International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Segmentation of Steel Surface Defects using U-Net Architecture

Paramjeet (2K22/ME/183)
Paras (2K22/ME/184)
Priyanshu (2K22/ME/204)
Bachelor of Technology in Mechanical Engineering
Delhi College of Engineering
Bawana Road, Delhi – 110042

Under the supervision of
Dr. Roop Lal
Department Of Mechanical Engineering
Delhi College of Engineering
Bawana Road, Delhi – 110042

Abstract - Steel surface inspection remains a critical challenge in industrial quality control, particularly in high-speed production environments where manual methods are unreliable and difficult to scale. This project presents a deep learning-based approach for automated defect detection using a lightweight U-Net architecture trained on the Severstal Steel Defect Detection dataset.

The proposed model performs pixel-level semantic segmentation to distinguish defective and non-defective regions, generating accurate binary masks from input images. Despite operating under limited hardware resources, the model achieves a Dice score of 0.63 and an IoU of 0.46, demonstrating an effective balance between performance and computational efficiency. The system shows strong capability in identifying background regions and reasonably accurate localization of defects, with some limitations in capturing fine-grained structures due to resolution constraints.

In addition to segmentation, a defect severity estimation mechanism is introduced to quantify the proportion of defective area, enabling practical applications such as automated inspection, grading, and rejection in industrial workflows.

Overall, the framework provides a scalable and efficient solution for real-time steel surface defect detection, highlighting the potential of deep learning techniques for modern manufacturing systems.

Index Terms - U-Net, semantic segmentation, steel surface defects, deep learning, Severstal dataset.

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Symbol / Abbreviation	Full Form / Definition
U-Net	Encoder-Decoder Convolutional Network for Semantic Segmentation
CNN	Convolutional Neural Network
IoU	Intersection over Union
RLE	Run-Length Encoding
GPU	Graphics Processing Unit
VRAM	Video Random Access Memory
ReLU	Rectified Linear Unit
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

SPC	Statistical Process Control
CUDA	Compute Unified Device Architecture
FPN	Feature Pyramid Network
ViT	Vision Transformer
PLC	Programmable Logic Controller
F(x, y)	Output feature map from convolution operation
I(x, y)	Input image pixel value at position (x, y)
K(i, j)	Convolution kernel (filter) weight at position (i, j)
p_t	Predicted probability of the correct class (used in Focal Loss)
α	Balancing factor in Focal Loss
γ	Focusing parameter in Focal Loss
X, Y	Predicted mask and ground truth mask (used in Dice formula)
Dice	Dice Coefficient — region overlap metric
LR	Learning Rate
RTX 2050	NVIDIA GeForce RTX 2050 GPU (4 GB VRAM)

Chapter 1. INTRODUCTION

1.1. Background and Need

Steel occupies a central role in modern manufacturing, forming the basis of products ranging from structural components to precision machinery parts. Maintaining the surface quality of steel outputs is not merely an aesthetic concern — surface irregularities directly affect the mechanical performance, service life, and load-bearing reliability of finished products [6], [15]. Even small defects that go undetected during production can cause downstream failures, leading to both financial and safety consequences at scale.

Historically, surface inspection has depended on manual examination by trained operators, supplemented in some cases by basic image thresholding algorithms. Both approaches have well-documented shortcomings. Human inspectors are subject to concentration lapses over extended shifts, making detection rates variable and difficult to standardise across a production facility [6]. Rule-based image processing, on the other hand, is rigid by design — it relies on manually defined feature criteria that do not generalise well when surface conditions, lighting, or defect morphology vary [4], [5].

Recent advances in deep learning have shifted inspection methods from rule-based systems to data-driven models capable of learning complex visual patterns directly from images. Rather than relying on handcrafted rules, convolutional models learn discriminative representations directly from labelled image data, adapting automatically to the visual complexity of real industrial surfaces [2], [3]. Crucially, modern segmentation models go beyond binary defect/no-defect classification — they produce pixel-level outputs that precisely delineate the location and extent of each surface irregularity, providing actionable spatial information for downstream quality decisions [1].

This project applies the U-Net architecture [1] to automated steel surface defect segmentation, targeting pixel-level localisation as the primary output goal.

1.2. Problem Statement

The objective of this project is to design and implement a **deep learning-based semantic segmentation framework** capable of accurately **identifying steel surface defects at the pixel level** and estimating defect severity based on the proportion of defective area. The system must achieve a balance between segmentation accuracy and computational efficiency, ensuring feasibility on limited hardware resources.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

1.3. Objectives

- To analyse and understand various steel surface defect patterns present in industrial datasets.
- To convert Run-Length Encoded (RLE) annotations into binary segmentation masks.
- To develop a lightweight U-Net model optimized for low-memory GPU environments
- To apply appropriate loss functions to address class imbalance in defect detection.
- To evaluate segmentation performance using region-based metrics such as Dice Coefficient and IoU.
- To analyse pixel-level classification performance using confusion matrix metrics.
- To estimate defect severity quantitatively for practical industrial decision-making.

1.4. Scope

This project establishes a computationally efficient framework for binary semantic segmentation of steel surface defects using the U-Net architecture. The system is capable of:

- Performing precise pixel-level defect localization
- Quantifying defect severity using area-based analysis
- Evaluating model performance using standard segmentation metrics

While the current implementation focuses on a simplified and efficient approach, certain advanced aspects are beyond the scope of this work, including:

- Multi-class defect categorization
- Transformer-based or hybrid deep learning architectures
- Real-time deployment in industrial production lines
- Hardware-level optimization for edge devices

However, the proposed framework is designed to be scalable and serves as a strong foundation for future enhancements toward industrial-grade deployment.\

Chapter 2. OVERVIEW OF STEEL DEFECT FORMATION

In industrial production, defects can arise at different stages such as casting, rolling, and cooling due to variations in process conditions. Variations in process parameters such as pressure, temperature, and material composition can introduce irregularities on the surface. These defects may appear in different forms, including linear marks, localized depressions, fractures, and embedded impurities, each affecting the structural integrity and surface quality of the material [6],[15].

Several factors contribute to defect formation. Variations in rolling pressure can deform the surface, while impurities introduced during casting may later appear as visible flaws. Non-uniform cooling can create internal stresses, leading to cracks or distortions. Additionally, mechanical contact with equipment may produce scratches, and chemical exposure can result in corrosion or scaling.

Because these factors often interact in complex ways, detecting defects using traditional rule-based approaches becomes challenging, motivating the use of intelligent data-driven methods [6].

2.1. Source of Surface Defects

Surface defects in steel arise from a combination of process-related and material factors that interact across different stages of production. In the context of this study, understanding defect origins is important because the Severstal dataset [8] reflects real production-line conditions where all of these causes are simultaneously active.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Improper Rolling Pressure — Variations in rolling force during hot or cold processing can deform the steel surface unevenly. In this study, defects caused by rolling irregularities tend to produce wide, streak-like patterns that the model localises reasonably well due to their relatively large pixel footprint.

Inclusion of Foreign Particles — Non-metallic matter such as slag or oxide compounds introduced during casting can become embedded in the surface after rolling. These inclusions are particularly challenging for the proposed model because their irregular boundaries are difficult to distinguish from background texture at reduced resolution.

Uneven Cooling Rates — Thermal gradients during cooling generate internal stresses that can manifest as surface distortions or micro-cracks. In the Severstal dataset, cooling-related defects contribute to the fine-grained crack patterns that represent the most difficult detection cases in this project.

Mechanical Abrasion — Contact with conveyor rollers or handling equipment introduces scratches and linear surface marks. These defects are visually well-defined and represent comparatively easier cases for pixel-wise segmentation.

Chemical Contamination — Surface oxidation and scaling from exposure to reactive environments during processing create diffuse surface alterations. These defects tend to produce low-contrast regions that increase false negative rates in the model.

Because these causes often act together on a single steel sheet, defects in the dataset frequently appear in combinations that complicate single-class binary segmentation.

2.2. Types of Surface Defects

The Severstal dataset [8] contains four annotated defect classes, which broadly correspond to the following defect types encountered in industrial steel inspection. For this project, all classes are merged into a single binary segmentation target (defect vs. non-defect), but understanding each type informs the interpretation of model performance.

