National College of Ireland

Machine Learning Techniques for Surface Defect and Anomaly Detection in Steel Sheets: A Hybrid Approach using Xception and Random Forest

MSc Research Project
Data Analytics

Ramit Dour
Student ID: x23102764

School of Computing
National College of Ireland

Supervisor: Dr. Anderson Simiscuka

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ramit Dour
Student ID:	x23102764
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Dr. Anderson Simiscuka
Submission Due Date:	12/08/2024
Project Title:	Machine Learning Techniques for Surface Defect and Anomaly Detection in Steel Sheets: A Hybrid Approach using Xception and Random Forest
Word Count:	5497
Page Count:	26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ramit Dour
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Machine Learning Techniques for Surface Defect and Anomaly Detection in Steel Sheets: A Hybrid Approach using Xception and Random Forest

Ramit Dour
x23102764

Abstract

This study investigates the application of multiple machine learning and image segmentation models for automated surface defect detection on steel sheets. Datasets such as the NEU Surface Defect Database, the Severstal Steel Defect Dataset, and KSDD2 are used for training. Traditional manual inspection methods for detecting surface defects like scratches, dents, and marks are laborious and can lead to errors, highlighting the need for automated solutions in industries like automotive, electrical appliances, and electronics. The research evaluates various deep learning models, including custom Convolutional Neural Networks (CNN), ResNet50, InceptionV3, EfficientNetB0, VGG19, Xception, and U-Net, to identify the most effective approach for defect detection. Among these, the hybrid model combining Xception for feature extraction and Random Forest for classification achieved the highest test accuracy of 82.22%, with a precision of 0.8967, making it the most accurate model in this study. Additionally, the Segment Anything Model (SAM) was evaluated for its segmentation capabilities, achieving a Dice coefficient of 0.72 on the validation set. These findings contribute to the development of scalable and reliable deep learning-based defect detection systems that can significantly enhance product output quality in production by reducing dependency on manual inspection.

1 Introduction

In manufacturing, the detection of surface defects is critical to ensuring product quality, particularly in industries like automotive, electronics, and metal sheet production. Defects such as scratches, dents, and marks can compromise both the functionality and appearance of the final products, leading to increased waste, customer dissatisfaction, and loss of profits. Conventionally, these defects have been identified through manual inspection, a process that is both time-consuming and prone to human error. This highlights the need for automated defect detection systems that can deliver higher accuracy and efficiency.

Recent advancements in deep learning have led to significant improvements in surface defect detection, particularly in industries requiring high precision, such as steel manufacturing. The literature demonstrates that models like EfficientNet, Cascade R-CNN, and U-Net have achieved remarkable accuracy in defect identification tasks, as shown by (Nagy and Czúni, 2022), (Akhyar et al., 2023), and (Pan et al., 2022), respectively. These

studies emphasize the need for advanced models to address the limitations of traditional inspection methods. Additionally, the successful integration of synthetic data generation for training neural networks, as highlighted by (Boikov et al., 2021), further supports the development of more robust automated systems. This research builds on these findings by exploring hybrid approaches that combine deep learning with traditional machine learning techniques to enhance the accuracy and scalability of defect detection systems, particularly for complex surfaces. This research focuses on exploring these advanced models to develop a robust system for defect detection, using datasets such as the NEU Surface Defect Database, the Severstal Steel Defect Dataset, and the KSDD2 Dataset.

1.1 Background & Motivation

Despite significant progress in the field, accurately detecting surface defects remains challenging due to the variations in defect appearance, size, environmental conditions, and surface texture types. Imbalanced datasets, where certain defect types are insufficient, lead to additional difficulties in training effective models for less frequent defects. This research aims to address these challenges by evaluating various deep learning architectures, optimizing the training process through techniques like data augmentation and hyperparameter tuning, and exploring the integration of image segmentation methods, like U-Net and the Segment Anything Model (SAM), for accurate defect spot finding.

1.1.1 Research Objectives

Research Question: How can new CNN, deep learning, and image segmentation models like Meta's SAM (Segment Anything Model) be used to automate the detection and classification of surface defects on steel sheets to improve manufacturing quality standards and reduce errors in industrial processes?

To address this research question, the research objectives include investigating different deep learning architectures, implementing these models for defect detection, and evaluating their performance using standard metrics such as accuracy, F1 score, and precision. Additionally, the research explores the application of image segmentation models to enhance the accuracy of defect localization. The findings of this research are expected to contribute to the development of more accurate and scalable automated detection systems, reducing dependency on manual inspection and enhancing product quality in manufacturing.

1.2 Report Structure

The structure of this report is as follows: the section 2 reviews the relevant literature on machine learning models for defect detection, followed by the methodology section 3 that details the experimental design and data processing techniques. The implementation section 5 describes the specific models used. The evaluation section 6 presents the results and compares the performance of different approaches. Finally, the conclusion section 7 summarizes the key findings and suggests directions for future research.

No.	Objective
1	Investigate various deep learning architectures, including CNNs and pre-trained models, to determine their effectiveness in detecting surface defects.
2	Optimize the training process by experimenting with data augmentation, class balancing, and hyperparameter tuning to improve model performance.
3	Evaluate the models using standard metrics such as accuracy, F1 score, and precision to compare their effectiveness.
4	Identify the most promising model architecture that can be used for real-world defect detection applications.
5	Explore and apply image segmentation models to isolate the defected area.
6	Evaluate the segmentation model using the Dice coefficient.

Table 1: List of objectives and tasks

2 Related Work

The recent advancements in steel surface defect detection and classification demonstrate significant progress through various deep learning techniques. (Nagy and Czúni, 2022) work utilizes EfficientNet with randomized classifiers to achieve near-perfect accuracy on benchmark datasets NEU and X-SSD, addressing challenges such as catastrophic forgetting and prolonged retraining times (Nagy and Czúni, 2022).

2.1 Machine Learning Models & CNN Convolutional Neural Network

(Akhyar et al., 2023) FDD, based on a cascade R-CNN architecture, demonstrates remarkable performance on multiple datasets, significantly outperforming YOLOv4 and YOLOv5 . (Akhyar et al., 2023) also presents an enhanced Cascade R-CNN model for steel defect detection, integrating deformable convolution and guided anchoring to improve accuracy. The model's innovative preprocessing and scaling techniques achieve a high mAP of 78.3% on the Severstal dataset, better than existing methods. However, the model's real-time applicability may be hindered by its computational demands and moderate inference speed, suggesting a need for further optimization.

Similarly, (Guan et al., 2020) employ VGG19 and DeVGG19 networks with feature visualization and quality evaluation to enhance classification accuracy and convergence speed, presenting the VSD network as a superior alternative to ResNet and VGG19 . (He et al., 2019) propose a system combining CNNs with a multilevel feature fusion network, achieving high mAP scores on the NEU-DET dataset, thereby improving detection efficiency and accuracy.

A study by (Huang et al., 2019) introduces an improved Cascade R-CNN model specifically designed for defect detection in metal cans. The study addresses the challenges posed by varying defect sizes, such as small scratches and larger printing errors, by proposing a Multi-Scale Feature Pair (MSFP) method. This method combines high-

layer features for object classification with low-layer features for bounding box regression, resulting in a significant increase in detection accuracy. The MSFP method, combined with ROI Boundary Extension, raises the model's average precision (AP@0.5) by 6.1%, achieving 39.04%—a marked improvement over the baseline Cascade R-CNN and other state-of-the-art algorithms like Faster R-CNN and SSD.

2.2 Metal Surface Defects

(Cheng, 2020) DEA_RetinaNet model incorporates difference channel attention and adaptively spatial feature fusion, significantly enhancing defect detection on the NEU-DET dataset. (Hamdi et al., 2018) explore an unsupervised algorithm for detecting defects in patterned fabrics using standard deviation filtering and K-means clustering, achieving a 95% detection success rate.

The article "Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning" by (Boikov et al., 2021), addresses the challenge of training neural networks for steel defect detection when real-world annotated data is scarce. The study proposes a method for generating synthetic datasets using Blender 3D graphics software to create photorealistic images of steel defects, which are then used to train two neural network models: Unet for segmentation and Xception for classification. The performance of these models was evaluated on real data from the Severstal: Steel Defect Detection dataset, with the Unet achieving a Dice score of 0.632 and the Xception classifier achieving a precision of 0.81 and a recall of 0.89.

2.3 Techniques used for Defect Detection

(Zhang et al., 2018) presents an image region annotation framework combining texture-enhanced JSEG segmentation and semantic correlation analysis to improve annotation accuracy. (Cao et al., 2019) propose a dual-channel CNN to enhance multilabel image labelling accuracy, particularly for low-frequency labels. (Narasimhan, 2022) study effectively demonstrates the capabilities of YOLOv7 in PCB defect detection, achieving high accuracy and efficiency. The research highlights the benefits of using advanced data augmentation and fine-tuning techniques, though further validation is needed to confirm its applicability across different datasets and industrial settings. This study underscores the potential of YOLOv7 in enhancing PCB defect detection processes, offering a promising direction for future research to optimize and extend these methodologies to broader applications.
(Wang et al., 2021) research presents a significant advancement in steel surface defect detection through the innovative IFDD (Incremental Few-Shot Defect Detection)framework. The study addresses the challenge of limited annotated data, demonstrating improvements in detection accuracy and robustness. Future research should focus on validating these findings across diverse industrial settings to ensure broader applicability. This study highlights the potential of few-shot learning in enhancing defect detection processes, offering a promising direction for future advancements in the field.

