METALSEER: Metallographic Segmentation for Efficient Extraction and Recognition of Component Structures in Steel Images

Or

Multi-Phase Steel Microstructure Segmentation using UNet: Generalization and Magnification Independence Analysis

# Abstract

Image segmentation plays a vital role in enhancing industrial processes by leveraging raw material imaging. In particular, the microstructures of ultra-high carbon steel contain valuable information about the steel's physical properties, determined by constituents such as Martensite, Ferrite, and Bainite. However, the current manual and subjective analysis of micrographs by material scientists is time-consuming and tedious. To address this, we propose an automated approach using deep learning-based image segmentation algorithms to label steel microstructures in a subset of ultrahigh carbon steel images. Our proposed UNet model requires only a small number of annotated micrographic images to train the segmentation algorithm. Our results demonstrate the effectiveness of the proposed approach in accurately segmenting steel microstructures in different magnified images and steel types, outperforming previous methods that either trained and predicted on the same dataset or failed to capture fine-detailed microstructures. The automatization of microstructure characterization is an important industrial application that requires active research, and our work contributes to addressing this need by providing a scalable and efficient metallographic segmentation solution.

# Introduction

The rapid advancement of Artificial Intelligence (AI) has revolutionized various industries, including manufacturing processes [1]. AI systems aim to support experts by encoding significant knowledge into models, becoming an integral part of the Industry 4.0 revolution [2]. In this context, image segmentation of metallographic images plays a crucial role in understanding the characteristics of materials and manufacturing procedures, thereby improving product quality assessment and facilitating the development of new products [3], [4]. Traditional qualitative and quantitative analyses of segmented microstructures are typically performed manually by materials scientists, which is time-consuming and labor-intensive.

Among the various applications in industrial manufacturing, our focus is on the segmentation of metallographic images. Image segmentation, a fundamental computer vision problem, involves partitioning an image into distinct regions based on certain criteria [5]. It has been extensively studied and finds application in numerous domains, such as medical image analysis [6], [7], autonomous driving [8], and robotics [9].

However, the pixel-wise annotation required for previous segmentation approaches is highly time-consuming, hindering the efficiency of microstructure characterization [10]. Therefore, automating the characterization of microstructures using DL-based image segmentation has gained significant attention, serving as an assistant tool for experts in this demanding task. The segmentation of microscopic steel images is a challenging task due to the intricate nature of steel microstructures and the similarity in appearance among different components when observed under varying magnifications. Unlike general image segmentation problems where objects have distinct boundaries, the components in steel microstructures often exhibit subtle differences and complex structures that require careful analysis for accurate segmentation [11]. Steel microstructures contain critical information about the mechanical properties and quality of the material, making it essential to accurately separate and identify the different components. This level of detail and precision is particularly crucial in applications such as additive manufacturing, quantitative analysis, and quality control.

Despite the significance of steel image segmentation, there is a scarcity of research specifically addressing this challenge. The existing literature in the field of metallographic segmentation predominantly focuses on limited datasets, where the models are trained and tested on the same set of data, limiting their generalizability. Moreover, previous methods often fail to capture the fine-detailed microstructures, impeding their practical utility in real-world applications [11, 12]. Given the complexity of steel microstructures, the variability of components under different magnifications, and the limited research in this domain, there is a pressing need for novel approaches that can effectively address these challenges. In this study, we propose a scalable metallographic segmentation solution that leverages deep learning techniques and incorporates advanced augmentation strategies to overcome the limitations of traditional methods. By training on a known set of inputs and testing on diverse magnified images and steel types, our approach aims to provide accurate and detailed segmentation results, advancing the field of steel image analysis and supporting industrial applications in a more robust and comprehensive manner.

By addressing the limitations of manual segmentation and utilizing DL capabilities, our research contributes to the advancement of AI in industrial manufacturing processes, promoting improved product quality assessment and supporting the development of innovative materials. The subsequent sections of this paper elaborate on our proposed methodology and present experimental results demonstrating its effectiveness in automating microstructure characterization. Section 2 reviews related works in the domain. Section 3 discusses the characteristics and challenges of microstructure segmentation, as well as the UNet architecture and dataset. Section 4 presents the methodology, including model architecture, augmentations, and training process. Section 5 focuses on experimentation, covering evaluation metrics, experimental setup, and results on both training and inference images. Section 6 provides a detailed discussion and analysis of the experimental results, addressing challenges, limitations, and factors influencing performance variations. Finally, section 7 concludes by summarizing the key findings, emphasizing the importance of accurate microstructure segmentation, and suggesting future research directions.

# Related Works

In the realm of microstructure segmentation, previous studies have explored various methodologies to achieve accurate classification and segmentation results. Velichko et al. [13] proposed a method based on data mining techniques, specifically extracting morphological features and utilizing a feature classification step using Support Vector Machines (SVMs). This approach was applied to cast iron and demonstrated the potential of machine learning in microstructural analysis. Similarly, Pauly et al. [14] adopted a similar approach on a contrasted and etched dataset of steel acquired through SEM and LOM imaging. However, the results yielded a relatively low accuracy of 48.89% in microstructural classification due to the complex nature of substructures and the lack of discriminative features.

Deep learning methods have gained significant attention in various computer vision tasks, including object classification and image semantic segmentation. The introduction of AlexNet by Krizhevsky et al. [15], a CNN architecture with 7 layers, revolutionized the field and achieved remarkable success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. The significant improvement in accuracy by AlexNet brought deep learning methods to the forefront of research.

Another influential CNN architecture, VGGNet by Simonyan et al. [16], further advanced the field by introducing a deeper network with improved accuracy. These models set the stage for the utilization of convolutional neural networks in various computer vision tasks, including microstructure segmentation.

Long et al. [17] proposed Fully Convolutional Neural Networks (FCNNs), a pioneering work that adapted object classification CNNs to semantic segmentation tasks. This approach, along with its subsequent extensions, has become state-of-the-art in semantic segmentation on benchmark datasets such as Pascal VOC image segmentation challenge [18] and Cityscape [19]. FCNNs have demonstrated their effectiveness in capturing fine-grained details and accurately segmenting objects in images.

While deep learning methods have achieved remarkable success in various domains, their application to microstructure segmentation is relatively nascent. The unique challenges posed by microstructure images, such as complex substructures and the need for discriminative features, require tailored approaches. Therefore, in this work, we aim to leverage the power of deep learning, particularly FCNNs, to tackle the challenges of microstructure segmentation in steel images captured through SEM.

# Background

This section provides an overview of the key aspects of the semantic segmentation problem in the context of microstructures. It also discusses the dataset utilized in this study and highlights the associated challenges.

## Characteristics and Challenges of Microstructure Segmentation:

Segmenting microstructures in metallographic images presents a highly challenging task due to the distinctive visual characteristics inherent in these images.