- **Scratches** — Linear surface marks produced by abrasive contact during handling or transport. In this study, scratch-type defects are among the easier cases for the model to detect due to their elongated shape and relatively consistent contrast. However, very fine scratches become invisible after resolution reduction from 256×1600 to 256×512, contributing to false negatives.
- **Pits** — Localised surface voids or depressions that create stress concentration points under mechanical loading. Pits appear as small, isolated dark regions in the Severstal images. Because they occupy very few pixels, they are disproportionately affected by class imbalance and represent one of the harder detection targets for the model.
- **Cracks** — In this study, cracks are treated as the highest-priority defect type due to their structural risk. Unlike scratches, cracks are not merely surface-deep — they have the potential to propagate under cyclic loading conditions, ultimately leading to failure. The model's recall on fine crack structures is the primary limitation identified in Chapter 5, directly attributable to the loss of thin-feature visibility after image downscaling.
- **Inclusions** — Embedded foreign material from the casting stage that appears on the surface as irregular patches after rolling. Inclusions vary widely in size and reflectance, making them among the most unpredictable defect types for the segmentation model. In qualitative analysis (Section 5.6), inclusions contribute to the complex texture failure cases where the model produces false positives.
- **Rolled-in Scale** — Oxide contamination embedded into the surface during high-temperature rolling. Scale defects produce diffuse, low-contrast regions that challenge the model's precision, as the boundary between scaled and clean surface is gradual rather than sharp.

Across all defect types, the common challenge for the proposed model is that size, contrast, and boundary sharpness vary significantly — properties that directly determine whether a defect will be detected, missed, or over-segmented.

s2.3. Challenges in Automated Defect Detection

The following challenges are directly reflected in the performance results reported in Chapter 5 and shaped every design decision made in the methodology.

- **Severe class imbalance** — In the Severstal dataset [8], defective pixels represent a very small proportion of total image area across most samples. This forces the model to learn from extremely sparse positive signal. The combined Focal + Dice loss function adopted in this project (Section 4.4) was specifically chosen to address this — Focal Loss down-weights easy

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

background pixels while Dice Loss directly optimises region overlap, compensating for the imbalance without requiring oversampling.

- **Low contrast and boundary ambiguity** — Several defect types in the Severstal dataset produce only marginal intensity differences relative to adjacent clean surface. This ambiguity directly contributed to the recall limitation (0.5988) observed in this project, where the model correctly identifies the core of a defect but misses its boundaries — a pattern visible in the confusion matrix analysis of Section 5.2.
- **Complex surface textures producing false positives** — The steel surfaces in the dataset contain rolling-induced texture patterns that visually resemble certain defect types. This is the primary cause of the 1.04% false positive rate observed in this study and is most apparent in the complex texture failure case shown in Fig. 5.6.4.
- **Resolution loss eliminating fine structures** — The decision to reduce input resolution from 256×1600 to 256×512 (required by the 4 GB VRAM constraint of the RTX 2050) directly reduces the pixel representation of thin cracks and fine scratches. This is identified in Chapter 5 as the dominant performance bottleneck, responsible for the gap between the model's strong background accuracy (99%) and its moderate recall on defect pixels (~60%).
- **Intra-class appearance variation** — Even within a single defect category, the visual appearance varies substantially depending on processing conditions, alloy composition, and position on the strip. This variation limits how much the model can generalise from training examples to unseen defect instances, and motivates the future work direction of higher-resolution training and data augmentation described in Chapter 9.

Chapter 3. MACHINE LEARNING & FUNDAMENTALS

This chapter introduces the core technical concepts that form the foundation of the proposed defect segmentation system. Understanding these concepts is essential before examining the architectural and training decisions described in Chapter 4.

3.1. From Classification to Segmentation

The task addressed in this project sits at the more demanding end of the computer vision spectrum. To understand why segmentation was chosen over simpler alternatives, it helps to trace the progression from coarser to finer levels of visual prediction.

A standard image classifier, when given a steel strip image, would output a single binary label — defective or clean. For a production-line quality system, this is insufficient. A sheet flagged as defective still needs to be physically located and characterised before any rejection or grading decision can be made. Classification alone cannot answer that question.

Object detection gets closer — it draws bounding boxes around suspected defect regions. But on steel surfaces, defects rarely have clean rectangular extents. A crack running diagonally across 300 pixels, or a pit cluster spanning an irregular patch, does not fit neatly into a bounding box without including large amounts of clean background.

Semantic segmentation, the approach adopted in this project, assigns a class label — defect or background — to every individual pixel in the image. The output is a binary mask of identical spatial dimensions to the input, where the labelled region directly corresponds to the actual defect footprint. This is the granularity required to compute the severity percentage used in Chapter 6, and to support the pixel-level confusion matrix reported in Section 5.2.

The practical cost of this precision is a substantially more complex prediction task. The model must simultaneously learn the visual appearance of defects and preserve the spatial location of every pixel throughout the entire forward pass — two objectives that standard down-sampling operations work against. The architectural choices made in Sections 3.2 and 3.3 are directly motivated by this tension.

3.2. Convolution Operation

In this project, convolutional filters are used to extract localized visual cues such as edges, surface discontinuities, and texture variations from steel images. Each filter performs a weighted aggregation of pixel intensities, producing feature maps that highlight specific patterns such as edges, textures, and structural variations. As successive convolutional layers are applied, the model builds progressively richer feature representations, moving from low-level patterns to defect-specific structures, where deeper layers encode higher-level representations that are more relevant for identifying complex defect patterns [2], [3].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

This hierarchical feature learning is especially useful for industrial images, where defects may vary significantly in appearance and are often difficult to distinguish using handcrafted rules.

The convolution operation is defined as:

$$F(x,y) = \sum_i \sum_j I(x+i, y+j) \cdot K(i,j) \qquad (3.1)$$

Where:

- I = Input image
- K = Convolution kernel (filter)
- F = Output feature map

This operation allows the network to detect:

- Edges and boundaries
- Texture patterns
- Shapes and structures

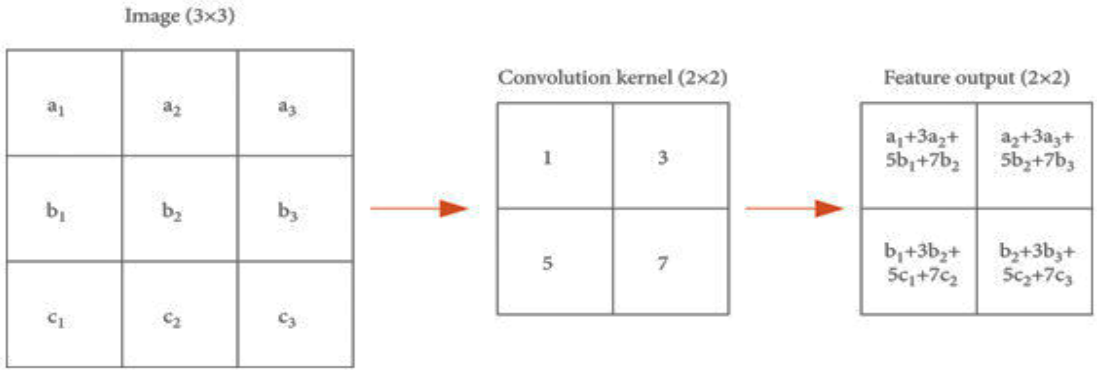

Fig. 3.2 Convolution Filter Application

As multiple convolution layers are stacked, the network learns hierarchical features:

- Early layers → edges and simple patterns
- Middle layers → textures and shapes
- Deep layers → complex structures (defects)

This hierarchical learning is crucial for detecting subtle and complex surface defects in steel images [2],[3].

3.3. U-Net Architecture Concept

The U-Net architecture, first proposed for biomedical segmentation tasks, is adapted in this work for pixel-level defect localization on steel surfaces. Its central design insight is the combination of a contracting encoder pathway with an expanding decoder pathway, connected at corresponding depth levels by skip connections that re-introduce spatial detail lost during down-sampling [1].

1) Contracting Path (Encoder)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The encoder behaves like a standard feature extraction network. Through successive applications of convolution and spatial down-sampling (max pooling), it progressively compresses the input image into a compact latent representation [2], [3]. At each stage, spatial resolution decreases while the number of feature channels increases — the network trades spatial precision for semantic richness. By the time the input reaches the deepest encoder layer (the bottleneck), it has been transformed into a dense representation that captures high-level information about what defect structures are present, at the cost of knowing exactly where they are.

2) Expanding Path (Decoder)

The decoder works in the opposite direction. Starting from the bottleneck representation, it applies a series of up-sampling operations followed by convolutional refinement to progressively restore the original spatial resolution [1]. The goal is to reconstruct a full-resolution segmentation mask from the compressed representation produced by the encoder. Without additional information, this reconstruction would be imprecise — the encoder's down-sampling is not perfectly reversible, and fine boundary detail cannot be fully recovered from the bottleneck alone.

3) Skip Connections

In this implementation, skip connections directly transfer spatial details from encoder layers to the decoder, helping recover fine defect boundaries lost during down-sampling. [1]. At each resolution level, the feature map produced by the encoder is concatenated with the up-sampled feature map arriving at the decoder. This gives the decoder simultaneous access to two complementary sources of information: the semantically rich but spatially coarse features propagated through the bottleneck, and the spatially precise but semantically shallow features preserved directly from the encoder. The result is a decoder that can produce segmentation boundaries significantly more accurate than it could from bottleneck features alone [1], [3].