(Janeja, 2020) dual-channel CNN for fabric defect detection demonstrates significant improvements over traditional methods. (Ran et al., 2020) utilize the SSD algorithm for PCB defect detection, achieving high precision and recall rates. (Khalilian et al., 2020) use denoising convolutional autoencoders for PCB defect detection and repair, showing a 97.5% detection accuracy. Despite these advancements, the dependency on specific

datasets and the need for broader applicability in real-world scenarios remain common challenges. Future research should focus on integrating diverse datasets and exploring practical industrial applications to further generalize these methodologies.

2.4 Instance Segmentation, Semantic Segmentation and Image Segmentation for Anomaly detection in Steel sheets

(Kirillov et al., 2023) introduce the Segment Anything Model (SAM), which excels in zero-shot generalization across diverse segmentation tasks, leveraging a robust image encoder and lightweight mask decoder. (Sai and Maheswari, 2024) enhance the U-Net model with attention mechanisms for improved steel defect detection, achieving high validation accuracy and IoU scores.

(Song et al., 2024) conducted a comparative analysis against 13 state-of-the-art models using benchmarks like the SD-saliency-900, MT, and NRSD-MN datasets. The findings revealed that while SAM exhibits potential, it significantly underperforms in industrial settings, with a performance gap of up to 87% in MAE compared to leading models like CSEPNet.

In the study "Multiple Prototype Guided Enhanced Network for Few-Shot Steel Surface Defect Segmentation" by (Liang and Bai, 2024) introduces MPENet, a novel few-shot segmentation approach for steel defects. MPENet improves upon traditional methods by employing Multi-Prototype Mask Average Pooling (Multi-MAP) and a Guided Prototype Enhancement (GPE) module, which together reduce semantic bias and enhance segmentation accuracy. Tested on the FSSD-12 dataset, the model shows a 5.4% improvement in mIoU for 1-shot learning, outperforming state-of-the-art methods. While effective, the model's reliance on a specific dataset may limit its broader applicability.

A study by (Pan et al., 2022) introduces an advanced deep learning-based method for detecting defects on mobile phone screens. The study proposes EU-Net, an optimized version of the U-Net architecture, incorporating EfficientNet-B0 as the encoder and the MBConv block as the decoder. This design aims to enhance both the efficiency and accuracy of defect detection. EU-Net was evaluated on a custom dataset of mobile phone screen defects, achieving a mean Intersection over Union (mIoU) of 70.2%, outperforming models like Deeplabv3 and Attention U-Net . Despite its impressive results, the study's focus on a specific dataset with only 37 samples limits its generalizability.

2.5 Limitations and Gaps

3 Methodology

This research focuses on leveraging machine learning, specifically deep learning models such as Convolutional Neural Networks (CNNs) , Deep Neural Networks , to automate the detection of surface defects in manufacturing factories. The KDD (Knowledge Discovery in Databases) process can be effectively applied to this research project to ensure a right approach to discovering insights from data. The Figure 1 shows the required steps which need to be considered to follow the KDD approach.

Table 2: Summary of Selected Papers on Steel Surface Defect Detection

Citation	**Technique**	**Remarks**
(Nagy and Czúni, 2022)	EfficientNet with randomized classifiers	Achieves near-perfect accuracy on benchmark datasets NEU and X-SSD, addressing challenges like catastrophic forgetting and prolonged retraining times.
(Boikov et al., 2021)	Synthetic data generation using Blender, Unet, and Xception	Generates synthetic datasets for steel defect detection; Unet achieved a Dice score of 0.632, and Xception classifier achieved precision of 0.81 and recall of 0.89.
(Wang et al., 2021)	Incremental Few-Shot Defect Detection (IFDD)	Addresses limited annotated data, showing improvements in detection accuracy and robustness, with a focus on broader applicability in industrial settings.
(Cheng, 2020)	DEA_RetinaNet with difference channel attention	Enhances defect detection on the NEU-DET dataset, significantly improving detection accuracy.
(Kirillov et al., 2023)	Segment Anything Model (SAM)	Excels in zero-shot generalization across segmentation tasks, but shows a significant performance gap in industrial settings compared to leading models like CSEPNet.

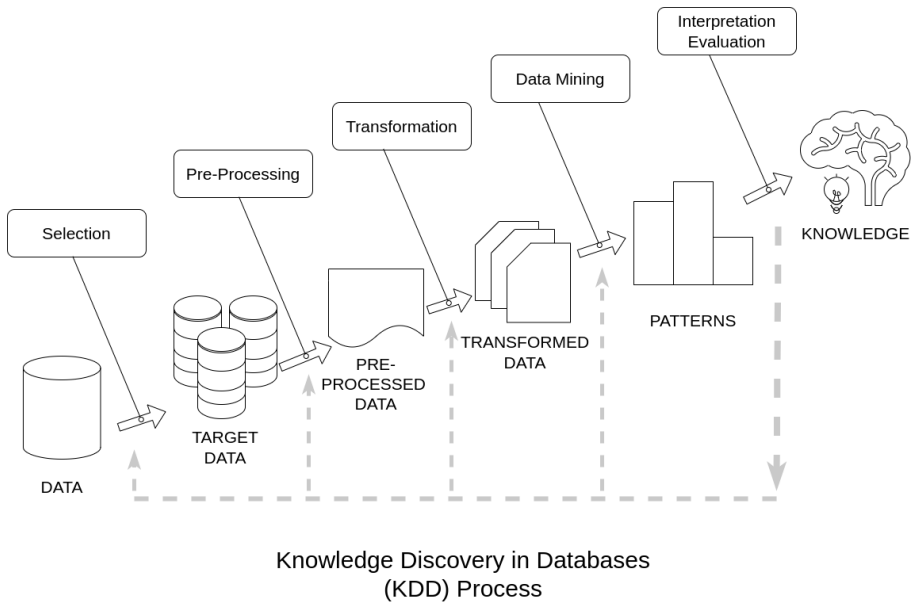


Figure 1: Knowledge Discovery in Databases (KDD) process in Steel Surface Anomaly Detection

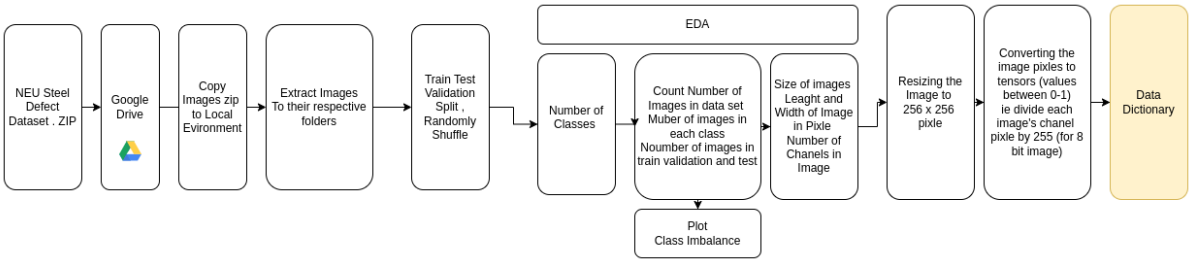


Figure 2: Data Preperation

3.1 Data Collection

In this study, the author has collected data from two primary sources: the NEU Surface Defect Database and the Severstal Steel Defect Dataset. The NEU dataset includes images of steel surfaces with six common types of defects. The Severstal dataset is of four class. With high-resolution images with various defects, sometimes defects multiple in one image. This makes it a good choice for testing. We split the data into training, validation, and test sets. This way, the models could learn from some images and then be tested on unseen images to check how well they perform. For Experiment 9 the author is using KolektorSDD2 data base , which is of binary classification with binary mask of defected segments.

3.2 Data Pre-Processing

Before using the images for model training, we had to process them for proper use in model. Author resized all images to 256x256 pixels, which made them easier for the models to handle. Next, normalize the pixel values between 0 to 1, for floating point precision. Data augmentation techniques like flipping and rotating images is also applied to remove class imbalance issues. This increased the diversity of the images, helping the models to generalize better. For images with multiple defects, we created multiple masks to label the specific areas with defects on images. This step was crucial for the models that focus on identifying and segmenting defects while fitting.

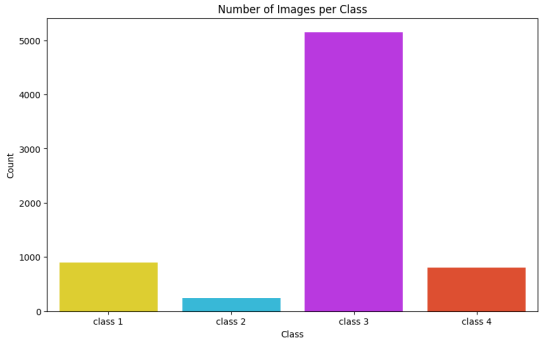
3.3 Data Transformation

Data transformation involved converting the processed images into a format that the models could use. For classification author turned the images into arrays of pixel values between 0-255, which were then organized into tensors by dividing 255. These tensors are the standard input for deep float32 tensor learning models.
Additionally, for the custom hybrid model in experiment 9, we extracted features from the images using the Xception model. These features were then flattened and used by the Random Forest classifier to make predictions about the type of defect in each image.