Microstructure segmentation in microscopic steel images presents a complex task due to multiple factors. Firstly, these images possess high-resolution details that require precise detection and segmentation of fine-grained structures. Furthermore, the inherent variability in textures and shapes within these images adds to the challenge of accurately distinguishing and segmenting different regions. Additionally, imbalanced class distributions, where some components are more prevalent than others, introduce difficulties in achieving accurate segmentation results across all classes. Moreover, the absence of landmark information in these images limits the availability of unique characteristics that could aid in capturing the distinct features of microstructures [11]. Due to heterogenous material properties in these metallographic images, it becomes difficult to accurately delineate and classify different regions. Heterogenous properties also heighten the variations in scale and magnification, leading to changes in the appearance and size of the microstructures.

Manual annotation of metallographic images is a labor-intensive and time-consuming process. As a result, there is a scarcity of large, labeled datasets specifically tailored for microstructure segmentation [28]. This limitation hinders the application of supervised learning methods, which typically require ample training data to achieve optimal performance.

## Leveraging U-Net for Precise Segmentation of Steel Microstructures

In steel image segmentation, the choice of an appropriate model is critical for achieving accurate results. Considering the challenges specific to steel microstructure segmentation, we selected the U-Net architecture [20] for its suitability.

Steel microstructure segmentation poses challenges due to fine-grained details, complex substructures, and imbalanced class distributions. Unsupervised and semi-supervised learning approaches may struggle to handle these challenges effectively and may result in suboptimal segmentation. They often have limitations in accurately classifying complex substructures and lack the required expert supervision for reliable results in steel image segmentation.

To overcome the challenges and limitations, we opted for a supervised learning approach using the U-Net architecture. U-Net has demonstrated exceptional performance in segmentation tasks, particularly in capturing fine details and handling varying scales of structures [21]. U-Net's U-shaped design preserves fine-grained details while capturing global context, making it suitable for steel microstructure segmentation. The incorporation of skip connections enables the fusion of features from multiple scales, enhancing the model's ability to capture intricate details. By leveraging the availability of pixel-wise labels generated through Electron Backscatter Diffraction (EBSD) [22], we train the U-Net model to accurately segment steel microstructures. The use of supervised learning with U-Net ensures reliable and precise segmentation results.

## Dataset

The dataset used in this study comprises 6 E type steel images captured using a Scanning Electron Microscope (SEM). To obtain ground truth labels, the corresponding label images were generated using Electron Backscatter Diffraction (EBSD). The label images provide precise information about the different components and phases present in the microstructures.

A picture containing fabric, pattern

Description automatically generatedA picture containing colorfulness, map, yellow, graphics

Description automatically generated

1. (b)

Fig. 1 (a) x2700 magnified E type SEM image (b) Label image by EBSD

Figure 1 visually illustrates an example pair of the SEM image and the corresponding label image. The label image represents different categories within the steel image, which are classified into three labels: Ferrite (represented by orange), Bainite (represented by purple), and Martensite (represented by yellow). These labels serve to identify and distinguish the different components and structures present in the steel image.

During the training phase, only 5 E type steel images were utilized to augment and train the model and the last one was used for testing. However, to evaluate the performance and generalization capability of the model, we conducted tests on steel images magnified at 3000x and 5000x. In addition to E type steel images, the testing dataset also included images of A type, H2 type, and D3 type steels.

A picture containing colorfulness, screenshot, art

Description automatically generated

(e)

(d)

(c)

(b)

(a)

Fig. 2. (a) x3000 magnified E type, (b) x5000 magnified E type, (c) x5000 magnified A type, (d) x5000 magnified D3 type, (e) x5000 magnified H2 type SEM and label image pairs.

In the inference phase, Figure 2 displays the images used for evaluating the models. The information bar located at the bottom of the images was removed prior to the inference process. The specific resolutions of the images can be found in Table 1. Notably, during training, the models were trained using images of size 830x820. However, during testing, the models were evaluated on images of different resolutions. This flexibility was achievable due to the utilization of a FCNN UNet model, which is not dependent on the input size of the image during the inference stage.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 1** | | | |
| Image Type | Images |  | Resolution |
| X2700 E type | 6 |  | 830x820 |
| X3000 E type | 10 |  | 1080x1024 |
| X5000 E type | 8 |  | 1080x1024 |
| X5000 A type | 1 |  | 1080x1024 |
| X5000 D3 type | 1 |  | 1080x1024 |
| X5000 H2 type | 1 |  | 1080x1024 |

In FCNNs, the convolutional layers are responsible for capturing local spatial information from the input images, while the pooling layers down sample the spatial dimensions to extract higher-level features. The use of convolutional and pooling operations allows FCNNs to maintain spatial information while reducing the dimensionality of the feature maps. As a result, the FCNN model can effectively process images of different sizes by adapting its receptive field and feature extraction capabilities [20]. This flexibility is advantageous when dealing with datasets that contain images of varying resolutions or when performing inference on images that differ in size from the training data.

# Methodology

In this section, we describe the methodology adopted for the segmentation of microstructures in steel images. Our approach leverages the power of deep learning, specifically utilizing the architecture of Fully Convolutional Neural Networks (FCNNs) in UNet for accurate and robust segmentation. UNet is a state-of-the-art UNet model, which has proven to be effective in medical image segmentation tasks which shares some common traits with steel segmentation. We employed various augmentation techniques to enhance the robustness of the model and improve its performance across different magnifications and steel types.

## Model Architecture

The proposed segmentation model is based on the U-Net architecture, which has been widely used for image segmentation tasks due to its effectiveness in capturing fine-grained details and preserving spatial information. The U-Net architecture consists of an encoder path that captures contextual information and a decoder path that enables precise localization.