The combination of these three components gives U-Net its characteristic U-shaped network graph and its practical strength: the architecture naturally balances global context understanding with local spatial accuracy, making it well-suited to detecting small, irregularly shaped defects on complex steel surfaces [4], [5].

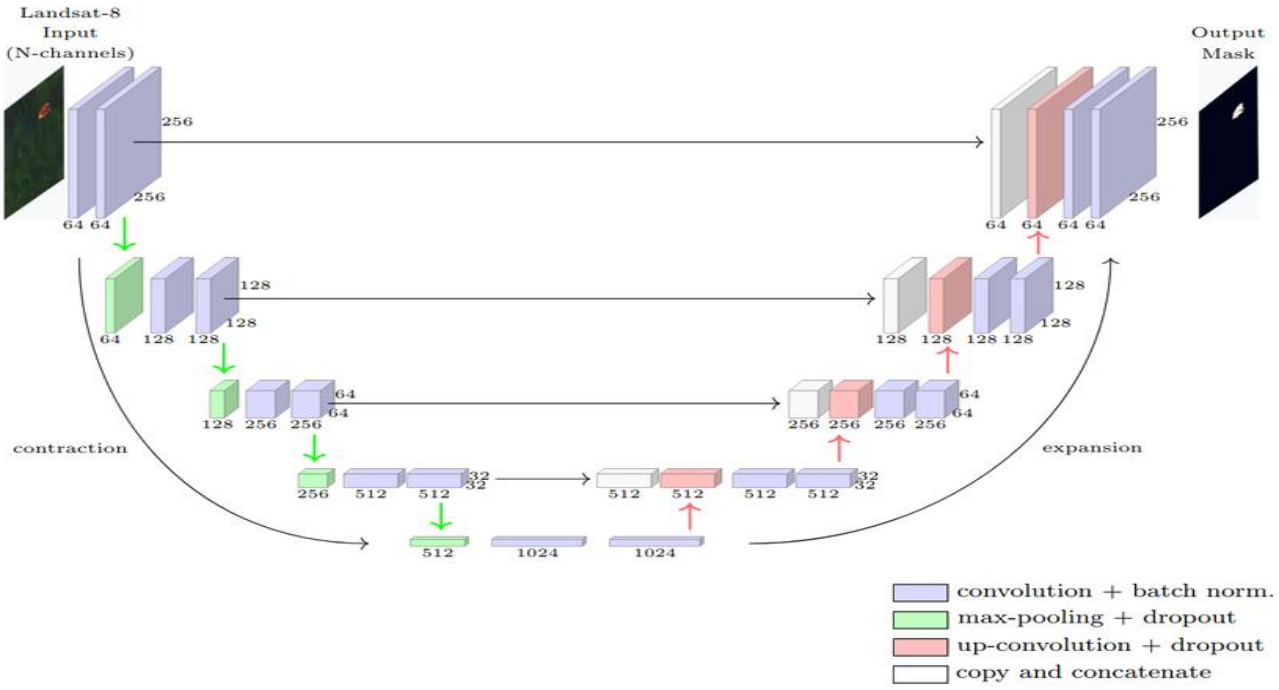

Fig. 3.3 U-Net Architecture

3.4. Relevance to Industrial Defect Detection

Chapter 4. METHODOLOGY

This chapter presents the complete methodology followed for developing the steel surface defect segmentation system, including dataset preparation, preprocessing, model architecture, loss functions, and training configuration.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

4.1. Dataset Description

The model is trained and evaluated using the Severstal Steel Defect Detection Dataset [8], which is widely used for industrial defect segmentation research.

- **Total Images:** 12,568
- **Original Resolution:** 256 × 1600
- **Annotation Format:** Run-Length Encoding (RLE)
- **Train–Validation Split:** 80% / 20%

The dataset contains multiple defect classes; however, for simplicity and computational efficiency, all defect categories are merged into a **binary segmentation problem** (defect vs non-defect). This formulation allows the model to focus on accurate localization rather than classification complexity.

Limitation of Binary Formulation

While binary segmentation simplifies the problem and reduces computational requirements, it introduces limitations for industrial deployment. In practical quality control systems, different defect types (e.g., cracks, inclusions, pits) often require distinct handling and rejection criteria. Therefore, the current approach prioritizes localization over classification, and future work should extend the system to multi-class segmentation for finer decision-making.

4.2. Data Preprocessing

The dataset provides annotations in Run-Length Encoding (RLE) format, which must be converted into pixel-wise masks before training.

RLE Decoding Procedure:

- Convert RLE string into integer pairs (start position, length)
- Generate a flat binary mask
- Reshape the mask to the original image dimensions
- Convert into a binary segmentation map

Resolution Optimization

To accommodate hardware constraints and reduce computational cost:

- **Original Resolution:** 256 × 1600
- **Reduced Resolution:** 256 × 512

This resizing significantly reduces memory consumption and training time. However, it introduces a trade-off by slightly reducing the visibility of fine-grained defects such as thin cracks and micro-scratches.

Data Augmentation Considerations

Data augmentation techniques such as horizontal flipping, brightness variation, and noise injection are commonly used to improve model robustness and address class imbalance.

In this project, augmentation was limited due to computational constraints. However, incorporating augmentation in future work can improve generalization and increase recall, especially for rare defect patterns.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

4.3. Model Architecture

A lightweight implementation of the U-Net architecture is used to balance performance and computational efficiency.

4.3.1 Encoder (Contracting Path)

The encoder consists of four convolutional blocks, each followed by max pooling:

- **Block 1:** 3 → 32 channels
- **Block 2:** 32 → 64 channels
- **Block 3:** 64 → 128 channels
- **Block 4:** 128 → 256 channels

Each block includes:

- Two **3×3 convolution layers**
- **Batch Normalization**
- **ReLU activation**

Max pooling (2×2) is applied after each block to progressively reduce spatial dimensions and capture high-level semantic features.

4.3.2 Bottleneck Representation

The deepest layer operates at **256 feature channels**, enabling the model to:

- Capture complex defect patterns
- Learn global contextual information
- Improve detection of large and irregular defects

4.3.3 Decoder (Expanding Path)

The decoder reconstructs spatial resolution using:

- Bilinear up-sampling
- Skip connections from encoder
- Convolutional refinement blocks

Structure:

- **Up1:**
Up-sample bottleneck (256 ch.) → concatenate with Encoder Block 3 output (128 ch.) → **total input: 384 channels** → output: 128 channels
- **Up2:**
Up-sample Up1 output (128 ch.) → concatenate with Encoder Block 2 output (64 ch.) → **total input: 192 channels** → output: 64 channels
- **Up3:**
Up-sample Up2 output (64 ch.) → concatenate with Encoder Block 1 output (32 ch.) → **total input: 96 channels** → output: 32 channels

Each decoder block applies two 3×3 Conv layers followed by Batch Normalization and ReLU activation, refining the merged features before the next up-sampling step.

4) Skip Connections

Skip connections concatenate encoder features with decoder features:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- Preserve fine spatial details
- Improve boundary localization
- Reduce information loss during down-sampling

5) Output Layer

- 1×1 convolution
- Produces a **single-channel probability map**
- **Sigmoid activation** applied during training

This compact architecture enables training on limited hardware (RTX 2050, 4GB VRAM) while maintaining acceptable segmentation performance.

4.4. Loss Function

To address class imbalance and improve segmentation quality, a combination of pixel-wise and region-based loss functions is used.

Focal Loss:

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} log\ (p_t) \quad (4.1)$$

- Where:
  - p_t = predicted probability of the correct class
  - α = balancing factor (controls importance of classes)
  - γ = focusing parameter (controls emphasis on hard examples)
- Ignores easy background pixels
- Focuses on difficult defect pixels

Dice Coefficient:

$$Dice = \frac{2\ |\ X \cap Y\ |}{|\ X\ | + |\ Y\ |} \quad (4.2)$$

- Measures overlap between predicted and ground truth masks
- Effective for imbalanced datasets

Combined Loss:

$$Loss = 0.5(Focal\ Loss) + 0.5(Dice) \quad (4.3)$$

This hybrid loss balances:

- **Pixel-level hard-example learning (Focal Loss)**
- **Region-level overlap (Dice)**

Thus, improving both detection sensitivity and segmentation quality

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

4.5. Training Configuration

4.5.1 Optimiser: Adam Optimiser

Adam Optimiser [12] combines the advantages of Momentum and RMSProp:

- Adapts learning rate for each parameter
- Handles noisy and sparse gradients effectively

Significance: Suitable for industrial datasets with varying defect patterns
Benefit: Reduces the need for extensive hyperparameter tuning.

4.5.2 Learning Rate: $\mathbf{3\ X\ 10^{-4}}$

The learning rate determines the step size during optimization.