3.4 Data Mining

In the data mining phase, we used machine learning algorithms to find patterns in the datasets. We started with exploratory data analysis to understand how defects were distributed. As seen in Fig 3 Each model gets weights and trains them to find features in image that helps in finding and classifying different types of defects.

(a) Class Distribution for Six types of defects in NEU dataset. *(Balanced Distribution)*



(b) Class Distribution for 4 types of defects in Severstal dataset. *(Unbalanced Distribution)*

Figure 3: Data Set Class Distribution

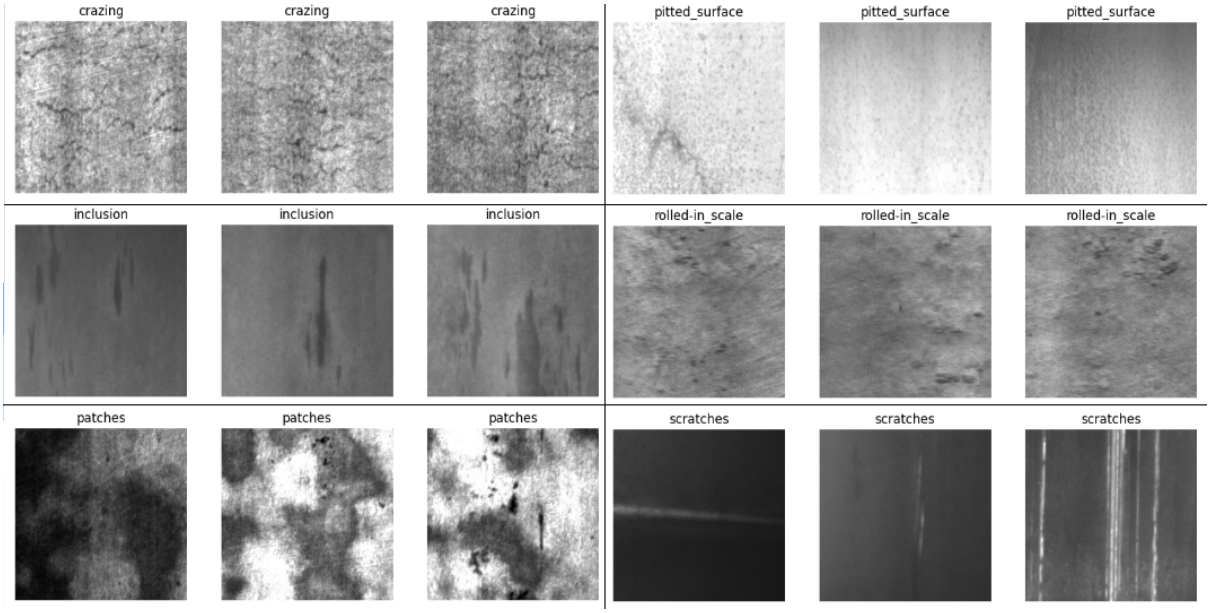
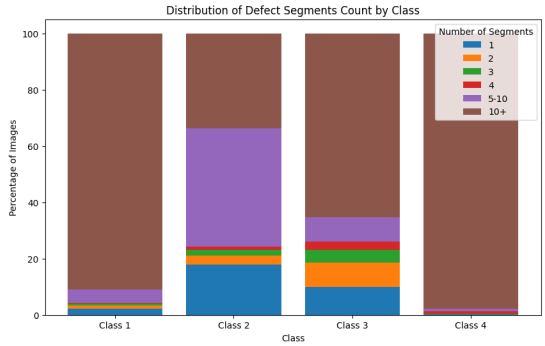
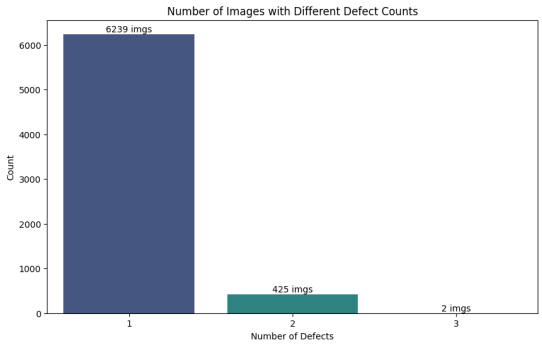


Figure 4: Six types of defects in NEU dataset



(a) Distribution of number of defects per image for each class in Severstal dataset.



(b) Single and Multiple class defect counts per image

Figure 5: Severstal Data Set Class Distribution

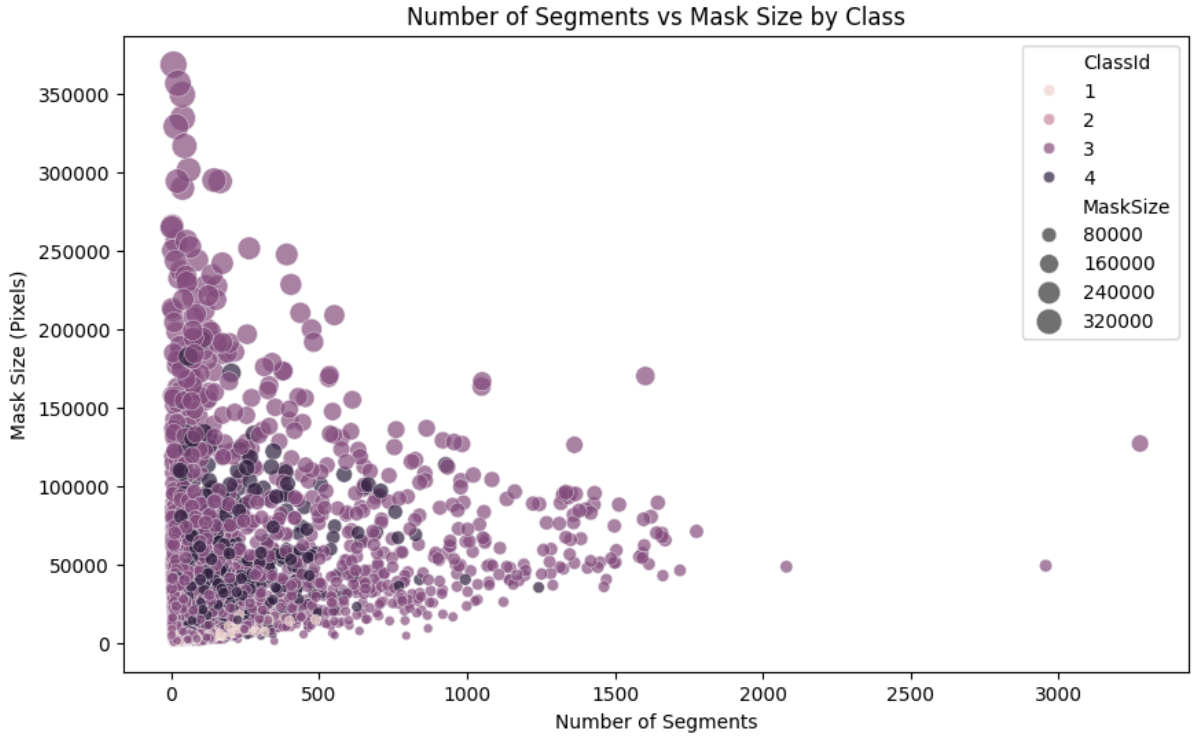


Figure 6: Number of Segments of defect with its mask size(number of pixels) for all four classes for each image in Severstal dataset

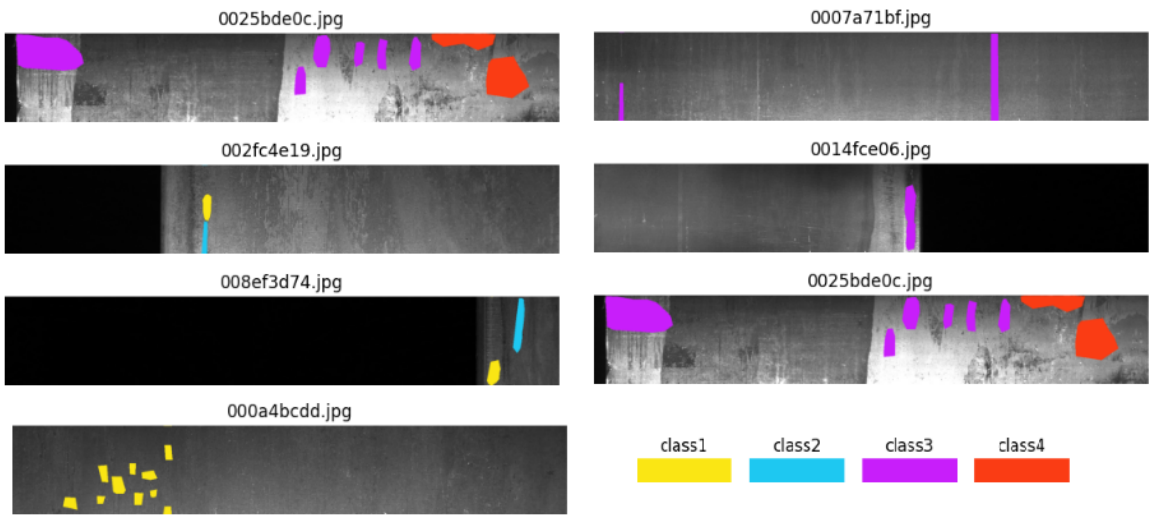


Figure 7: Four types of defects in Severstal Dataset with respective Masks

3.5 Data Pattern Interpretation and Evaluation

Using evaluation metrics like accuracy, F1 score, and precision to understand the strengths and weaknesses of models. For classification, the confusion matrices tells where the models made mistakes while testing. For segmentation, the Dice coefficient to measure how well the models identified defect areas as compared to ground truth.

