

Defect Detection in Manufacturing: An Integrated Deep Learning Approach

Tewogbade Shakir Adeyemi

CAPE Economics Research and Consulting, Emeryville, USA

Email: pingcommercial@gmail.com

How to cite this paper: Adeyemi, T.S. (2024) Defect Detection in Manufacturing: An Integrated Deep Learning Approach. *Journal of Computer and Communications*, 12, 153-176.
<https://doi.org/10.4236/jcc.2024.1210011>

Received: January 4, 2024

Accepted: October 22, 2024

Published: October 25, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This research investigates deep learning-based approach for defect detection in the steel production using Severstal steel dataset. The developed system integrates DenseNet121 for classification and DeepLabV3 for segmentation. DenseNet121 achieved high accuracy with defect classification as it achieved 92.34% accuracy during testing. This model significantly outperformed benchmark models like VGG16 and ResNet50, which achieved 72.59% and 92.01% accuracy, respectively. Similarly, for segmentation, DeepLabV3 showed high performance in localizing and categorizing defects, achieving a Dice coefficient of 84.21% during training and 69.77% during validation. The dataset includes steels which have four different types of defects and the DeepLab model was particularly effective with detection of Defect 4, with a Dice coefficient of 87.69% in testing. The model performs suboptimally in segmentation of Defect 1, achieving an accuracy of 64.81%. The overall system's integration of classification and segmentation, alongside thresholding techniques, resulted in improved precision (92.31%) and reduced false positives. Overall, the proposed deep learning system achieved superior defect detection accuracy and reliability compared to existing models in the literature.

Keywords

Defect, Detection, Classification, Segmentation, Deep Learning

1. Introduction

A key parameter for superior consideration in manufacturing industry is operational efficiency. Different errors are certain to happen during production cycle. Engineering functions are highly repetitive in nature and thus higher chances of errors, mistake and defects [1]. Defects detection is a quality control procedure in production settings which allow reduction of waste and enhancement of quality

of manufactured products. Various traditional means to detect defects in pre-industrial 4.0 era revolve around manual operations which have elongated cycle time and are labour intensive (**Table 1**).

Table 1. Various traditional approaches in defect detection.

Traditional Methods	Description	Weakness
Manual Measurement	<ul style="list-style-type: none"> - Making use of measuring tools to investigate defects due to shape or size - Tools like gauges, callipers and micrometres 	<ul style="list-style-type: none"> - Subjective - High human error - Limited precision and accuracy [2] [3].
Physical Inspection	<ul style="list-style-type: none"> - Trained inspection visually checked defects 	<ul style="list-style-type: none"> - Subjective - High human error - Limited precision and accuracy [4] [5].
Testing by Ultrasonic	<ul style="list-style-type: none"> - High frequency sound waves to detect defects 	<ul style="list-style-type: none"> - Directional sensitivity - Inaccurate interpretation by operation [6] [7].

As noted by [8], the traditional methods of detecting defects are subjective in nature due to over reliance on human judgement. These contribute to limitation in desired outcomes and the need for automation. While [9], asserted the shift in paradigm from manual to automation in industrial 4.0 age, [10] supported this view with highlight of various automated sensors that improve quality level in production cycle. With advanced and sophisticated technologies involving parallel computing, big data, machine learning, AI and others, most production processes are handled smartly to detect defects [11]. Machine learning is an approach where patterns are identified automatically from data which are used to make predictions [12]. It is the act of creating computer systems that learn from specific dataset and automatically produce models which deliver models. Generally, machine learning performs classification, clustering, selection of features, regression, ranking and prediction. Feature selection is the aspect of machine learning that deals with selecting highly relevant (non-redundant) variables and removing irrelevant (less weighted) ones [13]. With feature selection, model prediction accuracy is greatly improved while time for processing and cost of computation is reduced. Many machine learning algorithms have been used in detection of defect in production in the era of industry 4.0. Defect is an important phenomenon considering quality control in production [14]. Defect detection studies using machine learning often require labelled datasets to train and evaluate models. They are essential for training, evaluating, and fine-tuning machine learning models. The quality, diversity, realism, and size of the dataset directly impact the model's performance, generalization, and real-world applicability [15]-[17]. For defect