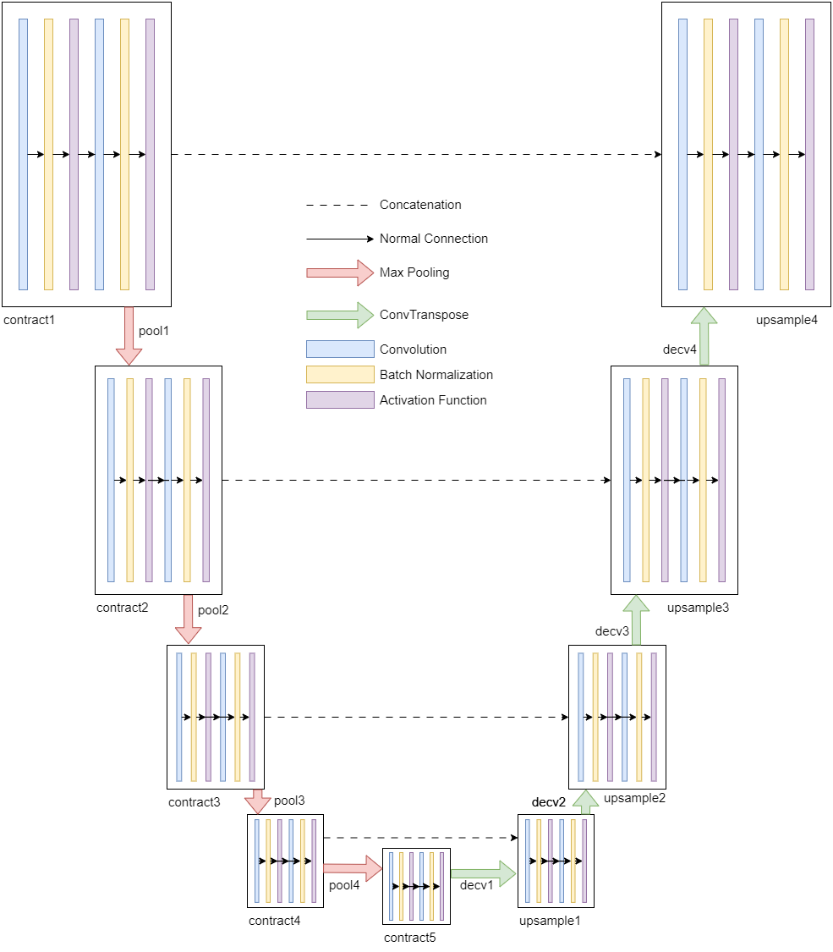


Fig. 3. UNet Architecture used in the proposed model.

The encoder path begins with a contraction module (contract1) that applies a series of convolutional operations to extract low-level features from the input image. This is followed by pooling operations (pool1, pool2, pool3, pool4) that progressively reduce the spatial dimensions of the feature maps while increasing the number of channels. Each contraction module (contract2, contract3, contract4, contract5) further applies convolutional operations to extract higher-level features.

The decoder path of the U-Net model aims to recover spatial information and refine the segmentation output. It starts with an expansion module (decv1, short for de-convolution) that performs either convolution transpose or pixel shuffling operations, depending on the specified anti-aliasing type. This is followed by an up-sampling operation (upsample1) and a concatenation step that combines features from the corresponding encoder path layer. Similar operations are performed in subsequent decoder modules (decv2, decv3, decv4) to further refine the features.

The final decoder module (decv5) applies a convolutional operation to map the extracted features to the desired number of output classes. The activation functions used in the model include sigmoid, ReLU, leaky ReLU, and swish, providing non-linearity and enabling better feature representation.

The U-Net model is trained in a supervised learning fashion, where the pixel-wise labeled images are used as ground truth for optimization. The model is trained to minimize the loss between the predicted segmentation and the ground truth labels. By leveraging the U-Net architecture, the proposed model can effectively capture the intricate microstructural details present in steel images and accurately segment different regions of interest. The combination of contraction and expansion modules allows the model to learn both global contextual information and local fine-grained details, leading to precise segmentation results.

## Augmentations

Augmentation techniques help alleviate the problem of insufficient data by artificially expanding the training set through various transformations and variations applied to the existing samples [23]. In this section, we describe the augmentation techniques used in our study.

* Flip (x, y): Flipping the images horizontally (x-axis) and vertically (y-axis) helps introduce mirror-like variations in the dataset. This augmentation is justified because the orientation of steel structures can vary, and flipping allows the model to learn robust features regardless of the direction of the structures.
* Rotation (0-90 degrees): Rotation augmentation randomly rotates the images within the range of 0 to 90 degrees. This augmentation is beneficial in scenarios where the steel structures may be present at different orientations in the SEM images. By applying rotations, the model becomes invariant to the specific angle of the structures and can generalize better.
* Zoom (1-2.5x): Zoom augmentation randomly scales the images within the range of 1 to 2.5 times their original size. This augmentation simulates variations in magnification levels, which can occur in different SEM images. By including zoomed-in or zoomed-out versions of the images, the model learns to handle different scales of steel structures.
* Intensity (0-10): Intensity augmentation applies random variations to the pixel intensities of the images. This augmentation introduces changes in the brightness or darkness of the image, mimicking diverse lighting conditions or material properties. It helps the model become robust to intensity variations in the steel images.
* Gamma (0-10): Gamma augmentation adjusts the gamma value of the images, which can enhance or suppress certain intensity levels. By altering the gamma, this augmentation simulates changes in contrast and brightness. It helps the model learn features that are invariant to different contrast levels.
* Contrast (HE): Contrast augmentation applies histogram equalization (HE) to enhance the contrast of the images. This technique redistributes the pixel intensities to increase the overall contrast. It helps the model capture finer details and improve segmentation accuracy in low-contrast areas of the steel images.
* Sliding Window: Sliding window augmentation involves dividing the large input images into smaller patches of size 800x800 pixels with a stride of 5 pixels. Keeping large patch sizes helps in capturing the structures in the image rather than making smaller patches where the larger information on structure of components is lost.

By augmenting the dataset with flipped, rotated, zoomed, and intensity-altered images, the model becomes more robust to different orientations, scales, and varied microstructures. The sliding window technique aids in making more patches of the images. Collectively, these augmentations help improve the model’s performance and ability to accurately segment steel structures in SEM images.

## Training process

The training process involved several parameter settings, optimization methods, and loss functions.

For optimization, the Adam optimizer was utilized, which is known for its efficient adaptive learning rate [24]. The kernel size for the convolutional layers in the model was set to 3, allowing the model to capture local information effectively. A batch size of 16 was chosen to balance memory usage and training efficiency. The learning rate was set to 0.00001, ensuring a small step size for smoother convergence during training. This value was carefully selected through experimentation and fine-tuning to achieve optimal performance. The model architecture consisted of 4 depth levels and had 16 channels. The depth of the model determines the number of contracting and expanding layers, allowing the model to capture both global and local features. To effectively train the model, we tried experimenting with both focal loss and Jaccard loss as the loss functions. The focal loss emphasizes challenging samples, helping the model focus on hard-to-segment regions. The Jaccard loss, also known as intersection-over-union (IoU) loss, measures the similarity between predicted and ground truth masks, encouraging accurate segmentation.

Various activation functions were explored during the training process. Initially, the Sinusoidal Representation Networks (Siren) [25] activation function was used, but after comparing its performance with the Rectified Linear Unit (ReLU) activation function, it was found that ReLU performed better. Additionally, the model was experimented with leaky ReLU and Swish activation functions to check and compare performance.

A yellow and purple spots on an orange background

Description automatically generated with low confidence

Fig. 4. An example of adding boundary class to x3000 E type steel label image.

Blur pooling layers were introduced as an experiment in the model and three variations were tested: up, down, and both up and down blur pooling layers. Blur pooling helps in reducing the spatial resolution of feature maps while preserving important information, potentially aiding the model in capturing larger-scale patterns and reducing overfitting [26]. Another experiment involved adding a boundary class to the labels by drawing edges between the classes. Figure 4 shows an example of an image with boundary class. This approach aimed to improve the model’s ability to accurately segment the boundaries of steel structures, which are often crucial for proper classification and analysis and often times improves performance of the model [27].