4 Design Specification

The design and implementation of the Ml models and segmentation models used in this study. The primary focus was on leveraging deep learning models such as Convolutional Neural Networks (CNNs), ResNet50, InceptionV3, EfficientNetB0, VGG19, and Xception for defect detection and classification. Additionally, U-Net and the Segment Anything Model (SAM) by META were employed for defect segmentation tasks. The custom hybrid approach of combining Xception with a Random Forest classifier was a novel aspect of this research. This custom model was designed to enhance classification accuracy by using deep learning for feature extraction and traditional machine learning for final classification. The overall framework involved several stages: data pre-processing, model training, feature extraction, and evaluation as shown in Fig 8. Each model was fine-tuned using hyperparameter optimization techniques to achieve the best performance.

The models were implemented using Python, TensorFlow, and Keras, and trained on Google Colab with dedicated GPU support to high computation the process. The implementation of segmentation models like U-Net and SAM aimed to provide precise localization of defects, offering an additional solution for automated surface defect detection in steel sheets using semantic segmentation.

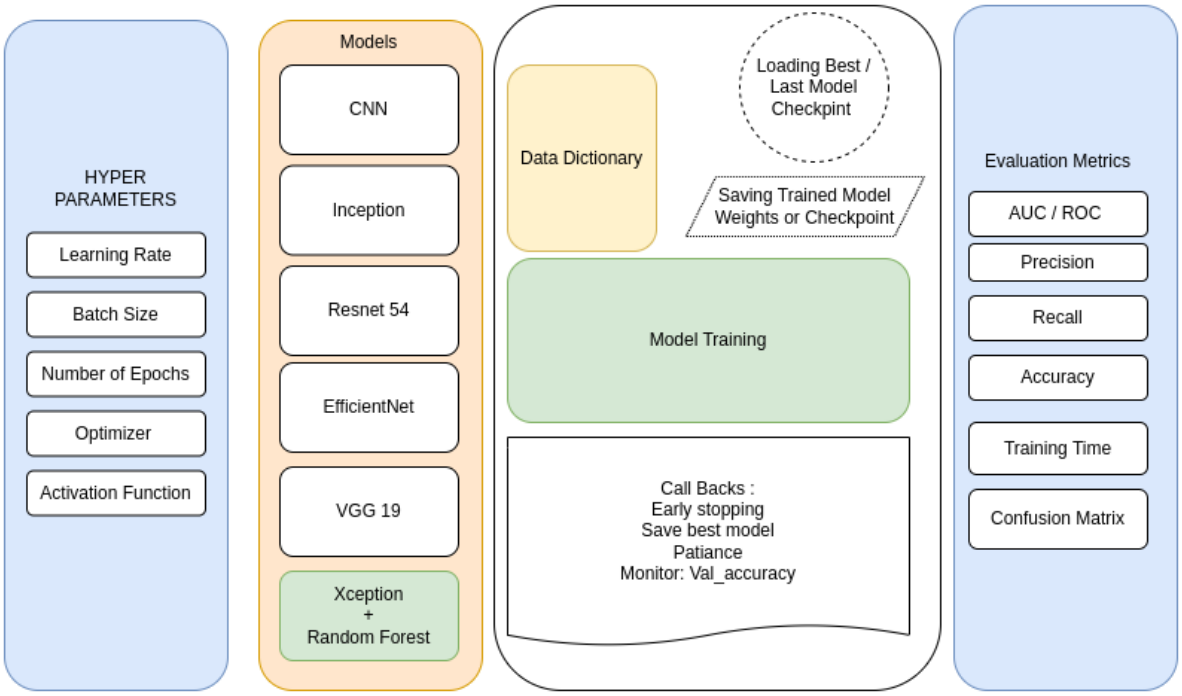


Figure 8: Project Design Flow

5 Implementation

In implementation of this study, various outputs were produced, including transformed datasets, custom code, and multiple deep learning models for surface defect detection. The data was preprocessed using Python libraries such as TensorFlow, Keras, and transformed into suitable formats for model training. Models like CNN, ResNet50, InceptionV3, and Xception were developed and fine-tuned using transfer learning. Additionally, a custom hybrid model combining Xception with a Random Forest classifier was implemented for experiment 9. The entire process, from data augmentation to model evaluation, was executed using Python in a Jupyter Notebook environment on Google Colab, leveraging both CPU and NVIDIA's A100 GPU resources for training ad testing.

5.1 Experiment 1-6: Combined Implementation

1. Dataset and Preprocessing

- **Dataset:** For Experiments 1 to 6, the NEU Surface Defect Database was utilized. This dataset consists of images of steel surfaces categorized into six defect types. The images were standardized to 256x256 pixels for uniformity across all experiments.
- **Data Augmentation:** To mitigate overfitting and improve the generalization of models, data augmentation techniques were applied during training. These included random rotations, horizontal and vertical flips, and zoom transformations. The augmented images helped the models to better learn the variations in defect patterns.
- **Common Preprocessing Steps:**

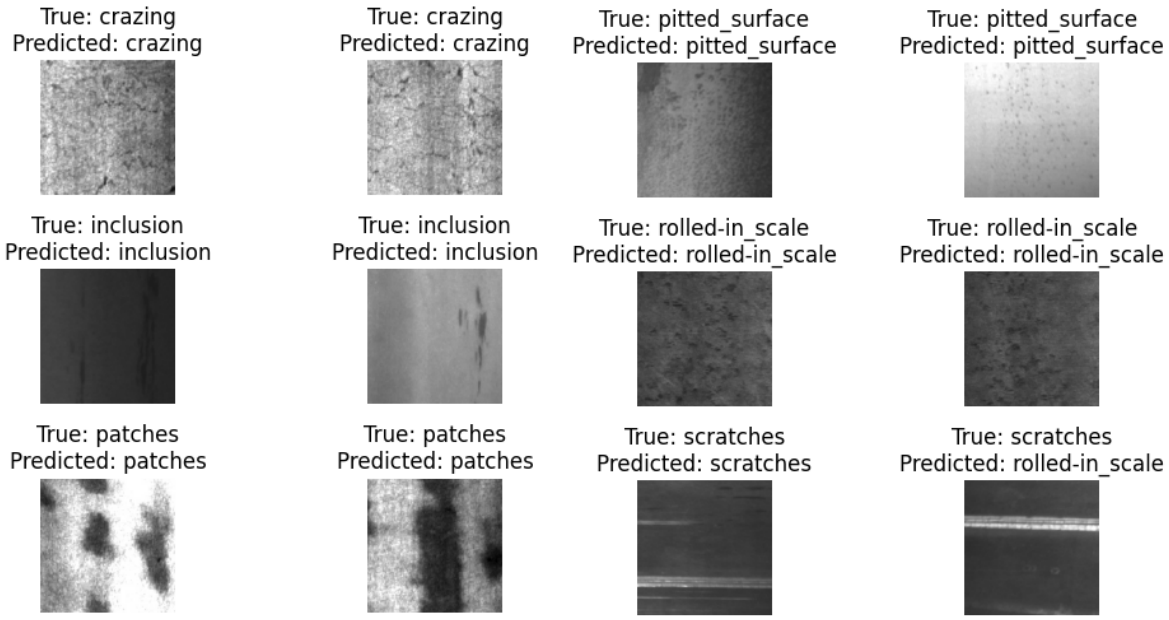


Figure 9: Actual and Predicted Labels of Images

- **Normalization:** All pixel values were normalized to the range [0, 1] to enable faster convergence during training.
- **Dataset Split:** The dataset was split into training and validation sets with an 80:20 ratio, ensuring that the models were evaluated on unseen data as well.

2. Model Implementations

- **2.1 Custom Convolutional Neural Network (CNN):** A simple CNN was constructed with three convolutional layers followed by max-pooling layers and dense layers. The model was designed to capture the spatial features of the surface defects through multiple layers of convolutions.
- **2.2 ResNet50 (Transfer Learning):** The ResNet50 model was pre-trained on the ImageNet dataset and fine-tuned on the NEU Surface Defect Database. Only the final few layers were trained, while the rest of the network weights were kept frozen.
- **2.3 InceptionV3 (Transfer Learning):** Similar to ResNet50, the InceptionV3 model was employed with a pre-trained architecture. Fine-tuning was performed on the final layers to adapt the model to the specific task of surface defect detection.
- **2.4 EfficientNetB0 (Transfer Learning):** EfficientNetB0, a more recent and efficient architecture, was used with transfer learning. The model's compound scaling approach allowed it to perform well even with fewer parameters.
- **2.5 VGG19 (Transfer Learning with Data Augmentation):** VGG19 was fine-tuned with additional data augmentation. The model's deep architecture provided detailed feature extraction, which was further enhanced by the augmented data.

