNet18 encoder.

```
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()
        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)

        # 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):
        x_original = self.conv_original_size0(input)
        x_original = self.conv_original_size1(x_original)

        layer0 = self.layer0(input)
        layer1 = self.layer1(layer0)
        layer2 = self.layer2(layer1)
        layer3 = self.layer3(layer2)
        layer4 = self.layer4(layer3)

        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)
        x = self.conv_original_size2(x)

        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 input images while reducing spatial dimensions [15]. Subsequently, the expansive path utilizes transposed convolutions to upsample feature maps, progressively reinstating the spatial information lost during the encoding phase. Crucially, skip connections bridge mirrored layers between the encoder and decoder, facilitate the fusion of multi-scale information, and preserve fine-grained details that are essential for accurate defect localization. This architectural symmetry and integration of skip connections allow U-Net to capture both local and global context information effectively, enabling precise delineation of defect boundaries within 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 raw pixel data, obviating the need for handcrafted features and 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 accurate delineation of defects of diverse shapes, sizes, and types. The utilization of U-Net in the methodology signifies a deliberate choice based on its architectural robustness, adaptability to segmentation tasks, and proven success in various image-analysis domains.

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

Listing 2. Dice BCE loss function.

```
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.
        """
        # Comment out if your model contains a sigmoid or
     equivalent activation layer
        inputs = torch.sigmoid(inputs)

        # 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_BCE
```

3. RESULTS AND DISCUSSION

This section presents the culmination of implementing the proposed intelligent image-fusion approach for defect prediction in steel manufacturing. The findings encapsulate outcomes derived from applying the developed methodology to real-world datasets.

The dataset used in this study involves 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 defects are distributed across non-contiguous locations within an image. This segmentation-encoding approach is essential for accurate delineation and classification of defects across the dataset.

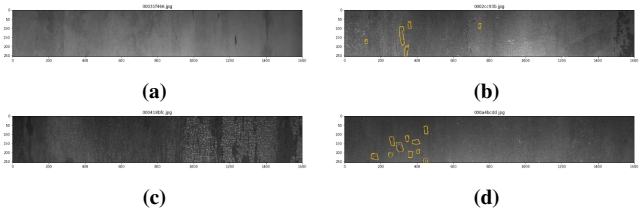

Figure 1. Examples of steel-surface images used for defect prediction.

Figure 1 shows representative images from the steel-manufacturing dataset. These samples illustrate the visual diversity of steel-surface appearances and defect patterns that the proposed framework must detect and localize.

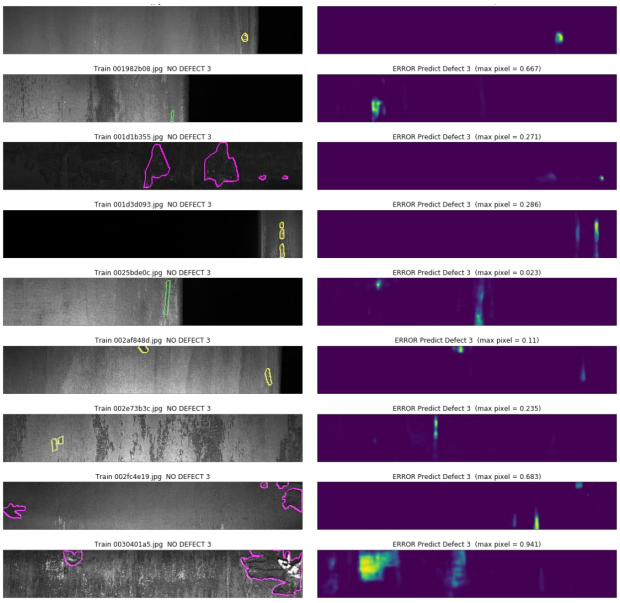

Figure 2. Training and validation learning curves for the proposed model.