detection, various classes of dataset have been used like image dataset, video dataset, sensor datasets, point cloud dataset, text or document datasets and composite datasets. Image datasets are commonly used and can be highly effective for defect detection studies using machine learning. There are several reasons why image datasets are often preferred: rich visual information, availability and ease of collection. The use of image dataset also provides improved synergy with deep learning and wider opportunities for transfer learning. The collaboration between deep learning and image datasets has revolutionized defect detection by enabling accurate and automated analysis of visual data. The ability of deep learning models to learn hierarchical features, train end-to-end, leverage transfer learning, and benefit from data augmentation has significantly advanced the field of defect detection in a wide range of industries and applications [18]-[20]. The aim of this research is to develop a very robust deep learning system that can detect defects with highest possible accuracy through integration of classification and segmentation techniques. The motivation of choosing deep learning for defect detection aligns well with the core principles of Industry 4.0. Deep learning techniques possess high computational efficiencies, and they have potentials to work in automated mode. While it is evident that traditional defect detection methods (**Table 1**) like manual inspection and ultrasonic testing are limited by human error, subjective interpretation, and inaccurate precision [21]. Manual methods also involve significant labour costs and extended turnaround times. This greatly reduces overall efficiency in modern industrial setup [8]. In contrast, deep learning models automatically learn and extract relevant features from large volumes of data while delivering high accuracy and reliability with defect detections [22]. These models can process images or sensor data faster and more consistently compare to traditional techniques while improving production quality [23]. Importantly, the ability of deep learning models to scale with increasing data availability is critical in Industry 4.0, where large-scale IoT sensor networks continuously generate data [24].

2. Literature Review

Noted by [25], in any standardized production setting, quality control and management must be optimized. Traditional means of handling quality of the processes and end-products are manual-based and highly subjective. Sophistication in entire engineering set-up demand for modern solutions to monitor various aspects of production and the products. With emergence of industry 4.0, manufacturing processes have advanced with strong support from computing technologies which lead to improved all-round efficiency. In this context, machine learning has been the new magic wand to detect anomalies and optimized processes [26]. Machine learning approaches possess stronger objectivity with lesser human interaction. The machine learning approach is modern and rely majorly on existence of suitable data which will be trained sufficiently to identify defect when apply to new data. Several studies have been conducted in this domain. The summary of the works is presented in **Table 2**.

Table 2. Summary of existing works.

Author	Summary of the Work	Results
Wu and Zhou (2021) [27]	Classified and detected defective components from industrial images using CNN and compared results with SVM, KNN, BPN, and MLP.	CNN: Accuracy 91.4%, Recall 84.9%, F1 Score 88.0%. MLP: Accuracy 85.5%. BPN: Accuracy 84.7%. KNN: Accuracy 79.6%. SVM: Accuracy 76.3%.
Westphal and Seitz (2021) [25]	Used VGG16 and Xception CNN to detect defects in Selective Laser Sintering (SLS) product powder bed.	VGG16: Accuracy 95.8%, Precision 93.9%, Recall 98.0%, F1 Score 95.9%, ROC-AUC 0.982. Xception: Accuracy 89.4%, F1 Score 89.7%, ROC-AUC 0.982.
Yang <i>et al.</i> (2020) [28]	Conducted a survey on defect detection and compared machine learning techniques across various products.	Liu <i>et al.</i> (2017) CNN: Accuracy 94.68%. Kumar <i>et al.</i> (2018) CNN: Accuracy 86.2%. He <i>et al.</i> (2019) [23] Fully CNN: Accuracy 99.14%. Lv and Song (2019): 97.25%.
Ramesh Rao and Bhelkar (2022) [29]	Studied defects in manufacturing using CNN, RCNN, Fast RCNN, and Faster RCNN models.	Faster RCNN: Accuracy 99.90%, Fast RCNN: Accuracy 98.70%, RCNN: Accuracy 98.70%, CNN: Accuracy 98.50%. Prediction Time: Faster RCNN: 0.2 s, RCNN: 40 s.
Khalfaoui <i>et al.</i> (2022) [10]	Detected defects using sensor data in automotive production using ML models like LR, GNB, DT, LDA, RF, and DNN.	DNN: Accuracy 74%, RF: 64%, LDA: 57%, DT: 56%, GNB: 63%, LR: 63%.
Wang, Wu and Wu (2020) [30]	Detected defects in vehicle parts using VGG16 compared to HOG + SVM.	VGG16: Accuracy 95.29%, HOG+SVM: Accuracy 93.88%.