- **2.6 Xception (Transfer Learning with Data Augmentation):** Xception, known for its depthwise separable convolutions, was also fine-tuned with data augmentation. This model was designed to capture complex patterns in the defect images with minimal computational cost.

3. Loss Function and Optimizer

- **Loss Function:** Cross-entropy loss was used for all models, as the task was a multi-class classification problem.
- **Optimizer:** The Adam optimizer was selected for all experiments due to its adaptive learning rate, which allows for efficient convergence. The learning rate was initially set to 0.0001 for fine-tuning.

4. Training Procedure

- **Training Duration:** All models were trained for 50 epochs with a batch size of 32.
- **Early Stopping:** Early stopping was implemented to halt training if the validation accuracy did not improve for 10 consecutive epochs.
- **Model Checkpoints:** Model checkpoints were saved whenever an improvement in validation performance was observed, ensuring that the best model weights were retained.

5.2 Experiment 7: Xception + Random Forest

Dataset and Preprocessing

The same NEU Surface Defect Database and preprocessing steps from Experiments 1 to 6 were used. However, Experiment 7 involved a two-step process that integrated deep learning with traditional machine learning methods.

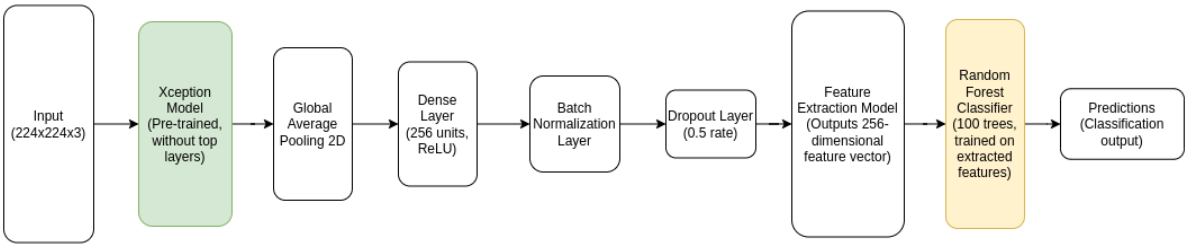


Figure 10: Xception + Random Forest for Classification (Experiment 7)

Feature Extraction with Xception

1. **Xception Model**: The Xception model was pre-trained on ImageNet and fine-tuned on the defect dataset to serve as a feature extractor. After training, the model was used to generate feature vectors for each image in the dataset by extracting the output from one of the final layers before the classification head.

2. **Feature Vector Creation**: These feature vectors captured high-level representations of the images, including complex patterns and textures indicative of surface defects. The extracted features were then flattened and stored as input for the next step.

Classification with Random Forest

1. **Random Forest Model**: A Random Forest classifier was employed using the feature vectors extracted by the Xception model. Random Forest is an ensemble learning method that builds multiple decision trees during training and merges them to produce more accurate and stable predictions.

2. **Working Mechanism**: The Random Forest model creates a large number of decision trees, each trained on a random subset of the data and features. The final classification is determined by aggregating the predictions from all individual trees, typically by majority voting. This method is particularly robust to overfitting, especially in cases with high-dimensional data like the feature vectors generated from Xception.

3. **Hyperparameters**: The number of trees in the forest was set to 100, with the maximum depth of each tree adjusted based on cross-validation results. Other hyperparameters, such as the minimum samples split and leaf, were fine-tuned to optimize performance.

Training Procedure

1. **Feature Extraction**: The feature vectors were first generated for the entire dataset using the fine-tuned Xception model.

2. **Random Forest Training**: The Random Forest classifier was trained on the extracted feature vectors using the training set. Validation was performed using a separate validation set to tune hyperparameters and avoid overfitting.

3. **Evaluation**: The model was evaluated on the test set, where it demonstrated the best performance among all experiments, with high accuracy and precision.

5.3 Experiment 8: U-Net Image Segmentation

In Experiment 8, the U-Net architecture was implemented to perform segmentation of surface defects in steel images. U-Net is widely used for image segmentation tasks due to its ability to precisely localize objects and is particularly effective in biomedical and industrial applications.

Dataset: The Severstal Steel Defect Dataset was employed for this experiment. This dataset comprises high-resolution images of steel surfaces with labeled defects. The images were resized to 256x1600 pixels to standardize the input for the U-Net model. The dataset was split into training and validation sets, with an 80:20 split.

Data Preprocessing:

- **Normalization:** The pixel values of the images were normalized to a range between 0 and 1.

- **Data Augmentation**: To enhance the model's ability to generalize, data augmentation techniques were applied to the training images. These included random rotations, shifts, and flips, which helped in simulating various defect orientations and positions on the steel surface.

Model Architecture: U-Net: The U-Net model was constructed with an encoder-decoder structure. The encoder path comprised several convolutional layers with ReLU activation functions and max-pooling layers to capture the context and reduce spatial dimensions. The decoder path used transposed convolutional layers for up-sampling, along with skip connections that allowed the model to retain fine-grained spatial information from the encoder's corresponding layers. The final layer of the U-Net model employed a sigmoid activation function to produce a binary mask, indicating the presence or absence of defects in each pixel.

Loss Function and Optimizer:

- **Loss Function:** The Dice coefficient was used as the loss function, chosen for its effectiveness in handling class imbalance, which is a common issue in segmentation tasks. The Dice loss measures the overlap between the predicted segmentation and the ground truth, focusing on the precision and recall of the segmented regions.
- **Optimizer:** The Adam optimizer was utilized for training, with a learning rate of 0.0001. Adam is well-suited for this type of task due to its adaptive learning rate capabilities, which help to converge faster while avoiding local minima.

Training Procedure:

- The model was trained for 50 epochs with a batch size of 32.
- **Early Stopping:** Early stopping was implemented to prevent overfitting, based on the validation Dice coefficient. If the validation performance did not improve for 10 consecutive epochs, training was halted, and the best model weights were restored.
- **Model Checkpointing:** Model checkpoints were saved at each epoch where an improvement in validation performance was observed, ensuring that the best model was retained.

5.4 Experiment 9 : Meta SAM (Segment Anything Model) Image Segmentation

In Experiment 9, the KSDD2 dataset, also known as the Kolektor Surface Defect Dataset 2, was indeed used for evaluating the Segment Anything Model (SAM). This dataset is specifically designed for the segmentation of surface defects in industrial settings, providing a more challenging and specialized test for SAM compared to general datasets.

The experiment was designed to leverage SAM's capabilities in segmenting various types of surface defects found in the KSDD2 dataset. This dataset contains high-resolution images of surfaces with different types of defects, including scratches, dents, and other anomalies. The images were processed to match SAM's input requirements, specifically resized to 256x256 pixels.

Model Architecture: SAM, which utilizes a vision transformer (ViT) backbone along with a segmentation head, was employed. This model is known for its ability to generate high-quality segmentation masks across a wide range of tasks, making it ideal for this application.

Training Procedure: Due to the complexity of the KSDD2 dataset, SAM was fine-tuned for several epochs using an Adam optimizer with a learning rate of 1e-5. The dataset was divided into training and validation sets, with a typical 80/20 split. During training, various prompts were used, including bounding boxes and points, to guide the model in generating accurate segmentation masks.

6 Evaluation

This section provides an in-depth evaluation of the performance of various image classification models used in this study. Each model's performance was assessed based on its validation accuracy, test accuracy, F1-score, precision, and the time taken for training. The evaluation was conducted using the NEU Surface Defect Database, which consists of images categorized into six distinct classes. The following models were evaluated:

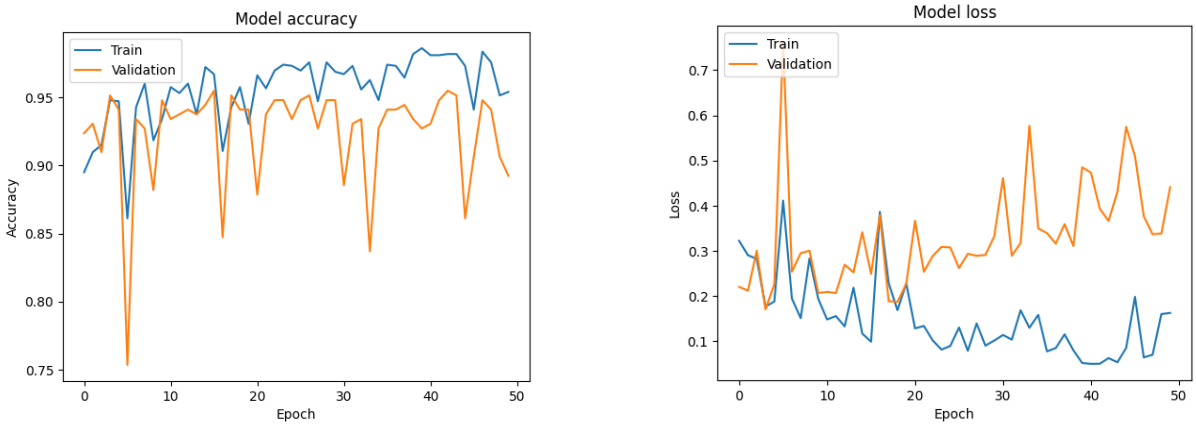
Table 3: Model Hyperparameters for Surface Defect Segmentation and Classification