Each experiment and modification in the training process was conducted with the goal of enhancing the model’s segmentation performance. By exploring different parameter settings, optimization methods, loss functions, activation functions, pooling techniques, and label modifications, the aim was to identify the best combination of these factors to achieve the most accurate and reliable steel image segmentation results.

# Experimentation

## Evaluation Metrics

Mean Pixel accuracy was calculated as an evaluation metrics where, is the number of the pixel of class predicted as class is the number of different classes and is the whole pixel number of class .

Additionally, the Dice score, also known as the F1 score, was calculated to evaluate the overlap between the predicted segmentation masks and the ground truth masks.

Where, A is the predicted segmentation mask, B is the ground truth mask, |A| represents the number of pixels in A, and |B| represents the number of pixels in B. The numerator, 2 \* |A ∩ B|, calculates the intersection between the predicted and ground truth masks multiplied by 2. The denominator, |A| + |B|, calculates the sum of the pixels in both masks. The Dice coefficient ranges from 0 to 1, with a value of 1 indicating a perfect overlap between the predicted and ground truth masks.

However, it was observed that relying solely on overall accuracy and Dice score could be misleading, as some classes might achieve high accuracy while the model struggles to accurately detect certain structures. This situation could skew the overall accuracy towards the higher side, giving a false impression of the model’s performance.

To address this issue, class-wise accuracy and the ratio of classes in predicted and labeled images were introduced as additional evaluation metrics. Class-wise accuracy measures the accuracy of the model’s predictions for each individual class, providing a more detailed understanding of its performance across different classes. This allows for identifying specific classes where the model may be underperforming. The ratio of classes in predicted and labeled images compares the distribution of classes in the predicted segmentation masks with the distribution in the ground truth labels. This metric helps identify any significant deviations in the predicted class proportions, which can be indicative of biases or imbalances in the model’s predictions.

## Experiment Setup

The experiments for the steel image segmentation model were conducted on a system with the following specifications: NVIDIA A6000 GPU, Intel i7 6700 CPU, Ubuntu 22.10 operating system, and PyTorch framework.

To create a training dataset, 5 out of the 6 available images were used for augmentation. The remaining image was reserved for testing. This splitting strategy ensured that enough images were available for training while still having a separate dataset for evaluating the model’s generalization performance. The training protocol involved the use of the augmented dataset with 5 images. The dataset was further split into a training set and a validation set. The split ratio was set to 0.8, meaning 80% of the augmented images were used for training, and the remaining 20% were used for validation. This split ensured that the model’s performance could be monitored and fine-tuned during the training process.

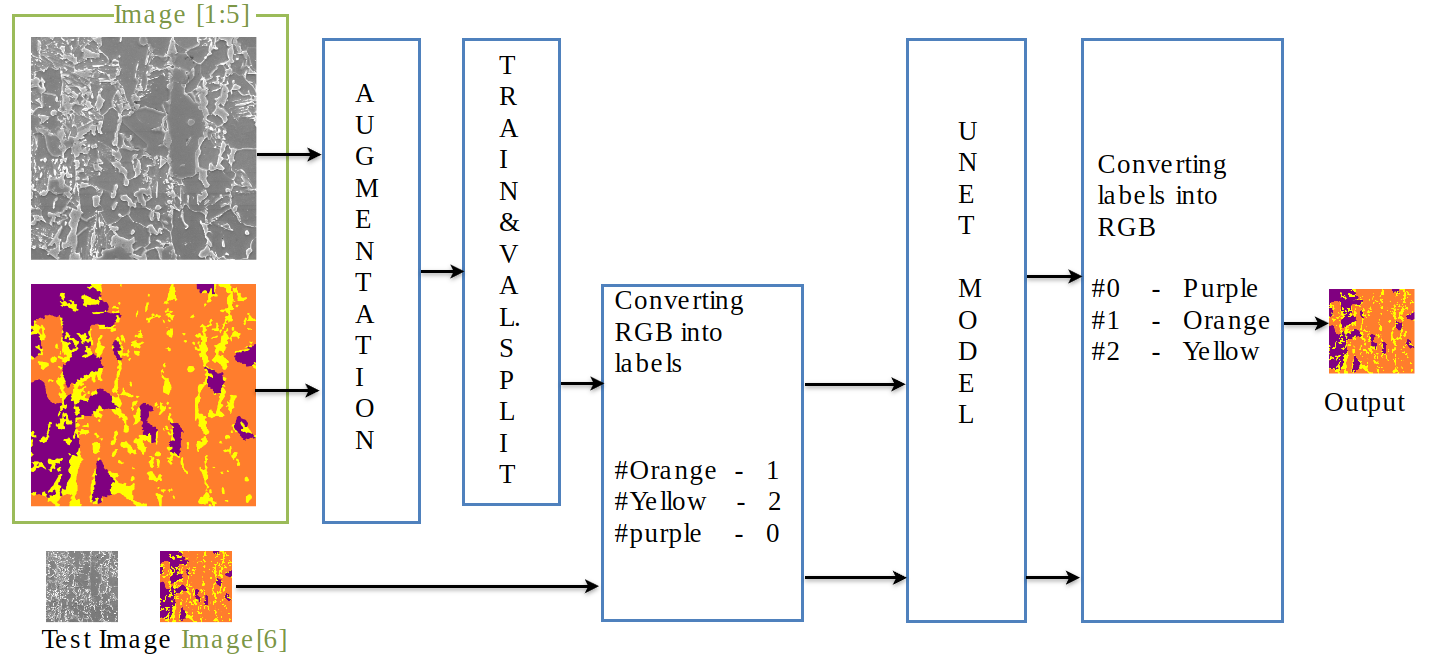


Fig. 5 illustrates the overall flow of the system.

After the split, the RGB pixel values of the label images are converted to class labels of 0, 1 and 2 where purple representing Bainite is denoted by 0, orange representing Ferrite is denoted by 1 and Yellow which is Martensite is denoted by 2. These are the outputs that are predicted by the UNet model which are again re-converted into corresponding pixel values after prediction to generate the output image.

The testing protocol involved evaluating the trained model on the reserved image that was not used for training or augmentation. This image served as an independent sample to assess the model's performance on unseen data and measure its ability to accurately segment the steel structures. X3000, x5000, A, H2, D3 steel type images were inferenced by loading the saved model after the model was trained.