The reviews show that deep learning possess enhanced capabilities with defect detection compared to traditional machine learning techniques. Conversely, most research focused solely on classification methods in addressing defect detections. This research will extend the knowledge around defect detection by combining classification and segmentation approach to build a robust system. Also, aside the general related works on defect detection, literatures with Severstal dataset was reviewed. These research in Section (2.1) are used as benchmarked for the current research.

Related Works: Benchmarked

Abu *et al.* 2020 [25] used SEVERSTAL dataset to predict surface defects in steel production using four deep learning techniques VGG16, MobileNet, Densenet121 and Resnet101. The dataset was pre-processed by using OpenCV to rescale images to 256×480 pixels while 50 epochs and 32 batch sizes were used for their models. The accuracy of their models is shown **Table 3**.

Table 3. Deep learning results of Abu *et al.* 2020 investigation.

Model	Accuracy
VGG	50.00%
MobileNet	79.91%
Densenet121	70.34%
Resnet101	70.50%

The results showed that MobileNet yielded highest accuracy compared to others.

Akhyar *et al.* 2023 [22] investigated surface defects in steel manufacturing using SEVERSTAL steel dataset, NEU dataset and DACM dataset. They proposed a deep learning approach termed forceful steel defect detector FDD which is rooted in R-CNN with deformable convolution and deformable ROI pooling which integrate with the geometric shape of defects. Their models accuracy were measured in terms of average recall (AR) and mean average precision (mAP) and comparison were made (Table 4).

Table 4. Deep learning results of Akhyar *et al.* 2023 [22] investigation.

Model	Author/Year	Model Backbone	AR	mAP
YOLO v4	Bochkovski, Wang and Liao 2020 [31]	CSPDarknet	0.904	0.608
YOLOv5	GitHub 2021 [32]	CSPDarknet	0.891	0.601
YOLOX	Ge <i>et al.</i> 2021 [33]	CSPDarknet	0.863	0.652
Cascade R-CNN	Qiao, Chen and Yuille 2020 [34]	Resnet 50	0.855	0.675
FDD (proposed)		Resnet 50	0.969	0.783

3. Methodology

The implementation of predictive system that detect defect is broken down into six segments as represented in the workflow in Figure 1. The six segments include data preparation, multi-label classification, segmentation and detection, thresholding and post-processing, output processing and evaluation of test set.

3.1. Data Preparation

In ML programming domain, dataset is necessary to build models that make predictions. The process involves inputting relevant dataset into algorithms that learn from the peculiar patterns embedded in the data to make informed predictions [24]. Clearly, the development of effective models starts with data preparation [35] [36]. When relevant dataset for targeted investigation is acquired, the data is prepared through cleaning, transformation and making it ready for computational analysis. The manufacturing dataset acquired in this research was published by [37]. It

was sourced from Severstal, a leading steel industry company located in Russia. Severstal created very large industrial data lake from production of flat sheet steel in order to monitor defects (**Figure 2**).