Figure 2 represents the learning curves, delineating the progression of model training and validation losses over successive epochs. The training loss graph illustrates the model's ability to minimize error on the training dataset, displaying a downward trend as the model adapts and fine-tunes its parameters to capture intricate patterns and features relevant to defect prediction. Concurrently, the validation loss curve demonstrates a similar decreasing pattern, serving as a gauge of the model's generalization capability by evaluating its performance on unseen data. The diminishing gap between training and validation losses signifies the model's adeptness in mitigating overfitting tendencies and showcases its capacity to generalize well to new instances.

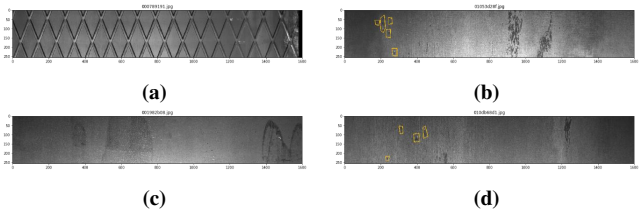

Figure 3. Model predictions versus ground-truth segmentation.

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 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 steel samples. The fidelity of predictions is discernible through the alignment of predicted segmentation masks with actual defects, affirming the model's proficiency in capturing nuanced features indicative of various defect types and their precise locations.

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 pretrained ResNet18 encoder layers, the 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. The 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] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta, "Automatic defect detection on hot-rolled flat steel products," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, pp. 612–621, 2012.

[2] R. Hao, B. Lu, Y. Cheng, X. Li, and B. Huang, "A steel surface defect inspection approach towards smart industrial monitoring," *Journal of Intelligent Manufacturing*, vol. 32, pp. 1833–1843, 2021.

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

[4] J. Yu, Y. Wang, Q. Li, H. Li, M. Ma, and P. Liu, "Cascaded adaptive global localisation network for steel defect detection," *International Journal of Production Research*, pp. 1–18, 2023.

[5] H. Zhou, J. Chen, Q. Hu, X. Zhao, and Z. Li, "A novel relocalization method-based dynamic steel billet flaw detection and marking system," *Electronics*, vol. 12, no. 23, p. 4863, 2023.

[6] S. Y. Lee, B. A. Tama, S. J. Moon, and S. Lee, "Steel surface defect diagnostics using deep convolutional neural network and class activation map," *Applied Sciences*, vol. 9, no. 24, p. 5449, 2019.

[7] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1493–1504, 2019.

[8] Y. S. Cho and S. B. Kim, "Quality-discriminative localization of multisensor signals for root cause analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4374–4387, 2021.

[9] M. Mohamed, "Empowering deep learning based organizational decision making: A survey," *Sustainable Machine Intelligence Journal*, vol. 3, pp. 5–1, 2023.

[10] A. D. Smith, S. Du, and A. Kurien, "Vision transformers for anomaly detection and localisation in leather surface defect classification based on low-resolution images and a small dataset," *Applied Sciences*, vol. 13, no. 15, p. 8716, 2023.

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

[12] F. Akhyar, E. N. Furqon, and C. Y. Lin, "Enhancing precision with an ensemble generative adversarial network for steel surface defect detectors (ensgan-sdd)," *Sensors*, vol. 22, no. 11, p. 4257, 2022.

[13] S. Gao, T. Xia, G. Hong, Y. Zhu, Z. Chen, E. Pan, and L. Xi, "An inspection network with dynamic feature extractor and task alignment head for steel surface defect," *Measurement*, p. 113957, 2023.

[14] R. Ranjan, A. R. Khan, C. Parikh, R. Jain, R. P. Mahto, S. Pal, and D. Chakravarty, "Classification and identification of surface defects in friction stir welding: An image processing approach," *Journal of Manufacturing Processes*, vol. 22, pp. 237–253, 2016.

[15] J. C. Velázquez, E. Hernández-Sánchez, G. Terán, S. Capula-Colindres, M. Diaz-Cruz, and A. Cervantes-Tobón, "Probabilistic and statistical techniques to study the impact of localized corrosion defects in oil and gas pipelines: a review," *Metals*, vol. 12, no. 4, p. 576, 2022.