Expt No.	**Model**	**Learning Rate**	**Batch Size**	**Optimizer**	**Other Hyperparameters**
1	CNN (Custom)	0.001	32	Adam	50 Epochs, Dropout: 0.5, Activation: ReLU
2	ResNet50	0.0001	32	Adam	50 Epochs, Fine-tuning layers: Last 10 layers
3	InceptionV3	0.0001	32	Adam	50 Epochs, Fine-tuning layers: Last 10 layers
4	EfficientNetB0	0.0001	32	Adam	50 Epochs, Fine-tuning layers: Last 5 layers
5	VGG19	0.0001	32	Adam	50 Epochs, Data Augmentation: Yes, Dropout: 0.5
6	Xception	0.0001	32	Adam	50 Epochs, Data Augmentation: Yes, Fine-tuning layers: Last 15 layers
7	Xception + Random Forest	0.0001	32	Adam	50 Epochs , Random Forest: 100 Trees, Max Depth: 10
8	U-Net	0.0001	32	Adam	50 Epochs, Loss: Dice Loss, Dropout: 0.5
9	SAM	0.00001	4	Adam	10 Epochs, Loss: DiceCELoss, Max Epochs: 10

6.1 Experiment 1: Custom Convolutional Neural Network (CNN)

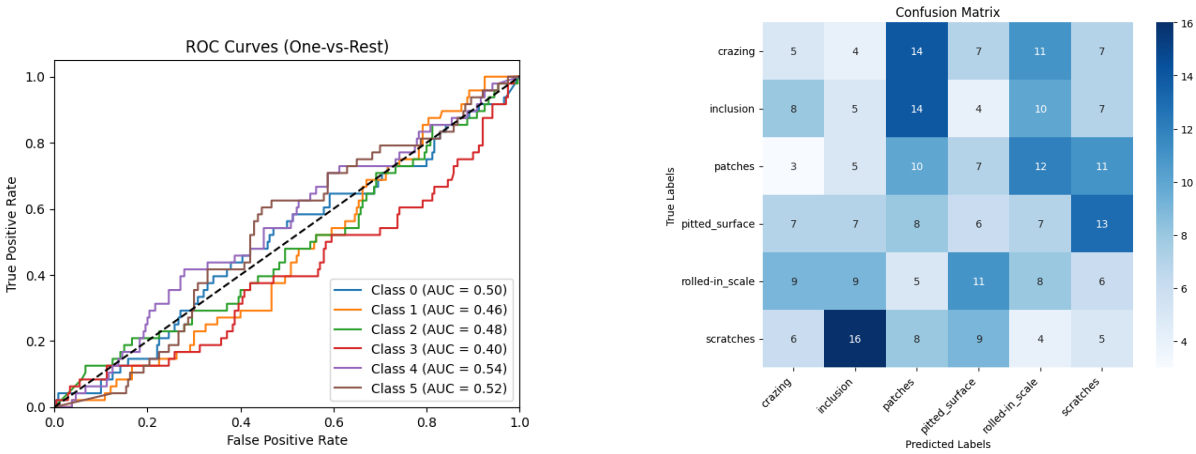
The first experiment involved a custom-built CNN model, which was specifically designed for the task of surface defect detection. The model achieved a validation accuracy of 89.24%. However, the test accuracy dropped significantly to 13.54%, indicating a severe overfitting issue. The F1-score and precision were both recorded at 0.13, reflecting poor performance on unseen data. The training process took approximately 30 minutes to complete.

Key Insights: Although the CNN model showed promising validation accuracy, its poor performance on the test set suggests that the model was unable to generalize well beyond the training data. This highlights the necessity of employing more robust models or regularization techniques to improve generalization.



(a) Accuracy Curve (b) Loss Curve

Figure 11: Accuracy and Loss curves for the CNN model.



(a) ROC Curve (b) Confusion Matrix

Figure 12: Accuracy and Loss curves for the CNN model.

6.2 Experiment 2: ResNet50 (Transfer Learning)

The second experiment utilized the ResNet50 architecture, a well-known model for image classification, through transfer learning. The model achieved a validation accuracy of

90.28%, yet the test accuracy was significantly low at 20%. The F1-score and precision were recorded at 0.20 and 0.18, respectively. The training process was relatively efficient, taking 25 minutes.

Key Insights: Despite leveraging a powerful pre-trained model, ResNet50 did not perform well on the test set, indicating that fine-tuning alone was insufficient to address the complexity of the surface defect dataset.

6.3 Experiment 3: InceptionV3 (Transfer Learning)

In the third experiment, the InceptionV3 model was used for transfer learning. This model achieved a validation accuracy of 87%, with a test accuracy of 19.44%. The F1-score and precision were around 0.1944 and 0.1979, respectively. The model training took 35 minutes.

Key Insights: InceptionV3 showed comparable validation and test performance, though both were relatively low. This indicates that while the model was somewhat balanced in its performance, it may not be the best choice for this particular classification task.

6.4 Experiment 4: EfficientNetB0 (Transfer Learning)

The fourth experiment involved using EfficientNetB0 for transfer learning. This model achieved the lowest validation accuracy of 16.67%, with a test accuracy of 17%. The F1-score was recorded at 0.17, with a precision of 0.1733. The model took the longest time to train, approximately 45 minutes.

Key Insights: EfficientNetB0 underperformed compared to other models, both in terms of validation and test accuracy. The extended training time also did not translate into better performance, suggesting that this model may not be well-suited for the dataset.

6.5 Experiment 5: VGG19 (Transfer Learning with Data Augmentation)

In the fifth experiment, the VGG19 model, enhanced with additional data augmentation techniques, was evaluated. This model achieved the highest validation accuracy of 96.67%. However, the test accuracy was 18.05%, indicating a significant drop in performance. The F1-score and precision were recorded at 0.1813 and 0.21469, respectively, with a training time of 40 minutes.

Key Insights: VGG19 demonstrated excellent validation accuracy but suffered from overfitting, as indicated by the lower test accuracy. The data augmentation techniques did not sufficiently address this issue.

6.6 Experiment 6: Xception (Transfer Learning with Data Augmentation)

The sixth experiment used the Xception model, also with data augmentation. The model achieved the second-best validation accuracy of 99.17% and a test accuracy of 19.166%. The F1-score and precision were both around 0.19166 and 0.1834, respectively. Notably, this model had the shortest training time, taking only 15 minutes.

Key Insights: Xception showed remarkable validation accuracy and quick training time, making it a highly efficient model. However, similar to other models, it struggled with test performance, which suggests overfitting.

6.7 Experiment 7: Xception + Random Forest (Feature Extraction and Classification)

The final experiment combined feature extraction using Xception with classification via a Random Forest model. This approach achieved a validation accuracy of 84% and a test accuracy of 82.22%, the highest test accuracy among all models. The F1-score was recorded at 0.82, and precision was the highest at 0.8967. The total training time was 10 minutes.

Key Insights: The combination of Xception for feature extraction and Random Forest for classification yielded the best overall performance on the test set. This model not only addressed the overfitting issue seen in other models but also did so with a significantly lower training time.

The evaluation of various models for image classification of NEU surface defects reveals that while traditional models like CNN and transfer learning architectures such as ResNet50, InceptionV3, EfficientNetB0, VGG19, and Xception show varying degrees of overfitting, the combination of Xception and Random Forest stands out as the most effective approach. This hybrid model provided a balanced performance with the highest test accuracy, demonstrating its superior generalization capabilities. Future work could explore further fine-tuning of this approach or investigate other hybrid techniques to continue improving classification accuracy on unseen data.

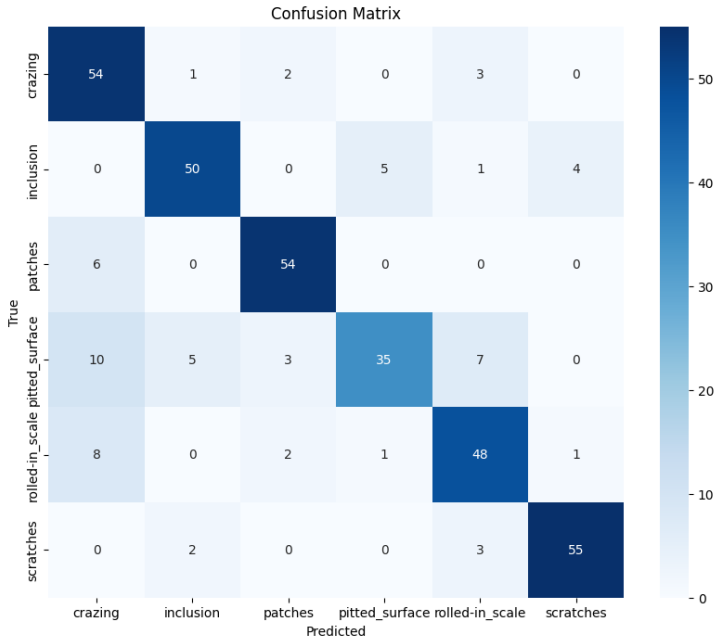


Figure 13: ConfusionMatrix for Experiment 7 : Xception + Random Forest for Classification

6.8 Experiment 8: U-Net Image Segmentation

The U-Net model's performance was evaluated based on several metrics, including Dice coefficient, validation loss, and visual inspection of the segmented masks.