## Experiment Results

### Activation function

In this experiment, we explored the effects of different activation functions on the performance of our model. Siren activation functions enable neural networks to model continuous and periodic functions efficiently. They have been shown to provide better representation of high-frequency patterns and enable neural networks to generalize well across different input spaces. ReLU is widely used due to its simplicity and ability to alleviate the vanishing gradient problem. It introduces non-linearity, allowing the model to learn complex patterns. As Siren is known for capturing complex patterns and functions it was expected that it could effectively model intricate and non-linear relationships within the data, potentially improving the performance of the model. However, in practice, it was observed that ReLU performed better than Siren in the context of the steel image segmentation task as can be seen in Table 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2 – Showing accuracy and dice measurements along with class-wise accuracy for models with different activation functions | | | | | | | | | | | | | | | |
| Image Type  Activation | X3000 E Type | | | X5000 E Type | | | A Type | | | D3 Type | | | H2 Type | | |
| Siren | **86.12 (0.8487)** | | | 89.77 (0.8853) | | | 74.87 (0.7383) | | | 40.02 (0.3994) | | | 69.8 (0.6855) | | |
| 63 | 96.4 | 31.2 | 66.2 | 97.8 | 34.8 | 47.6 | 91 | 58.7 | 60.7 | 67.8 | 34 | 72.5 | 90.3 | 24.6 |
| ReLU | 83.97 (0.8272) | | | **90.01 (0.8976)** | | | **77.41 (0.7716)** | | | **64.75 (0.6397)** | | | **73.89 (0.7364)** | | |
| 60.1 | 91.4 | 55.1 | 74.3 | 94.2 | 62.2 | 50 | 81.8 | 82.3 | 74 | 32.1 | 63.5 | 81.5 | 74 | 62.4 |
| Leaky ReLU | 84.77 (0.8452) | | | 88.53 (0.8831) | | | 75.19 (0.7493) | | | 59.59 (0.5855) | | | 69.69 (0.6938) | | |
| 66.2 | 94.9 | 49.9 | 62.8 | 97.2 | 38.5 | 43.3 | 91.4 | 69.7 | 74.2 | 41.2 | 59.8 | 70.4 | 86.4 | 37.1 |
| Swish | 84.21 (0.8396) | | | 88.67 (0.8842) | | | 72.41 (0.7217) | | | 56.54 (0.5649) | | | 70.00 (0.6973) | | |
| 64 | 95.2 | 44.9 | 63.4 | 97.4 | 36.2 | 46.5 | 92.4 | 61.2 | 77.3 | 47.1 | 52.2 | 66.9 | 90.2 | 36.4 |

### Effects of Blur Pooling

Adding blur pooling layers in model are known to benefit from noise reduction, scale-invariant representation, reduced sensitivity to local variations, and improved object boundary definition. These factors collectively contribute to enhancing the model's ability to find and segment microstructures in steel images, leading to more accurate and reliable results.

To explore the impact of Blur Pooling on the model's performance, we conducted experiments incorporating different types of Blur Pooling layers. These included Down Blur-pooling, Up Blur-pooling, and both Up and Down Blur-pooling. Interestingly, we observed that Blur Pooling showed improvements when combined with the Siren activation function. However, when used in conjunction with the ReLU activation function, Blur Pooling performed poorly.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 3 - Showing accuracy and dice measurements along with class-wise accuracy for models with different Blur pooling layers | | | | | | | | | | | | | | | |
| Image Type  Blur Pool | X3000 E Type | | | X5000 E Type | | | A Type | | | D3 Type | | | H2 Type | | |
| None | 83.97 (0.8272) | | | **90.01 (0.8976)** | | | **77.41 (0.7716)** | | | **64.75 (0.6397)** | | | **73.89 (0.7364)** | | |
| 60.1 | 91.4 | 55.1 | 74.3 | 94.2 | 62.2 | 50 | 81.8 | 82.3 | 74 | 32.1 | 63.5 | 81.5 | 74 | 62.4 |
| Down | 83.53 (0.8329) | | | 87.69 (0.8743) | | | 70.83 (0.7061) | | | 52.5 (0.5227) | | | 68.02 (6777) | | |
| 60.9 | 95.2 | 42.6 | 61.7 | 97.2 | 38.4 | 39.5 | 93.4 | 58.8 | 68.2 | 56.6 | 49.1 | 62 | 90.9 | 33.9 |
| Up | 84.41 0.8416 | | | 87.93 (0.8768) | | | 76.73 (0.7648) | | | 45.04 (0.4479) | | | 69.0 (0.6875) | | |
| 65.5 | 94.7 | 47.2 | 62.3 | 96.9 | 44 | 47.8 | 89.9 | 74.5 | 64.7 | 67 | 40.3 | 65.8 | 89.5 | 35.1 |
| Down + Up | **84.3 (0.8405)** | | | 87.51 (0.8726) | | | 74.5 (0.7425) | | | 40.24 (0.3999) | | | 66.78 (0.6653) | | |
| 64.9 | 94.6 | 44.2 | 61.1 | 97.5 | 33.5 | 44.1 | 90.7 | 68.6 | 62.5 | 75.9 | 24.9 | 60.9 | 90.4 | 31.1 |

### Loss Functions

We experimented with two different loss functions: Focal Loss and Jaccard Loss. Both loss functions have specific properties that make them suitable for image segmentation tasks, such as finding microstructures in steel images. By using Focal Loss, we aim to improve the model's ability to distinguish and accurately classify microstructures, even in the presence of class imbalance and challenging examples. Jaccard Loss is particularly well-suited for segmentation tasks as it directly evaluates the quality of the segmentation output. It encourages the model to produce precise and accurate boundaries of microstructures, as it explicitly penalizes false positives and false negatives in the predicted mask.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 4 - Showing accuracy and dice measurements along with class-wise accuracy for models with different loss functions | | | | | | | | | | | | | | | |
| Image Type  Loss | X3000 E Type | | | X5000 E Type | | | A Type | | | D3 Type | | | H2 Type | | |
| Focal | 83.97 (0.8272) | | | **90.01 (0.8976)** | | | 77.41 (0.7716) | | | 64.75 (0.6397) | | | 73.89 (0.7364) | | |
| 60.1 | 91.4 | 55.1 | 74.3 | 94.2 | 62.2 | 50 | 81.8 | 82.3 | 74 | 32.1 | 63.5 | 81.5 | 74 | 62.4 |
| Jaccard | **84.11 (0.8389)** | | | 89.29 (0.8892) | | | **77.78 (0.7731)** | | | **68.32 (0.6809)** | | | **73.92 (0.7376)** | | |
| 68.11 | 93.8 | 45.3 | 70.9 | 96.6 | 41.7 | 51.4 | 92.1 | 75.5 | 68.2 | 56.6 | 49.1 | 78.5 | 88.1 | 52.6 |