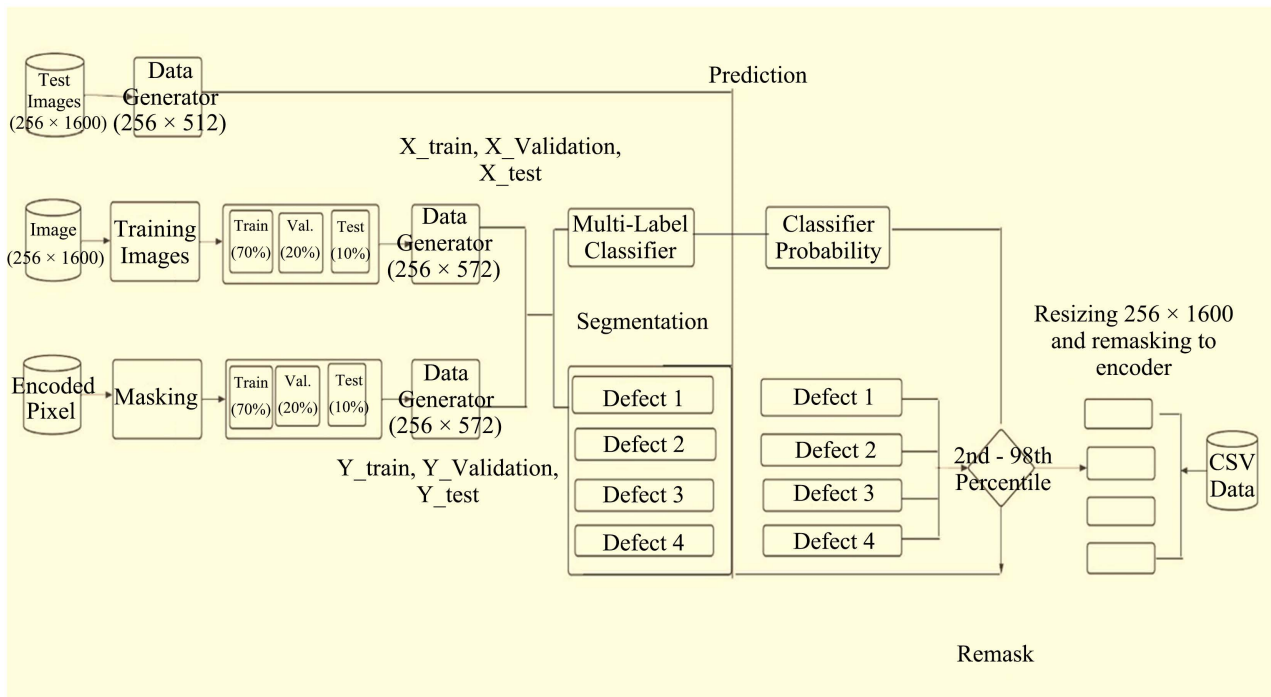


Figure 1. System workflow.

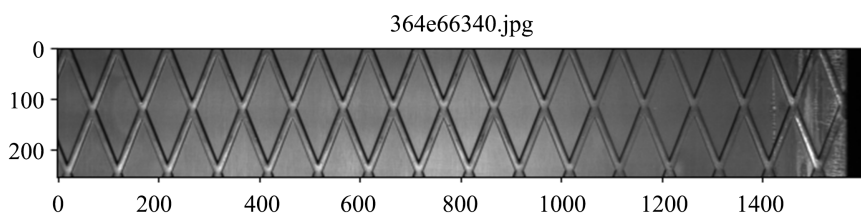


Figure 2. Sampe flat sheet steel from severstal dataset.

The dataset is in three segments: train images (count = 12,568), test images (count = 5506) and train CSV (7095,3). The 12,568 images exist in four classes with various kinds of defects. Images with common defect are present in class 3 (**Figure 3**).