Training and Validation Performance: The model achieved a final Dice coefficient of approximately 0.72 on the training set, indicating good segmentation performance. However, the validation Dice coefficient plateaued at 0.648, suggesting that the model may have started to overfit the training data. Despite this, the model demonstrated a consistent ability to identify defects across multiple classes during validation.

Qualitative Results: Visual inspection of the predicted masks showed that the U-Net model was able to accurately segment the defect areas in most cases. However, the model struggled with smaller and less distinct defects, leading to occasional false positives and missed detections. These challenges were likely due to the complex and variable nature of the defect patterns in the dataset.

Quantitative Results: The U-Net model's performance was quantified using the following metrics:

- **Dice Coefficient:** The model achieved a Dice coefficient of 0.648 on the validation set, reflecting its ability to overlap the predicted and actual defect regions accurately.
- **Validation Loss:** The final validation loss was recorded at 0.019, indicating a reasonable level of segmentation accuracy.
- **Segmentation Accuracy:** The model was able to segment defects with an accuracy rate of 71.66% during training, but this decreased slightly to 64.77% during validation, highlighting the challenge of generalizing to unseen data.

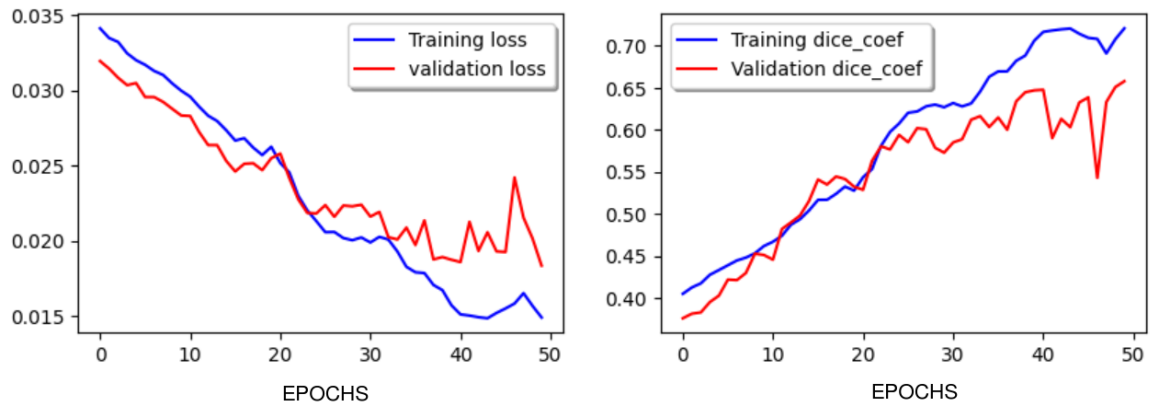


Figure 14: Unet model for 50 Epoch vs Loss and Dice Coeff. For training and validation

6.9 Experiment 9 : Meta SAM (Segment Anything Model) Image Segmentation

The performance of SAM on the KSDD2 dataset was evaluated both quantitatively and qualitatively.

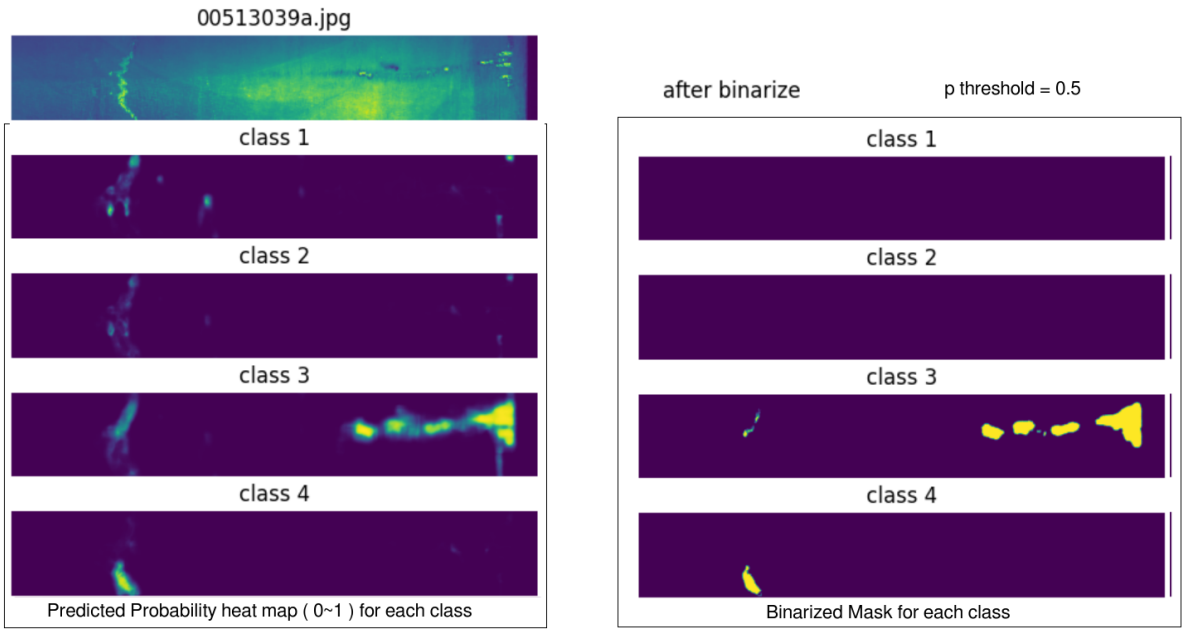
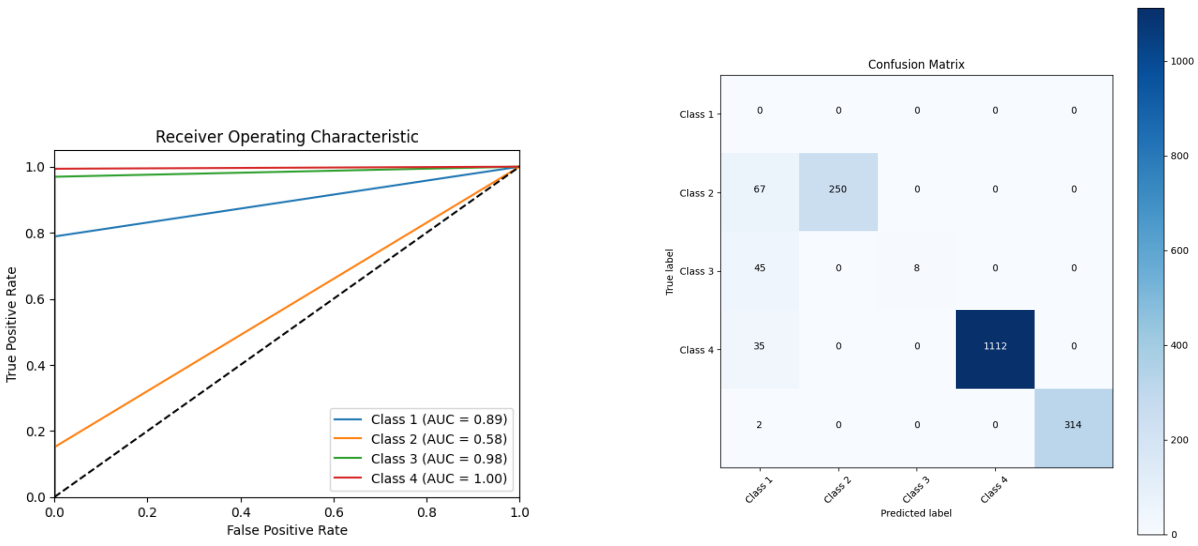


Figure 15: Prediction probability Masks (on left) for each class 1-4 and its binary mask (on right)



(a) ROC Curve for Unet Segmentation

(b) Unet model Confusion Matrix

Figure 16: Unet model Metrics

Quantitative Metrics:

- **Dice Coefficient:** SAM achieved a Dice coefficient of 0.72 on the validation set, indicating a reasonable overlap between predicted masks and the ground truth.
- **Validation Loss:** The validation loss stabilized at 0.021, showing that the model had converged during training.

Qualitative Analysis: Visual inspection of the segmentation outputs confirmed SAM's ability to accurately detect and segment defects in most cases. However, as with other datasets, SAM occasionally struggled with very fine or ambiguous defects, resulting in partial segmentations or misses.