### Classification with Boundary Class

The addition of a boundary class in the classification task aims to improve the model's performance by explicitly addressing the challenge of accurately delineating the boundaries of microstructures in steel images. By incorporating a boundary class into the classification task, we expect the model to achieve better performance in accurately segmenting microstructures, particularly by improving boundary localization, enhancing discrimination, reducing ambiguity, and enabling fine-grained analysis.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 5 - Showing accuracy and dice measurements along with class-wise accuracy for models with boundary class and non-boundary class | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Image  Class | X3000 E Type | | | | | X5000 E Type | | | | | A Type | | | | | | | D3 Type | | | | | | H2 Type | | | | | |
| No Boundary Class | **84.11 (0.8389)** | | | | | **89.29 (0.8892)** | | | | | **77.78 (0.7731)** | | | | | | | **68.32 (0.6809)** | | | | | | **73.92 (0.7376)** | | | | | |
| 68.1 | 93.8 | | 45.3 | | 71 | 96.6 | | 41.7 | | 51.4 | | 92.1 | | 75.5 | | | 68.2 | | 56.6 | | 49.1 | | 78.5 | | 88.1 | | 52.6 | |
| Boundary Class | 80.04 (0.7979) | | | | | 84.94 (0.8469) | | | | | 70.95 (0.707) | | | | | | | 62.53 (0.6228) | | | | | | 65.02 (0.6477) | | | | | |
| 57 | 95 | 40 | | 57 | 57 | 97 | 33 | | 53 | 51 | 92 | | 69 | | 46 | 69 | | 30 | | 64 | | 48 | 70 | 87 | | 35 | | 45 |

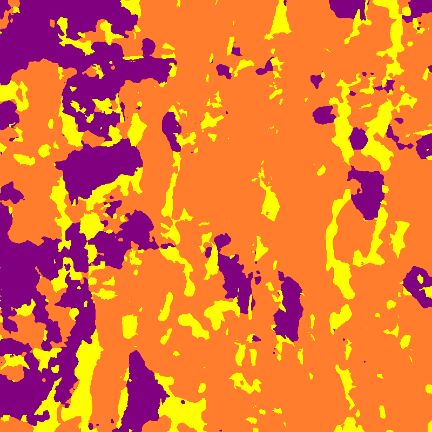
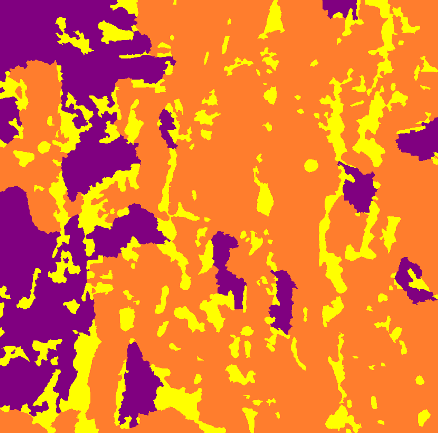
## Results on Inference Images

In this section, we present the results obtained from various types of inference images. We categorize the inference scenarios into three sets: inferencing with the same steel type and magnification as the trained model, inferencing with the same steel type but different magnification levels, and inferencing with different steel types and different magnification levels compared to the trained model.

### Inference on Same Steel and Same Magnification

We evaluated the trained model using a test image that was kept isolated during the training process. The model achieved an overall accuracy of 91.58% with a mean Intersection over Union (mIoU) of 0.46825. Table 4 presents the accuracy and Dice score of the predicted image, as well as the accuracy per class.

A close-up of a grey surface

Description automatically generated with low confidence

|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |

Fig. 6. (a) x2700 magnified E type SEM test image (b) Corresponding label image by EBSD (c) Image predicted by our model.

Analyzing the accuracy per class, we observed that the model achieved an accuracy of 83.7% for Martensite, 96.1% for Ferrite, and 81.3% for Bainite. These results indicate that the model performed relatively well in accurately predicting the presence of Ferrite, Martensite and Bainite.

|  |  |  |  |
| --- | --- | --- | --- |
| Table 6 - Showing accuracy and dice measurements of test image | | | |
| Image Type | Test Image | | |
| Accuracy | 91.58 (0.9174) | | |
| 83.7 | 96.1 | 81.3 |

### Inference on Same Steel with Different Magnification

We conducted inference on images of the same Steel type (E type) but captured at different magnification levels. Specifically, we examined images captured at x3000 and x5000 zoom levels.

A close-up of a grey surface

Description automatically generated with low confidenceA picture containing yellow, colorfulness, orange, amber

Description automatically generatedA picture containing colorfulness, yellow, orange, amber

Description automatically generated

|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |

A close-up of a grey surface

Description automatically generated with low confidenceA picture containing colorfulness, yellow, map, amber

Description automatically generatedA yellow and purple spots on an orange background

Description automatically generated with low confidence

|  |  |  |
| --- | --- | --- |
| (d) | (e) | (f) |

Fig. 7. (a), (b) x3000 magnified E type SEM test image and corresponding label; (d), (e) x5000 magnified E type SEM test image and corresponding label; (c), (f) Images predicted by our model.

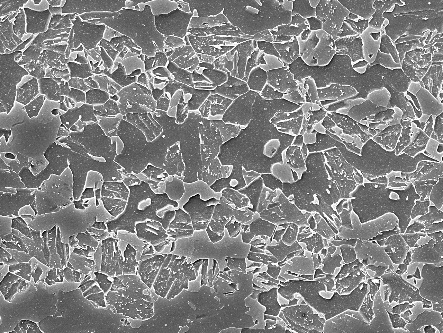
The results of inference on the x3000 zoom images revealed an accuracy of 84.11%. Similarly, for the x5000 zoom images, the accuracy achieved was 89.3%. To further assess the performance, we calculated additional evaluation metrics including the Dice score, accuracy per class, and class ratio. We also calculated Dice score, accuracy per class, class ratio which is ratio of the components (Martensite, Ferrite and Bainite) in the image as evaluating measures which provided information on the relative proportions of these components within the image as can be seen in Table 7.

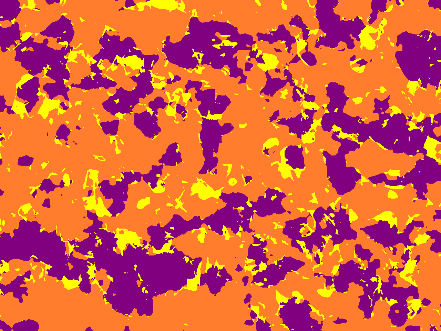
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 7 -** Showing different metrics of predicted images of x3000 and x5000 E type steel | | | | | | |
| Image Type  Metrics | **X3000 E Type** | | | **X5000 E Type** | | |
| Accuracy | 84.11 | | | 89.29 | | |
| Dice Score | 0.8377 | | | 0.8852 | | |
| Accuracy Per Class | 68.11 | 93.77 | 45.33 | 70.87 | 96.57 | 41.73 |
| Class Ratio [label] | 0.1982 | 0.7029 | 0.0989 | 0.176 | 0.7789 | 0.0451 |
| Class Ratio [Predicted] | 0.1514 | 0.7904 | 0.0583 | 0.1282 | 0.8196 | 0.0522 |
| Error Margin of Class Ratio | -0.0438 | +0.0875 | -0.0406 | -0.0478 | +0.0407 | +0.0071 |