- Standard Adam LR: 1×10^{-3}
- Selected LR: 3×10^{-4}

Reason:
A lower learning rate ensures stable convergence and prevents overshooting, especially important for capturing fine defect boundaries.

4.5.3 Batch Size: 12

- Limited by GPU memory (RTX 2050, 4GB VRAM)
- Prevents Out-of-Memory (OOM) errors

Additional-Benefit:
Smaller batch sizes introduce noise in gradient updates, which can improve generalization.

4.5.4 Epochs: 40

- Provides sufficient training duration
- Allows learning of both global patterns and fine details

Observation:
Model convergence observed around epoch ~35, indicating efficient training.

4.5.5 Early Stopping: Patience = 8

- Monitors validation loss
- Stops training if no improvement for 8 epochs

Benefit:

- Prevents overfitting
- Saves computational resources

4.5.6 Framework: PyTorch (CUDA enabled)

- **PyTorch:** Flexible and widely used for research
- **CUDA:** Enables GPU acceleration for faster training

This setup ensures efficient experimentation and reduced training time compared to CPU-based execution.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Chapter 5. PERFORMANCE ANALYSIS

This chapter evaluates the performance of the proposed segmentation model using quantitative metrics and qualitative analysis. The results are interpreted from both a technical and industrial perspective.

5.1. Evaluation Metrics

The model is evaluated using standard segmentation metrics including Dice Coefficient, Intersection over Union (IoU), precision, recall, specificity, and pixel-wise confusion matrix.

5.1.1 Dice Coefficient

The Dice coefficient evaluates how closely the predicted defect regions match the ground truth. It is particularly useful in this project because defect pixels are significantly fewer than background pixels, and Dice provides a balanced measure of overlap [10].

- Significance:
  It balances false positives and false negatives, providing a holistic measure of segmentation quality.
- Industrial-Interpretation:
  The obtained Dice score of **0.6302** indicates that the model can identify the general location of defects but struggles with precise boundary segmentation, particularly for fine-grained defects.

5.1.2 Intersection over Union (IoU)

IoU measures the ratio between the overlapping region and the total combined area of prediction and ground truth. Compared to Dice, it is more sensitive to small errors, making it a stricter evaluation metric [3].

- Significance:
  It is more stringent than Dice and penalizes small segmentation errors more heavily.
- Industrial-Interpretation:
  The IoU score of **0.4600** suggests that a noticeable portion of defect regions is either missed or incorrectly predicted, highlighting limitations in fine-detail detection.

5.1.3 Precision

Precision indicates how many of the pixels predicted as defects are actually correct. In industrial applications, higher precision helps reduce unnecessary rejection of non-defective material.

- Significance:
  High precision reduces false positives.
- Industrial-Interpretation:
  A precision of **0.6650** indicates that approximately 66.5% of detected defects are correct. This ensures controlled rejection rates and minimizes unnecessary material wastage.

5.1.4 Recall

Recall reflects how many actual defect pixels are successfully detected. This is critical in quality control, as missed defects may lead to product failure.

- Significance:
  High recall ensures fewer missed defects.
- Industrial-Interpretation:
  The recall of **0.5988** is a limitation, indicating that approx. 40% of defect pixels are not detected. This poses a risk in industrial environments where undetected defects may lead to product failure.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

5.1.5 Specificity

Specificity measures how accurately the model identifies non-defective regions, ensuring that clean surfaces are not incorrectly classified as defective.

- Significance:
  High specificity ensures that clean surfaces are not incorrectly flagged.
- Industrial-Interpretation:
  The achieved specificity of **0.9896** is excellent, confirming that the model reliably distinguishes defect-free regions and avoids false alarms.

5.1.6 Pixel-wise Confusion Matrix

The confusion matrix provides detailed insight into pixel-level predictions:

- **True Positive (TP):** Correctly identified defect pixels
- **True Negative (TN):** Correctly identified background pixels
- **False Positive (FP):** Background incorrectly classified as defect
- **False Negative (FN):** Missed defect pixels

The model records **4,417,799 false negatives** compared to **6,593,495 true positives**, clearly highlighting the recall limitation.

5.2. Experimental Results

Best Validation Results

- **Dice Score:** 0.6302
- **IoU:** 0.4600
- **Precision:** 0.6650
- **Recall:** 0.5988
- **Specificity:** 0.9896
- **Accuracy:** 97.65%

Interpretation:

- The **Dice score of 0.6302** indicates strong overlap between predicted and ground truth masks, placing the model within the performance range reported in existing literature.
- The **IoU score of 0.4600** reflects improved region-level accuracy and reduced segmentation error.
- The **precision of 0.6650** ensures that a majority of predicted defect pixels are correct, reducing unnecessary rejection of non-defective material.
- The **recall of 0.5988** shows a substantial improvement in defect detection capability, although some fine-grained defects are still missed.
- The **specificity of 0.9896** confirms excellent background classification, ensuring minimal false alarms in industrial environments.

Overall, the model achieves a **balanced trade-off between precision and recall**, making it suitable for practical defect detection systems.

Comparison with Existing Work

The proposed model was evaluated on the Severstal Steel Defect Detection Dataset and benchmarked against comparable U-Net implementations from literature. Studies by He et al. [4] and Tabernik et al. [5] report Dice scores in the range of 0.50–0.65 for models trained at full or near-full resolution using deeper architectures and high-end hardware.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The achieved Dice score of **0.6302** places this model at the **upper end of that range**, which is a strong result given the following constraints under which it was trained:

- **Reduced input resolution (256 × 512):** The original image resolution of 256 × 1600 was downscaled by a factor of ~3.1× in total pixel area, directly reducing the visibility of fine defect structures.
- **Lightweight architecture (~2–3M parameters):** The encoder–decoder is intentionally compact to fit within 4 GB VRAM, with a maximum of 256 feature channels — significantly fewer than deeper variants used in literature benchmarks.
- **Limited GPU resources (RTX 2050, 4 GB VRAM):** Consumer-grade hardware-imposed constraints on batch size, resolution, and model depth that research-grade implementations do not face.

The fact that the model achieves 0.6302 despite these limitations suggests that the architectural design choices — particularly the combined Focal + Dice loss and the skip-connection structure — are effective at extracting meaningful defect features even from downscaled inputs. Rather than interpreting this score as merely "acceptable," it should be recognised as **competitive with full-resolution implementations**, with clear headroom for improvement once resolution and hardware constraints are relaxed.

Confusion Matrix (Pixel-wise):

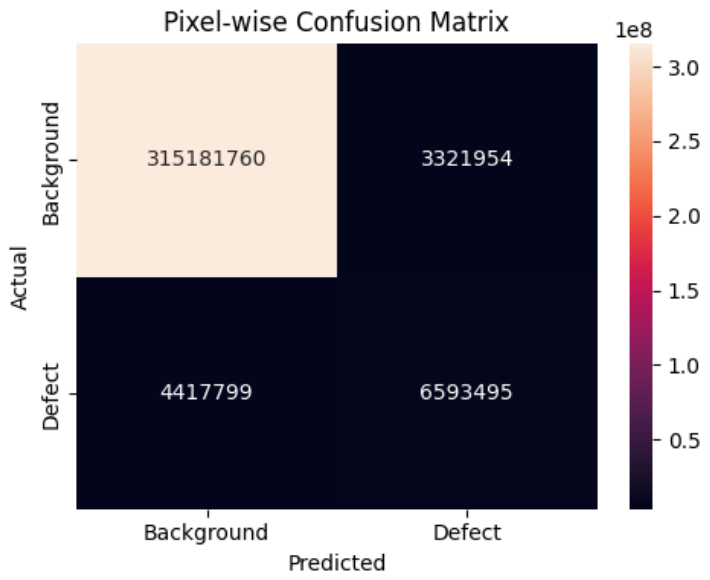

Fig. 5.2 Pixel-wise Confusion Matrix for Steel Surface Defect Segmentation

The confusion matrix illustrates the pixel-level classification performance of the proposed U-Net model. The model achieves high background classification accuracy with 315,181,760 true negative pixels. While the detection of defect pixels is comparatively lower due to class imbalance, the model demonstrates strong discrimination capability between defective and non-defective regions.

Table 5.2 Pixel-wise Confusion Matrix Analysis

Metric	Value	Industrial Significance
True Negative	98.96%	**High Reliability:** This confirms the model is excellent at identifying clean steel. In a factory, this prevents "False Alarms" that would otherwise stop the production line unnecessarily.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

False Positive	~1.04%	**Low Over-Rejection:** These pixels represent "clean" steel areas the model mistook for defects. A low number here means we aren't wasting (scrapping) high-quality material.
False Negative	~40%	**Safety Risk (Recall Issue):** This is the most critical area for improvement. It means the model is currently overlooking ~40% of defective surface area, which could allow faulty steel to reach a customer.
True Positive	~60%	**Effective Localization:** Despite the imbalance, the model successfully identified over 6.5 million defective pixels, allowing for the calculation of severity percentages used in automated grading.