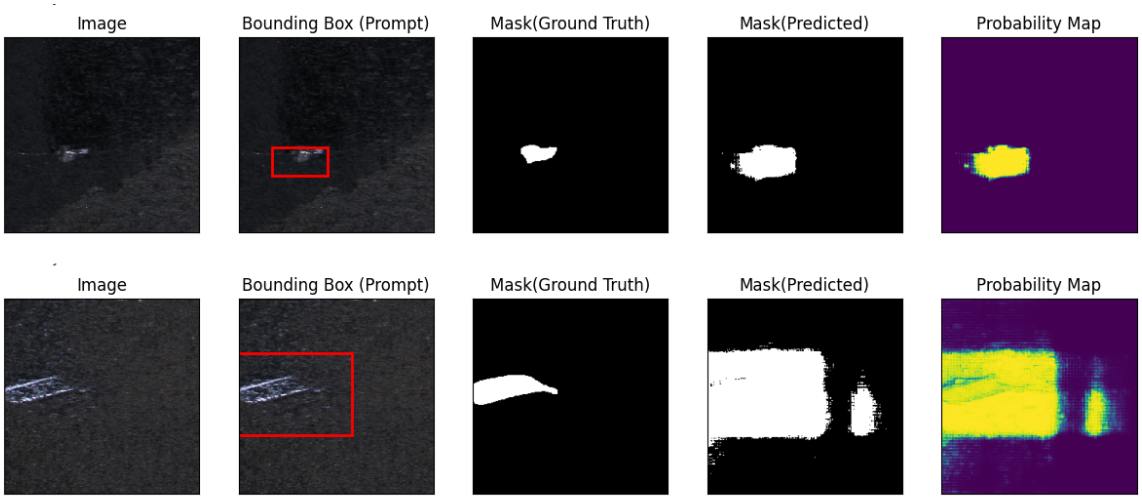


Figure 17: Prediction of defects using Meta SAM segmentation model

6.10 Model Comparison

In Experiment 7 the combination of deep feature extraction with Xception and classification with Random Forest proved to be highly effective. Xception captured the intricate patterns within the defect images, while Random Forest provided a robust and interpretable classification method. This hybrid approach leveraged the strengths of both deep learning and ensemble methods, resulting in superior performance compared to traditional models trained directly on the image data.

The U-Net model showed strong performance in segmenting surface defects, particularly for larger and more distinct defects. Its ability to combine low-level spatial information with high-level contextual information through skip connections proved effective in this task. The model's performance decreased slightly on the validation set, suggesting potential overfitting. Additionally, the model had difficulty accurately segmenting smaller or less distinct defects, which could be addressed by further tuning the model architecture or augmenting the dataset.

6.11 Discussion

The author critically evaluated the findings from the nine experiments conducted to classify and segment surface defects using various machine learning models. The author examined the strengths and weaknesses of each approach, comparing the performance metrics like Dice coefficient, segmentation accuracy, and training times. The discussion addresses the effectiveness of different architectures, such as U-Net and SAM, and the integration of Random Forest with Xception for improved classification. Additionally, we critique the design choices, suggest possible improvements, and contextualize the results within existing literature, identifying areas where future research could enhance model performance.

Table 4: Metrics for Steel Defect Segmentation Models

Metric	**U-Net: Training Set**	**U-Net: Validation Set**	**SAM: Training Set**	**SAM: Validation Set**
Dice Coefficient	0.72	0.648	0.78	0.72
Segmentation Accuracy	71.66%	64.77%	74.50%	70.56%
Validation Loss	-	0.019	-	0.021
Training Time	120 minutes		40 minutes	

Table 5: Comparison of Different Models for Image Classification

No.	**Model Used**	**Type of Classification Method**	**Val Accuracy**	**Test Accuracy**	**F1-Score**	**Precision**	**Time Taken (minutes)**	**Notes**
1	CNN (Custom)	Custom Convolutional Neural Network (CNN)	89.24	13.54	0.13	0.13	30	High validation accuracy but poor test performance.
2	ResNet50	Transfer Learning using ResNet50	0.9028	0.2	0.2	0.18	25	Decent validation accuracy, but low test performance.
3	InceptionV3	Transfer Learning using InceptionV3	0.87	0.1944	0.1944	0.1979	35	Comparable validation and test performance, but both are low.
4	EfficientNetB0	Transfer Learning using EfficientNetB0	0.1667	0.17	0.17	0.1733	45	Low validation and test performance.
5	VGG19	Transfer Learning using VGG19 with additional data augmentation	**0.9667**	0.1805	0.1813	0.21469	40	Highest validation accuracy, but test accuracy is low.
6	Xception	Transfer Learning using Xception with additional data augmentation	<u>0.9917</u>	0.19166	0.19166	0.1834	**15**	Second-best validation accuracy and quickest training time, but test performance is low.
7	**Xception + Random Forest**	Feature extraction using Xception, followed by Random Forest for classification	0.84	**0.8222**	**0.82**	**0.8967**	<u>10</u>	Best overall test performance and high precision, though validation accuracy is lower than others.

7 Conclusion and Future Work

In conclusion this study demonstrates that combining deep learning models with traditional machine learning techniques can improve the accuracy and reliability of surface defect detection in steel sheets. The custom hybrid model, using Xception for feature extraction and Random Forest for classification, is the most effective approach, offering the best balance between validation and test performance. Additionally, the U-Net model and the Segment Anything Model (SAM) showed promising results for defect segmentation,for multiple defects

However, challenges remain, particularly in the segmentation of smaller or less distinct defects, and in further reducing the model's accuracy. Future work should focus on refining these models through advanced data augmentation techniques, the development of ensemble models, and the application of post-processing steps to enhance segmentation accuracy. The recent release of SAM 2 by Meta presents another opportunity for future exploration, which could potentially offer improvements in segmentation performance. Testing these models in real-world manufacturing environments would also be essential to validate their effectiveness and scalability.

References

Akhyar, F., Liu, Y., Hsu, C.-Y., Shih, T. K. and Lin, C.-Y. (2023). Fdd: A deep learning based steel defect detectors, *The International Journal of Advanced Manufacturing Technology* **126**(3): 1093–1107.

Boikov, A., Payor, V., Savelev, R. and Kolesnikov, A. (2021). Synthetic data generation for steel defect detection and classification using deep learning, *Symmetry* **13**(7): 1176.

Cao, J., Wu, C., Chen, L., Cui, H. and Feng, G. (2019). An improved convolutional neural network algorithm and its application in multilabel image labeling, *Computational Intelligence and Neuroscience* **2019**: 2060796.

Cheng, Xun Yu, J. (2020). Retina net with difference channel attention and adaptively spatial feature fusion for steel surface defect detection, *IEEE Transactions on Instrumentation and Measurement* **70**: 1–11.

Guan, S., Lei, M. and Lu, H. (2020). A steel surface defect recognition algorithm based on improved deep learning network model using feature visualization and quality evaluation, *IEEE Access* **8**: 49885–49895.

Hamdi, A. A., Sayed, M. S., Fouad, M. M. and Hadhoud, M. M. (2018). Unsupervised patterned fabric defect detection using texture filtering and k-means clustering, *2018 International Conference on Innovative Trends in Computer Engineering (ITCE)*, IEEE, pp. 113–118.

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

Huang, Z., Xiao, H., Zhang, R., Wang, H., Zhang, C. and Shi, X. (2019). Multi-scale feature pair based r-cnn method for defect detection, *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, pp. 46–51.

Janeja, L. (2020). *Identification of Defects in the Fabric using Deep Convolutional Neural Networks*, PhD thesis, Dublin, National College of Ireland.

Khalilian, S., Hallaj, Y., Balouchestani, A., Karshenas, H. and Mohammadi, A. (2020). Pcb defect detection using denoising convolutional autoencoders, *2020 International Conference on Machine Vision and Image Processing (MVIP)*, IEEE, pp. 1–5.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al. (2023). Segment anything, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026.

Liang, C. and Bai, S. (2024). Multiple prototype guided enhanced network for few-shot steel surface defect segmentation, *2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, IEEE, pp. 1216–1219.

Nagy, A. M. and Czúni, L. (2022). Classification and fast few-shot learning of steel surface defects with randomized network, *Applied Sciences* **12**(8): 3967.

Narasimhan, M. (2022). *Printed Circuit Board Defect Detection using YOLOv7*, PhD thesis, Dublin, National College of Ireland.

Pan, J., Zeng, D., Tan, Q., Wu, Z. and Ren, Z. (2022). Eu-net: A novel semantic segmentation architecture for surface defect detection of mobile phone screens, *IET Image Processing* **16**(10): 2568–2576.

Ran, G., Lei, X., Li, D. and Guo, Z. (2020). Research on pcb defect detection using deep convolutional neural network, *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, IEEE, pp. 1310–1314.

Sai, D. G. S. M. and Maheswari, U. (2024). Enhanced steel defect detection using u-net model with attention mechanism, *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, Vol. 5, IEEE, pp. 1700–1705.

Song, K., Cui, W., Yu, H., Li, X. and Yan, Y. (2024). Samera: Can it segment any industrial surface defects?, *Computers, Materials & Continua* **78**(3).

Wang, H., Li, Z. and Wang, H. (2021). Few-shot steel surface defect detection, *IEEE Transactions on Instrumentation and Measurement* **71**: 1–12.

Zhang, J., Mu, Y., Feng, S., Li, K., Yuan, Y. and Lee, C.-H. (2018). Image region annotation based on segmentation and semantic correlation analysis, *IET Image Processing* **12**(8): 1331–1337.