### Inference on Different Steel and Different Magnification

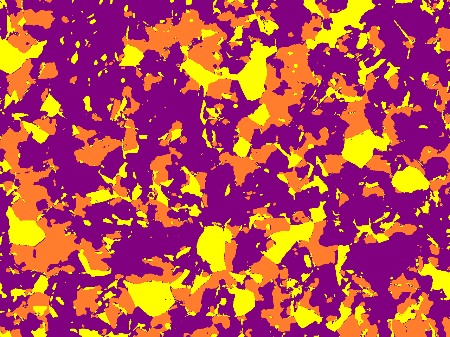
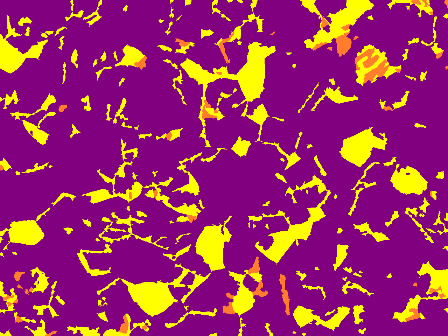
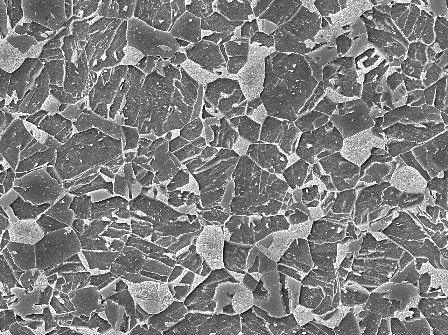
We explored the performance of our model on images of different steel types and magnification levels that were not included in the training data. This allowed us to assess the generalization capability of our model.

The results indicated that the model achieved an accuracy of 74.78% on the A type steel images, 68.3% on the D3 type steel images, and nearly 74% on the H2 type steel images. It is important to note that these images presented challenges to the model as they contained different microstructures and compositions compared to the training data. Despite these challenges, our model demonstrated decent accuracy on these unseen images, indicating its ability to generalize well. Particularly, the D3 type image posed a difficult problem due to its extreme composition and the presence of very similar microstructures. However, the model still performed fairly-well on this image, considering that such cases were not encountered during the training phase. These findings emphasize the robustness of our model in handling images of different steel types and magnification levels, reinforcing its potential for real-world applications where variations in steel composition and image quality may occur.

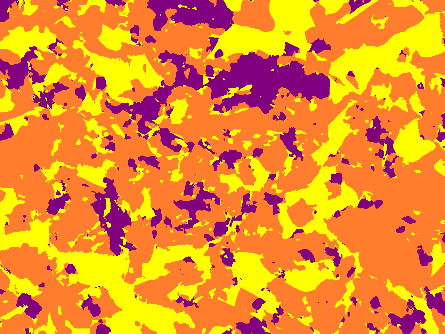
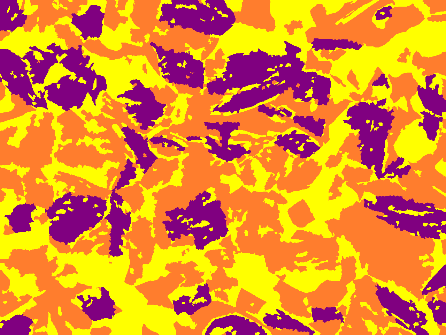
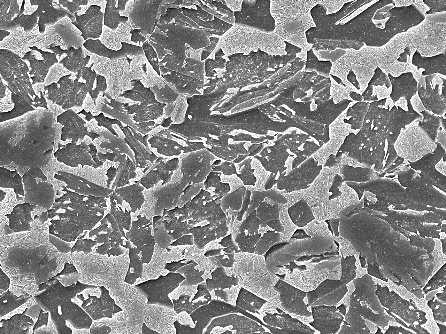
A picture containing yellow, colorfulness, art

Description automatically generated

|  |  |  |
| --- | --- | --- |
| (a) | (b) | (c) |



|  |  |  |
| --- | --- | --- |
| (d) | (e) | (f) |



|  |  |  |
| --- | --- | --- |
| (g) | (h) | (i) |

Fig. 8. (a), (b) x5000 magnified A type SEM test image and corresponding label; (d), (e) x5000 magnified D3 type SEM test image and corresponding label; (g), (h) x5000 magnified H2 type SEM test image and corresponding label; (c), (f) and(i) Images predicted by our model.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 8 -** Showing different metrics of predicted images of different types of steel | | | | | | | | | |
| Image Type  Metrics | **A type** | | | **D3 Type** | | | **H2 Type** | | |
| Pixel Accuracy | 74.78 | | | 68.32 | | | 73.89 | | |
| Dice Score | 0.7365 | | | 0.6791 | | | 0.7248 | | |
| Accuracy Per Class | 51.04 | 92.11 | 65.45 | 79.64 | 24.41 | 66.89 | 78.52 | 88.11 | 37.62 |
| Class Ratio [label] | 0.1441 | 0.4281 | 0.4278 | 0.1742 | 0.0184 | 0.8074 | 0.3404 | 0.4313 | 0.2283 |
| Class Ratio [Predicted] | 0.1 | 0.5683 | 0.3317 | 0.1796 | 0.236 | 0.5844 | 0.2819 | 0.573 | 0.1451 |
| Error Margin of Class Ratio | -0.0441 | 0.1402 | -0.0961 | 0.0054 | 0.2176 | -0.223 | -0.0585 | 0.1417 | -0.0832 |

# Discussion

The experimental results provided valuable insights into various aspects of our proposed approach for steel image segmentation. In this section, we will analyze and interpret the results, discuss the strengths and limitations of our approach, explore factors influencing performance variations, address encountered challenges, and outline potential applications, future directions, and areas for further improvement.

## Experimental Results Analysis and Interpretation

The experiments conducted in this study covered different aspects of the model, including activation functions, blur pooling, loss functions, and the inclusion of a boundary class. The results demonstrated that ReLU activation function outperformed Siren, LeakyReLU, and Swish activation functions, contrary to initial expectations. Blur Pooling showed promising results when combined with the Siren activation function but performed poorly with ReLU. Jaccard Loss outperformed Focal Loss in terms of accuracy and segmentation. Also, the inclusion of a boundary class did not yield significant improvements in the segmentation performance. The experimental results and comparative analysis revealed that the combination of the ReLU activation function and Jaccard loss yielded the best results in terms of accurate steel image segmentation.

## Challenges and Limitations

The limited training dataset, with only six images, poses a significant limitation. The small dataset may not fully represent the diversity and complexity of real-world steel images, potentially impacting the generalization capabilities of the model. There are many instances where a microstructure in the steel images might look similar but have different compositions. In such cases it is a difficult job for the model to find such intricate details in the images.