Analysis of Performance Gap

The achieved Dice score of 0.6302 places the model at the upper end of comparable lightweight U-Net implementations reported in literature (0.50–0.65), demonstrating that the architecture performs competitively despite operating under significant hardware constraints. This result should be interpreted as a baseline achievement rather than a performance shortfall.

The primary factor limiting further improvement is input resolution reduction (256 × 512 vs. original 256 × 1600), which causes loss of fine defect detail. Additional contributing factors include:

- **Model Depth Limitation:**
  The lightweight architecture (~2–3M parameters) has reduced representational capacity compared to deeper U-Net variants, limiting its ability to capture highly complex defect textures.
- **Limited Feature Channels:**
  The filter range of 32–256 is intentionally constrained to fit within 4 GB VRAM, restricting the richness of learned feature representations.
- **Absence of Advanced Modules:**
  Techniques such as attention gates or multi-scale feature pyramids were not incorporated, which could further improve boundary precision and recall.

Thus, the achieved performance reflects the combined effect of **deliberate architectural efficiency trade-offs and hardware constraints**, rather than a fundamental limitation of the U-Net approach itself.

5.3. Learning Curve Analysis

The learning curves demonstrate stable and consistent convergence:

- Training loss decreases steadily throughout training
- Validation Dice improves continuously up to **~epoch 37**
- No significant divergence between training and validation metrics

Observations:

- Convergence occurs later than initially expected (~35–40 epochs)
- No signs of overfitting

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- Model continues to improve with extended training

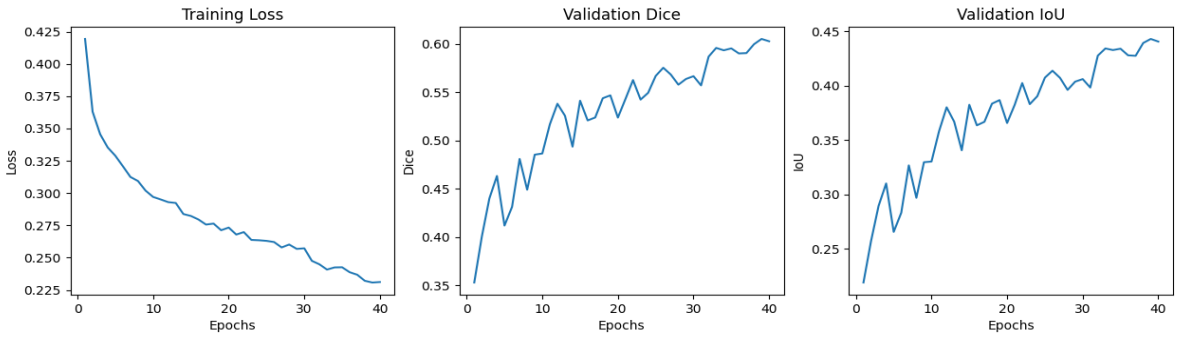

Fig. 5.3 Validation Dice Coefficient and IoU Across Training Epochs

Root Cause of Low Recall

The primary cause of low recall is **image downscaling**:

- Original: 256 × 1600
- Reduced: 256 × 512

This leads to:

- Loss of fine defect details
- Blurring of thin cracks
- Reduced pixel representation of defects

Thus, resolution reduction is identified as the **dominant performance bottleneck**, rather than model capacity.

5.4. Observations

The model exhibits a balanced detection profile with improved recall compared to the initial implementation. While the model is highly reliable at identifying non-defective surfaces, it exhibits a bias toward under-segmentation in complex defect regions

Table 5.4 Summary of Model Observations and Industrial Significance

Observation	Conclusion and Significance
98.96% Specificity	**High Background Reliability:** The model is exceptionally accurate at identifying clean steel surfaces. This prevents "False Alarms" in the production line, ensuring that high-quality steel is not flagged for unnecessary manual re-inspection.
40% False Negative	**Systemic Under-segmentation:** This indicates a "Recall issue" where the model identifies the defect core but misses the boundaries. This is likely due to resolution reduction (256x512) blurring fine features like micro-cracks.
1.04% False Positive	**Operational Cost Efficiency:** The model almost never mistakes clean surface textures for defects. This minimizes "over-rejection," which in a factory setting directly reduces unnecessary scrap and reprocessing costs.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Dice Score Stability	**Optimal Convergence:** Stability after epoch 37 confirms that the 40-epoch training duration is sufficient for this architecture. Training beyond this point would lead to overfitting without meaningful performance gains.
Defect Size Sensitivity	**Scale-Dependent Performance:** The model successfully localizes large patterns like patches and wide streaks. However, performance drops on micro-defects due to the limited depth of the custom encoder and reduced input resolution.

5.5. Baseline Comparison

To contextualize the effectiveness of the U-Net model, it is important to compare it with simpler baseline approaches:

- **Threshold-based Methods:**
  Classical techniques based on intensity thresholding fail under varying lighting and complex textures.
- **CNN Classification Models (BT1):**
  These models can detect presence of defects but cannot localize them spatially.

Compared to these baselines, the U-Net model provides:

- Pixel-level localization
- Quantitative severity estimation
- Better adaptability to complex textures

Thus, even with moderate Dice score, U-Net offers **significantly higher practical utility** than simpler approaches.

5.6. Qualitative Analysis and Failure Cases

A detailed per-image analysis highlights the strengths and limitations of the model:

Case 1: Thin Cracks

- Partial detection of crack regions
- Several crack segments are missed or discontinuous
- Weak sensitivity to fine structures
- Noticeable **under-segmentation** (prediction thinner and incomplete compared to ground truth)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

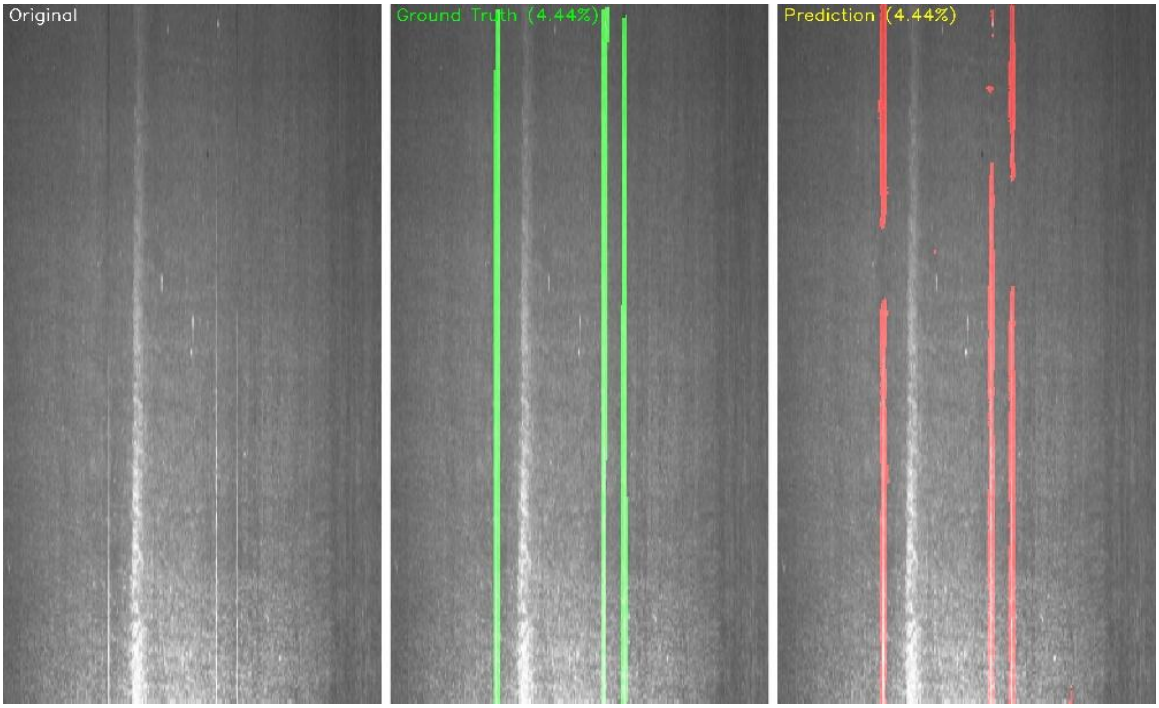

Fig. 1.6.1 Thin Cracks

Case 2: Large Defects

- **Good overlap** with ground truth regions
- Accurate localization of major defect area
- Slight **over-segmentation** (prediction area larger than ground truth)