The data preparation step involves resizing the images to match the fixed dimension of the input shape, creating arrays to hold the images and their corresponding class identifiers, and transforming the categorical labels into integers using LabelEncoder (**Figure 4**). The images' pixel intensities were also normalized to be within the range of 0 - 1. To handle overfitting and enable the model to learn from a more generalized set of features, data augmentation techniques were applied, including shifting, rotation, zooming, and flipping. Transforming the provided run-length encoded masks into binary masks for the segmentation task.

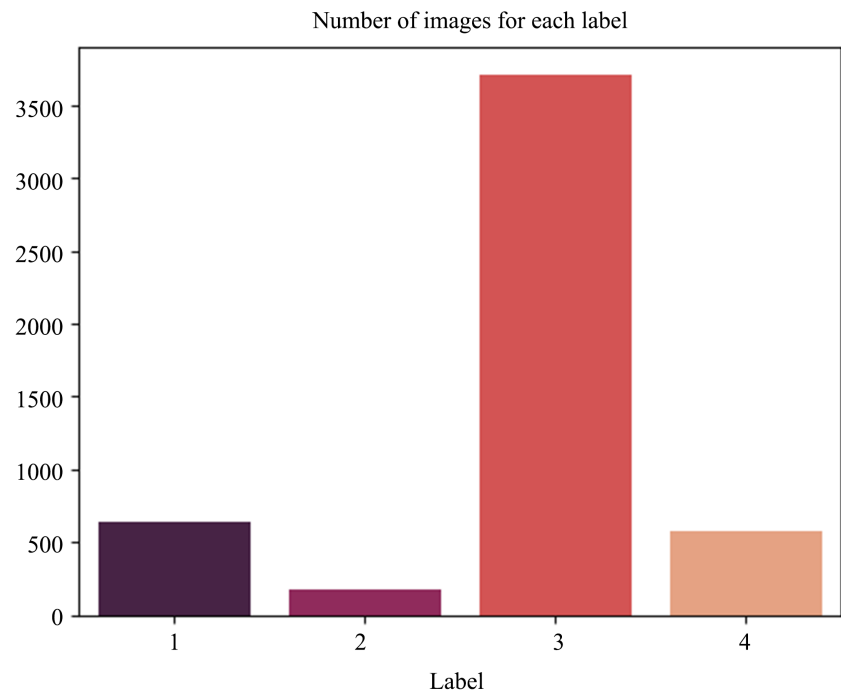


Figure 3. Defect type count plot.

```

Conversion of Encoded Pixel to Mask

1 # function to convert EncodedPixels into mask
2 def rle2mask(mask_rle, shape=(1600,256)):
3
4     s = mask_rle.split()
5     starts, lengths = [np.asarray(x, dtype=int) for x in (s[0:][::2], s[1:][::2])]
6     starts -= 1
7     ends = starts + lengths
8     img = np.zeros(shape[0]*shape[1], dtype=np.uint8)
9     for lo, hi in zip(starts, ends):
10        img[lo:hi] = 1
11    return img.reshape(shape).T
12
13 def mask2rle(img):
14
15    pixels= img.T.flatten()
16    pixels = np.concatenate([[0], pixels, [0]])
17    runs = np.where(pixels[1:] != pixels[:-1])[0] + 1
18    runs[1::2] -= runs[:-2]
19    return ' '.join(str(x) for x in runs)ode here

```

Figure 4. Snippet of Python code for pre-processing.

3.2. Multi-Label Classification

Machine learning systems are versatile tools capable of handling a variety of tasks which includes classification, regression, and segmentation [38] [39]. Classification tasks are common, and they involve categorization of data into predefined classes based on input features [40] [41]. When two label exist in the dataset, the classification is binary. When label is more than two, classification becomes multi-

label. The Severstal dataset contains a target variable with more than two labels (defects). The dataset has four types of defects: defect 1, defect 2, defect 3 and defect 4. A multi-label classifier is employed in the ML system set-up of the current research. This enables prediction of different defect class in the dataset. The classifier is trained on the training set and evaluated using the validation set. The classifier's input includes the generated training images (256×512) and corresponding labelled masks. As represented in **Figure 1**, for each prediction, a classifier probability score is calculated. This reveals the model's confidence in the presence of each defect that is detected.