## Factors Contributing to Performance Variations

The observed performance variations can be attributed to several factors. The choice of activation function significantly influenced the model's ability to capture relevant features and achieve accurate segmentation. The interaction between activation functions and blur pooling demonstrated the importance of considering their compatibility. The selection of an appropriate loss function played a vital role in guiding the model's training process and achieving accurate segmentation results. The inclusion of a boundary class did not yield significant improvements, suggesting that the boundary information alone may not be sufficient for enhanced segmentation performance.

# Conclusion

This research aimed to address the challenge of accurate microstructure segmentation in metallographic images using a deep learning-based approach. Through extensive experiments and analysis, we have made several key findings and contributions. First, we investigated different aspects of the proposed approach, including activation functions, loss functions, and the effects of blur pooling. Our experiments revealed that ReLU activation function and Jaccard loss yielded the optimal results. Additionally, we observed that blur pooling showed improvements with certain activation functions but performed poorly with ReLU. Furthermore, we conducted inference on various types of images to evaluate the generalization capability of our model. The results demonstrated that the model achieved high accuracy on images of the same steel and magnification levels as the trained model, as well as on images of the same steel with different magnification levels. Moreover, the model exhibited reasonable accuracy even on images of different steel types and magnification levels, showcasing its ability to generalize well.

Accurate microstructure segmentation in metallographic images holds significant importance in various industrial applications, such as quality control in steel production and material characterization. Our proposed approach has shown promising results in addressing this task, providing a reliable and automated solution that can significantly reduce human effort and enhance efficiency in analyzing metallographic images. The effectiveness of our approach, coupled with its potential impact on industrial applications, highlights the practical value of this research. By accurately segmenting microstructures, manufacturers can gain valuable insights into material properties, make informed decisions, and improve overall quality control processes. Looking ahead, there are several avenues for future research and development. Firstly, exploring the use of advanced architectures and techniques, such as attention mechanisms and multi-scale analysis, may further improve the accuracy and robustness of the segmentation model. Additionally, extending the research to include more diverse steel types and investigating the impact of various imaging conditions would enhance the model's applicability in real-world scenarios.

# References

1. Bo-hu Li, Bao-cun Hou, Wen-tao Yu, Xiao-bing Lu, Chun-wei Yang, Applications of artificial intelligence in intelligent manufacturing: a review, Front. Inf. Technol. Electron. Eng. 18 (1) (2017) 86–96.
2. Ray Y. Zhong, Xun Xu, Eberhard Klotz, Stephen T. Newman, Intelligent manufacturing in the context of industry 4.0: a review, Engineering 3 (5) (2017) 616–630.
3. Joon B. Park, Roderic S. Lakes, Characterization of Materials — I, Springer New York, New York, NY, 2007, pp. 41–81.
4. Horacio Espinosa, Leonardo Pagnotta, Maria Pantano, Mechanical characterization of materials at small length scales, J. Mech. Sci. Technol. 26 (2012) 545–561.
5. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer Science & Business Media, 2010.
6. Dzung L. Pham, Chenyang Xu, Jerry L. Prince, Current methods in medical image segmentation, Annu. Rev. Biomed. Eng. 2 (1) (2000) 315–337.
7. Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, Paul Kennedy, Deep learning techniques for medical image segmentation: achievements and challenges, J. Digit. Imaging 32 (4) (2019) 582–596.
8. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.
9. Berthold Horn, Berthold Klaus, Paul Horn, Robot Vision, MIT Press, 1986.
10. Marta Fernández-Moreno, Bo Lei, Elizabeth A. Holm, Pablo Mesejo, & Raúl Moreno (2023). Exploring the trade-off between performance and annotation complexity in semantic segmentation. Engineering Applications of Artificial Intelligence, 123, 106299.
11. Julián Luengo, Raúl Moreno, Iván Sevillano, David Charte, Adrián Peláez-Vegas, Marta Fernández-Moreno, Pablo Mesejo, & Francisco Herrera (2022). A tutorial on the segmentation of metallographic images: Taxonomy, new MetalDAM dataset, deep learning-based ensemble model, experimental analysis and challenges. Information Fusion, 78, 232-253.
12. Suresh, S. K.(2019). Machine Learning Based Ultra High Carbon Steel Image Segmentation. (Master's thesis). Retrieved from <https://scholarcommons.sc.edu/etd/5519>
13. Velichko A (2020) Neural network for low-memory IoT devices and MNIST image recognition using kernels based on logistic map. Electronics 9(9):1432
14. Yu-Kun Lai, Shi-Min Hu, Ralph R. Martin, & Paul L. Rosin (2009). Rapid and effective segmentation of 3D models using random walks. Computer Aided Geometric Design, 26(6), 665-679.
15. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems. Curran Associates, Inc..
16. Karen Simonyan, Andrea Vedaldi, & Andrew Zisserman. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.
17. Jonathan Long, Evan Shelhamer, & Trevor Darrell. (2015). Fully Convolutional Networks for Semantic Segmentation.
18. Everingham, M., Van~Gool, L., Williams, C., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303–338.
19. Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2015). The Cityscapes Dataset. In CVPR Workshop on The Future of Datasets in Vision.
20. Olaf Ronneberger, Philipp Fischer, & Thomas Brox. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.
21. Reza Azad, Ehsan Khodapanah Aghdam, Amelie Rauland, Yiwei Jia, Atlas Haddadi Avval, Afshin Bozorgpour, Sanaz Karimĳafarbigloo, Joseph Paul Cohen, Ehsan Adeli, & Dorit Merhof. (2022). Medical Image Segmentation Review: The success of U-Net.
22. Michael, J. (2002). Electron Backscatter Diffraction in the SEM: A Tutorial. Microscopy and Microanalysis, 8(S02), 724–725.
23. Mingle Xu, Sook Yoon, Alvaro Fuentes, & Dong Sun Park (2023). A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. Pattern Recognition, 137, 109347.
24. Diederik P. Kingma, & Jimmy Ba. (2017). Adam: A Method for Stochastic Optimization.
25. Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, & Gordon Wetzstein. (2020). Implicit Neural Representations with Periodic Activation Functions.
26. Thi Le, P., Pham, T., Hsu, Y.C., & Wang, J.C. (2022). Convolutional Blur Attention Network for Cell Nuclei Segmentation. Sensors, 22(4).
27. Dimitrios Marmanis, Konrad Schindler, Jan Dirk Wegner, Silvano Galliani, Mihai Datcu, & Uwe Stilla. (2017). Classification With an Edge: Improving Semantic Image Segmentation with Boundary Detection.
28. Roberts, G., Haile, S.Y., Sainju, R. et al. Deep Learning for Semantic Segmentation of Defects in Advanced STEM Images of Steels. Sci Rep 9, 12744 (2019).