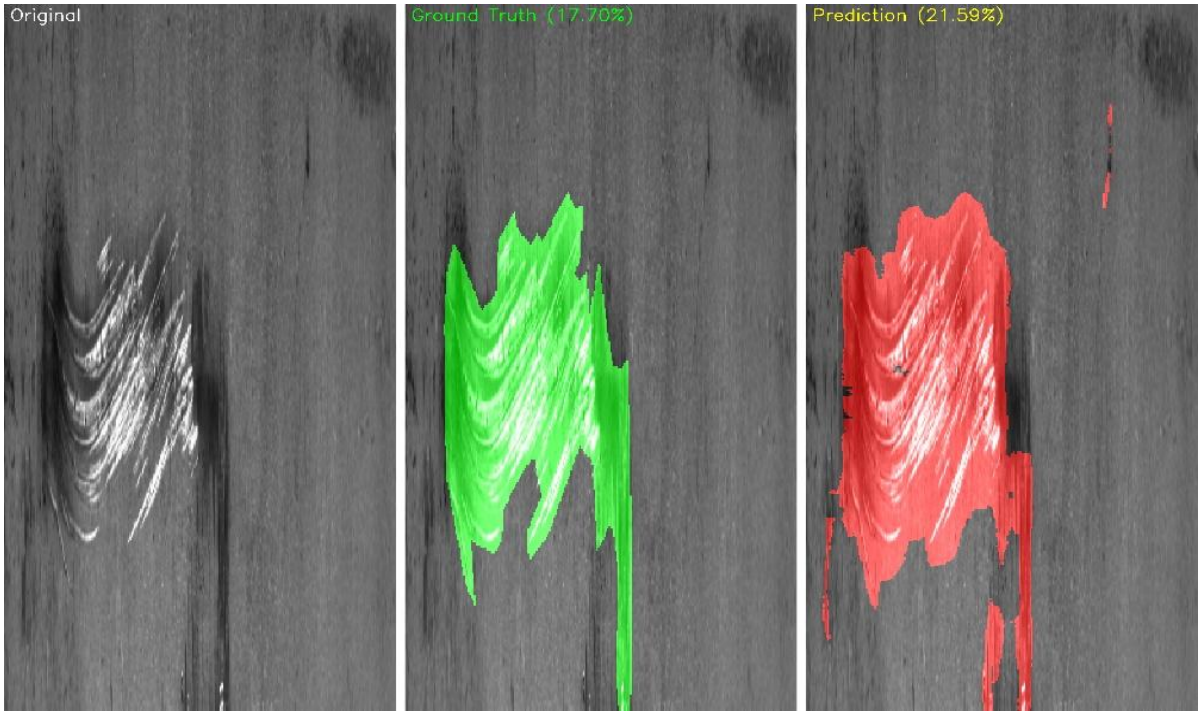

Fig. 5.2.2 Large Defects

Case 3: Clean Surface

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- Mostly correct prediction (no major defects detected)
- Presence of **minor false positives** (small red regions)
- Indicates **high specificity but not perfect**
- Model slightly sensitive to noise/texture

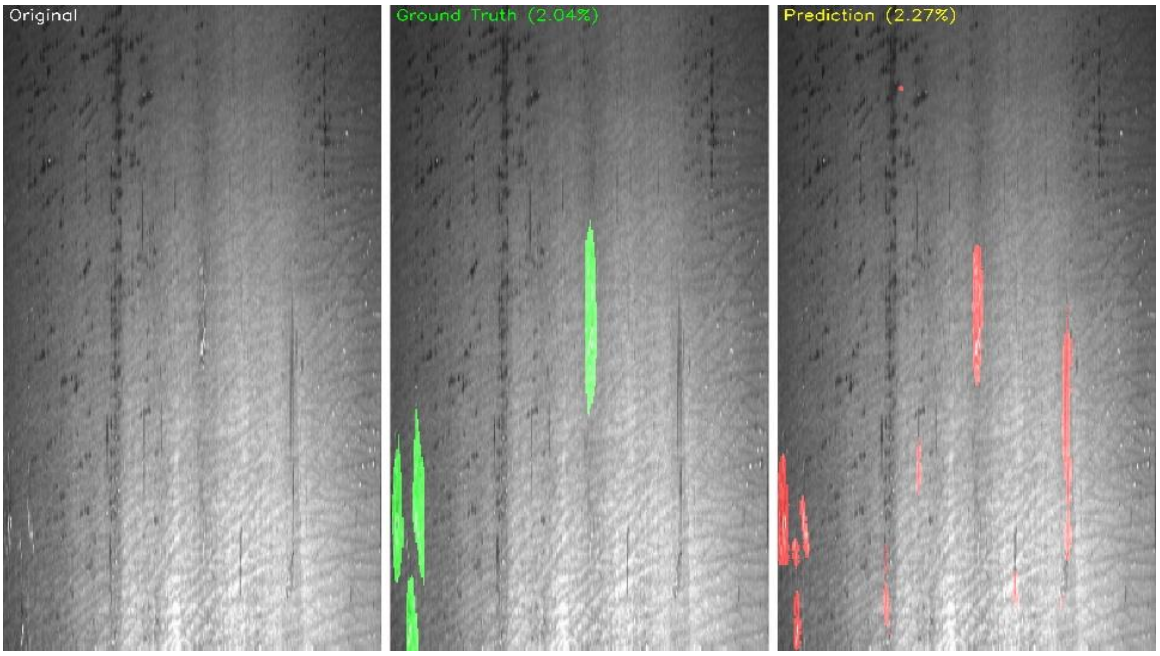

Fig. 5.6.3 Clean Surface

Case 4: Complex Textures

- Detection of major defect regions is **partially correct.**
- Noticeable **false positives on textured background**
- Background patterns are sometimes **misclassified as defects**
- Slight **over-segmentation and noise sensitivity**

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

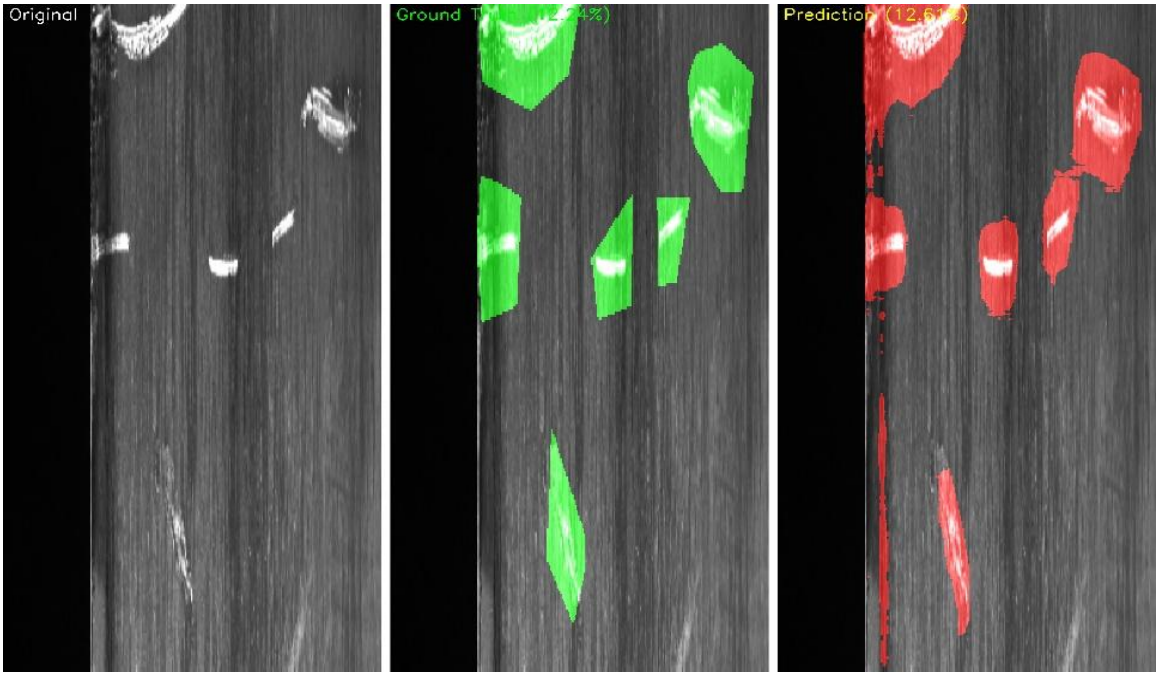

Fig. 5.6.4 Complex Texture

Overall Insight

The model performs effectively on **large and high-contrast defects**, but struggles with **fine-grained structures** due to resolution limitations and class imbalance.

Despite these limitations, the system provides reliable defect localization and enables severity estimation, making it suitable as a **baseline industrial inspection solution**.

Chapter 6. DEFECT SEVERITY ESTIMATION

This chapter presents the methodology used to quantify defect severity based on segmentation outputs and highlights its importance in industrial decision-making.

6.1. The Severity Formula

Defect severity is calculated using a pixel-ratio approach, which estimates the proportion of the steel surface affected by defects. This transforms the segmentation output into a meaningful numerical indicator.

Severity formula:

$$Severity(\%) = \frac{\text{Defect Pixels}}{\text{Total Pixels}} \times 100 \qquad (6.1)$$

Where:

- **Defect Pixels:** Total number of pixels classified as defect (value = 1) in the predicted mask
- **Total Pixels:** Total number of pixels in the input image (e.g., 128 × 256 = 32,768)

This formulation provides a simple yet effective way to quantify the extent of surface damage

Worked Numerical Example

- Image size = 128 × 256 = 32,768 pixels

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- Defect pixels = 3,276

Severity = (3276 / 32768) × 100 = **10%**

Interpretation:

- If rejection threshold = 5% → **Reject**
- If tolerance threshold = 15% → **Accept**

This demonstrates how segmentation output directly translates into actionable industrial decisions.

6.2. Industrial Applications

Quantifying defect severity enables multiple automated and data-driven processes in industrial environments:

- **Quality Grading:**
  Steel sheets can be categorized into predefined grades (e.g., Grade A: <0.5%, Grade B: 0.5–2%, Reject: >2%) based on severity thresholds.
- **Automated Rejection Systems:**
  If the defect severity exceeds a threshold, the system can trigger actuators (e.g., robotic arms or diverters) to remove defective sheets from the production line in real time.
- **Statistical Process Control (SPC):**
  Continuous monitoring of severity trends allows engineers to detect process degradation. For example, a gradual increase in defect percentage may indicate equipment wear or misalignment.
- **Batch-level Defect Analytics:**
  Aggregating severity data across multiple samples enables generation of quality reports and certification, providing objective and quantitative measures of surface integrity.
- **Process Optimization:**
  Severity feedback can be used to adjust rolling pressure, cooling rates, or surface treatment parameters, improving overall production efficiency.

6.3. Severity in Practice

The proposed system demonstrates practical applicability by predicting defect severity alongside segmentation masks.

For example:

- **Ground Truth Severity:** 11.18%
- **Predicted Severity:** 18.03%

Analysis of Severity Estimation Error

The difference between predicted severity (18.03%) and ground truth (11.18%) indicates a relative overestimation. This error appears to be **systematic rather than random**, primarily due to:

- **False Positives:** Background regions incorrectly classified as defects
- **Boundary Expansion:** Over-segmentation around defect edges
- **Resolution Artifacts:** Blurring effects increasing predicted defect area

This suggests that the model tends to **over-predict defect regions**, leading to inflated severity values.

A simple post-processing calibration (e.g., scaling factor or threshold tuning) can significantly improve reliability of severity estimation for industrial use.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

This deviation arises due to segmentation inaccuracies such as:

- Over-segmentation (false positives)
- Under-segmentation (missed defect regions)

Despite these variations, the predicted severity still provides a **consistent and objective baseline** for decision-making, which is significantly more reliable than manual inspection.

Key Insight

Even with moderate segmentation performance, severity estimation remains valuable because:

- It converts visual data into **quantitative metrics**
- It enables **automation of quality control**
- It supports **real-time industrial decision systems**

Thus, defect severity estimation bridges the gap between deep learning outputs and practical industrial applications.

Chapter 7. INDUSTRIAL IMPLICATIONS

This chapter discusses the practical significance of the proposed segmentation system in real-world industrial environments, highlighting both its advantages and limitations.

7.1. Deployment Benefits

The adoption of automated inspection systems enables consistent and repeatable evaluation of steel surfaces across production environments. Unlike manual inspection processes, which may vary due to operator-dependent factors, data-driven models provide uniform predictions based on learned representations. Furthermore, integration with production pipelines allows continuous monitoring and quantitative analysis of defect characteristics, supporting process optimization and quality assurance [6], [15].

- **Reduced Human Dependency**
  Automated inspection reduces dependency on operator-based evaluation, thereby minimizing variability introduced by human factors such as fatigue and subjective judgment.
- **Enhanced Inspection Consistency**
  Unlike human inspectors whose performance may vary over time, the model provides a consistent and objective evaluation of every steel sheet based on learned patterns.
- **Scalable Monitoring**
  A trained model can be deployed across multiple production lines simultaneously, enabling high-throughput inspection without proportional increases in manpower or cost.
- **Early Defect Detection**
  Integration of the system at early stages of production allows defects to be identified immediately, preventing further processing of defective material and reducing downstream losses.
- **Cost Reduction**
  Automation reduces labour costs and minimizes material wastage due to missed defects or over-rejection. Over time, this leads to significant operational savings and improved profitability.
- **Data-Driven Decision Making**
  The system generates quantitative outputs such as defect severity, enabling manufacturers to make objective decisions regarding grading, rejection, and process optimization.
- **Improved Traceability and Quality Assurance**
  Digital records of defect detection and severity enable better tracking of production quality, supporting audits, compliance, and customer confidence.

7.2. Limitations and Challenges

Despite its advantages, several technical and practical challenges must be addressed before full-scale industrial deployment:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- **Limited Generalization to Unseen Conditions**
  Deep learning models, including U-Net, are highly dependent on training data. Variations in alloy composition, surface finish, or manufacturing conditions not represented in the dataset may reduce model accuracy.
- **Sensitivity to Lighting Variations**
  Industrial environments often involve inconsistent lighting, shadows, reflections, and glare. Since the model relies on pixel-level features, such variations can lead to false detections or missed defects.
- **Resolution Constraints and Missed Micro-Defects**
  Due to computational limitations, input images are downscaled, which reduces the visibility of fine defects such as micro-cracks. This directly impacts recall and detection accuracy.
- **Hardware Requirements**
  Real-time high-resolution segmentation requires powerful GPUs or edge devices such as the NVIDIA Jetson. The cost of such hardware may be a barrier for small-scale industries.
- **Latency and Real-Time Constraints**
  Industrial production lines operate at high speeds (5–20 m/s). The model must process images within milliseconds to be practically deployable, requiring further optimization.
- **Class Imbalance and Detection Bias**
  Since defect pixels occupy a very small portion of the image, the model tends to bias toward background prediction, leading to lower recall.

7.3. Path Toward Industrial Deployment

To transition from a research prototype to a production-ready system, the following improvements are recommended:

- Adoption of **higher-resolution or tiled inference** for better defect visibility
- Integration of **advanced loss functions (e.g., Tversky or Boundary-aware losses)** to improve recall
- Deployment of **model optimization techniques** such as quantization and pruning
- Implementation of **robust data augmentation** to handle lighting and texture variations
- Real-time pipeline development with optimized inference latency

Overall Insight

The proposed system demonstrates strong potential for industrial adoption by combining automation, consistency, and quantitative analysis. While certain challenges remain, the framework provides a scalable foundation for next-generation intelligent quality control systems in steel manufacturing.

Chapter 8. CONCLUSION

Despite using a standard U-Net architecture, the project demonstrates strong engineering adaptation under hardware constraints, highlighting practical applicability rather than theoretical novelty.

8.1. Summary of Technical Achievements

This project successfully developed a deep learning-based framework for steel surface defect segmentation using a lightweight U-Net architecture. The model was specifically designed to operate under constrained hardware conditions, utilizing an NVIDIA RTX 2050 GPU with limited memory.

By optimizing the encoder–decoder structure, the proposed model achieved a compact size of approximately **3 million parameters**, ensuring computational efficiency while maintaining effective segmentation capability. The system successfully converts raw steel surface images into pixel-level defect maps and further translates these outputs into quantitative severity measures, bridging the gap between visual inspection and numerical analysis.

8.2. Performance vs. Industry Realities

Although the obtained Dice score (0.6302) and IoU (0.4600) are moderate from a purely mathematical perspective, they must be interpreted within the context of real-world industrial constraints.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- **Class Imbalance Handling:**
  The model achieves a high specificity of **0.9896**, demonstrating excellent reliability in identifying non-defective regions. This is critical in industrial environments to avoid unnecessary rejection of high-quality steel.

- **Localization Capability:**
  The system effectively identifies and localizes prominent defect regions such as patches, streaks, and surface irregularities, enabling actionable insights through severity estimation.

- **Performance Constraints:**
  The primary limitation in recall (0.5988) is attributed to input resolution reduction, which impacts the detection of fine-grained defects. This highlights a key trade-off between computational efficiency and segmentation accuracy.

Overall, the results demonstrate that the proposed system performs reliably under constrained conditions while providing a strong foundation for further improvements.

8.3. Operational Impact

The integration of deep learning-based segmentation into industrial inspection workflows offers significant practical benefits:

- **Consistency and Reliability:**
  The system eliminates subjectivity associated with manual inspection and ensures uniform evaluation across all samples.
- **Quantitative Decision Support:**
  Defect severity estimation transforms pixel-level predictions into measurable indicators, enabling automated grading, rejection, and process monitoring.
- **Cost and Efficiency Gains:**
  By reducing human dependency and minimizing both over-rejection and missed defects, the system contributes to improved operational efficiency and cost savings.

8.4. Final Verdict

This project establishes a **robust, scalable, and computationally efficient baseline** for automated steel surface defect detection. Despite limitations in detecting micro-defects, the framework demonstrates that deep learning-based semantic segmentation is a practical and viable solution for modern industrial quality control.

With further enhancements such as higher-resolution processing, advanced loss functions, and optimized deployment strategies, the system has strong potential to evolve into a fully autonomous, real-time inspection solution for industrial environments.

Chapter 9. FUTURE WORK

While the current project establishes a computationally efficient baseline for steel surface defect segmentation, several enhancements can significantly improve accuracy, robustness, and industrial applicability. The following directions outline potential future developments.

9.1. Technical Model Enhancements

To improve detection of fine-grained defects and overall segmentation performance, future work will focus on architectural and loss-function improvements:

- **Attention U-Net Integration**
  Incorporating attention gates [13] into skip connections will enable the model to selectively emphasize defect-relevant features while suppressing irrelevant background noise. This is expected to improve precision by reducing false positives in complex textures.
- **Advanced Loss Functions (Tversky / Boundary-Aware Variants)**
  Since the current model already employs a combined Focal + Dice loss to address class imbalance, future work can

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

explore Tversky Loss (which independently controls the penalty for false negatives via its β parameter) or boundary-aware loss functions that explicitly penalise segmentation errors near defect edges. These approaches are expected to further improve recall and reduce under-segmentation of fine-grained defect boundaries.

- **Transformer-based Architectures**
  Exploring hybrid models such as Swin-UNet [14] or Vision Transformers (ViTs) can provide better global context understanding. This is particularly useful for detecting long-range patterns such as streaks and repetitive surface anomalies.
- **Deeper and Multi-scale Feature Extraction**
  Enhancing the encoder depth or incorporating feature pyramid networks (FPN) can improve the detection of defects at multiple scales.

9.2. Data and Resolution Optimization

Improving data quality and input resolution is critical for enhancing segmentation performance:

- **Higher Resolution Training**
  Increasing input resolution from 256 × 512 toward the original 256 × 1600 will preserve fine details such as micro-cracks and thin scratches, directly improving Dice score and recall.
- **Tiled Inference Strategy**
  Instead of resizing the entire image, processing smaller patches (e.g., 256 × 512) can retain spatial detail while maintaining manageable memory usage.
- **Multi-class Segmentation**
  Extending the model from binary to multi-class segmentation will allow classification of defect types (e.g., pits, cracks, rolled-in scale), enabling more detailed industrial quality grading.
- **Data Augmentation and Robustness**
  Applying augmentation techniques such as brightness variation, noise injection, and geometric transformations can improve robustness to real-world industrial conditions.

9.3. Deployment and Industrial Integration

To transition from a research prototype to a production-ready system, several deployment-focused improvements are required:

- **Real-time Industrial Pipeline**
  Development of an end-to-end pipeline integrating high-speed industrial cameras with optimized inference models capable of processing images within milliseconds.
- **Edge-device Optimization**
  Techniques such as model quantization (FP32 → INT8), pruning, and knowledge distillation can significantly reduce model size and latency, enabling deployment on edge devices such as NVIDIA Jetson modules (embedded AI hardware).
- **Inference Time Benchmarking**
  Measuring and optimizing inference latency (target <20 ms per image) to meet industrial conveyor speed requirements.
- **Integration with Control Systems**
  Connecting the model output with programmable logic controllers (PLCs) or robotic actuators for automated rejection and sorting.

9.4. Reliability and Validation Improvements

For industrial acceptance, further validation and robustness testing are essential:

- **K-Fold Cross Validation**
  Performing multiple training-validation cycles to provide statistically reliable performance estimates.
- **Failure Case Analysis**
  Systematic analysis of misclassified images to identify model weaknesses and guide improvements.
- **Environmental Testing**
  Evaluating performance under real-world conditions such as dust, vibration, and varying lighting.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Overall Perspective

The proposed system demonstrates strong potential as a foundation for intelligent industrial inspection. With advancements in model architecture, data handling, and deployment optimization, it can evolve into a high-precision, real-time, and fully autonomous defect detection system suitable for modern manufacturing environments.

Chapter 10. REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Proc. Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015.

[4] 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, 2020.

[5] D. Tabernik, Š. Sela, J. Skvarč, and D. Skočaj, "A Segmentation-Based Deep Learning Approach for Surface Defect Detection," *Journal of Intelligent Manufacturing*, vol. 31, pp. 759–776, 2020.

[6] Q. Luo, X. Fang, L. Liu, C. Yang, and Y. Sun, "Automated Visual Defect Detection for Flat Steel Surface: A Survey," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 3, pp. 626–644, 2020.

[7] X. Zhang, Q. Yang, and Y. Wang, "Steel Surface Defect Classification Using Convolutional Neural Networks," *Materials Science and Engineering: A*, vol. 802, Art. no. 140667, 2021.

[8] Kaggle, "Severstal: Steel Defect Detection," 2019. [Online]. Available: https://www.kaggle.com/c/severstal-steel-defect-detection [Accessed: Jan. 12, 2026]

[9] K. Song and Y. Yan, "A Noise Robust Method Based on Completed Local Binary Patterns for Hot-Rolled Steel Strip Surface Defects," *Applied Surface Science*, vol. 285, pp. 858–864, 2013.

[10] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," *Proc. Int. Conf. on 3D Vision (3DV)*, pp. 565–571, 2016.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Proc. Int. Conf. on Learning Representations (ICLR)*, 2015.

[13] O. Oktay *et al.*, "Attention U-Net: Learning Where to Look for the Pancreas," arXiv:1804.03999 [cs.CV], Apr. 2018. [Online]. Available: https://arxiv.org/abs/1804.03999

[14] H. Cao et al., "Swin-UNet: UNet-Like Pure Transformer for Medical Image Segmentation," *Proc. European Conf. on Computer Vision (ECCV) Workshops*, 2022.

[15] Q. Wang, M. Wang, J. Sun, D. Chen and P. Shi, "Review of Surface-Defect Detection Methods for Industrial Products Based on Machine Vision," *IEEE Access, vol. 13, pp. 90668-90697*, 2025

[16] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

APPENDIX A. TRAINING PROTOCOL

This appendix details the technical environment and specific configurations used to conduct the experiments.

A. Hardware and Software Environment

The model was trained on a consumer-grade laptop, reflecting a focus on developing a **computationally efficient** solution capable of running on accessible hardware.

- **GPU: NVIDIA RTX 2050 (4 GB VRAM):** The Graphics Processing Unit (GPU) is the primary engine for training. The 4 GB of Video RAM (VRAM) acted as a critical constraint, necessitating the use of a lightweight U-Net architecture and a smaller batch size to prevent memory overflow.
- **CPU: Intel Core i5-12450H:** The CPU manages data loading and preprocessing tasks. This 12th-generation processor ensured that the GPU remained saturated with data, preventing bottlenecks during the training loop.
- **RAM: 16 GB:** Sufficient system memory allowed for the handling of the **Severstal Steel dataset** in-memory during preprocessing stages without causing system instability.
- **Framework: PyTorch v2.4.1 (CUDA Enabled):** PyTorch was selected for its flexibility and widespread use in the research community. CUDA (Compute Unified Device Architecture) support was enabled to offload the heavy mathematical computations (convolutions and backpropagation) to the GPU cores.
- **Python: v3.12:** The latest stable version of Python was used to leverage modern performance optimizations and library compatibility.
- **Operating System: Windows 11:** The training execution was performed in a Windows-based environment, utilizing standard drivers for NVIDIA hardware.

B. Common Training Settings

These settings define the "hyperparameters" that governed the model's learning process.

- **Batch Size: 12:** This represents the number of training samples processed before the model's internal parameters were updated. A batch size of 12 was chosen to balance the **stochastic nature of the gradient descent** with the 4 GB VRAM limit of the hardware.
- **Epochs: 40:** The model was exposed to the full training dataset 40 times. As noted in the **Learning Curve Analysis**, the model reached convergence (stabilization of loss and Dice score) around epoch 35, making 40 epochs a sufficient duration for peak performance without wasting computational time.

ACKNOWLEDGEMENT

We owe a debt of gratitude to **Dr. Roop Lal**, our project guide from the Department of Mechanical Engineering at Delhi Technological University, Delhi, for his continuous guidance, essential advice and unwavering encouragement during this project. He provided us with great support and encouragement, for which we will always be thankful. We owe a debt of gratitude to each panellist who assessed our work, offered advice and encouragement along the way, and provided us with creative ideas, information, and inspiration to continue with this project.

Paramjeet (2K22/ME/183)

Paras (2K22/ME/184)

Priyanshu (2K22/ME/204)