3.3. Segmentation and Detection

Segmentation in machine learning is another important task and it is common with image processing, object identification and biomedical applications [42]-[44]. The main target of segmentation in all these applications is streamline the outlook of image being investigated into item that is more coherent for visual sighting [44] [45]. In this research, segmentation has been integrated with classification to further streamline the image of defect that is detected by classification. The multi-label classifier in Section (3.2) identifies types of defects that exist in a particular sheet material while segmentation is implemented to understand the part of the sheet where defect is detected. The multi-label classifier identifies different types of defects (Defect 1, Defect 2, Defect 3, Defect 4) that exist in an image. For each defect, predictions are produced that help in understanding which part of the image contains each defect.

3.4. Thresholding and Post-Processing

Thresholding is important in situation where classification is integrated with segmentation. This enables proper refinement of predictions made by the classification algorithm. Basically, the purpose of thresholding is to establish a cutoff point for classifier outputs to decide which predictions should be considered valid detections of defects. With adoption of thresholds between the 2nd and 98th percentiles, the thresholding method ensures that predictions with low confidence (potential false positives or false negatives) are discarded. As affirms by [46] and [47], thresholding increases the reliability of model's performance. The remasking process that follows thresholding is specified to further improve the accuracy of the segmentation. Remasking involves reintroducing the predicted defects and comparing them against the original pixel mask (ground truth). This process facilitates refinement of the segmented regions through elimination of possible noise while validating the areas where defects are predicted [48].

3.5. Output Processing

The re-masked images are resized back to the original resolution (256×1600) before final output. The final outputs consist of a CSV file containing: **ImageId** (the unique identifier for each image) **ClassId** (the defect class identified Defect

1, Defect 2, Defect 3 and Defect 4) and the **Encoded Pixels** (the encoded pixel values representing the locations of the defects)

3.6. Evaluation on Test Set

Evaluation of machine learning models directly involves assessment of model performance [49]. The evaluation reflects the applicability and efficiency of the models across different domains. Evaluation is important to determine the suitability of models implemented for specific investigation while ensuring their effectiveness in real-world applications [50]. There are many metrics used in machine learning domain to measure model performances [51]. Based on the integration of classification and segmentation, the applicable metrics includes accuracy, precision, F1-score, recall, dice coefficient and dice loss. Accuracy measures the proportion of correct predictions made by a model out of all predictions [52]. Precision focuses on the ability of the model to correctly identify only the relevant positive instances [53]. It is computed as the ratio of true positives to the sum of true positives and false positives. Recall, also known as sensitivity, evaluates the model's ability to detect all possible positive cases by dividing true positives by the sum of true positives and false negatives. The F1-score is the harmonic mean of precision and recall. It yields a balanced metric when trade-off exist between precision and recall. The Dice coefficient is one of the metrics for similarity [54] [55]. It measures the overlap between predicted and actual segmentations, with values ranging from 0 (no overlap) to 1 (perfect overlap). Conversely, dice loss is a loss function derived from the Dice coefficient. It is used to train models by penalizing the model for poor overlap between predicted and actual regions.

4. Model Selection and Implementation

4.1. Initial Experimentation: Classification

Prior testing before final model selection was adopted to yield improved model performance in the overall system. Initial experimentation was carried out using four models for classification: EfficientNetB1, ResNet50, DenseNet121 and VGG16. EfficientNetB1 is part of a family of models that uses compound scaling method, balancing network depth, width, and resolution to achieve high accuracy with fewer parameters [56]. The EfficientNet architecture is designed to optimize both accuracy and efficiency. This makes it suitable for large-scale image classification tasks [57]. Just like EfficientNetB1, ResNet50 is a member of family of ResNet. According to its name, ResNet50 maintain depth with 50 layers. ResNet50 is deep CNN architecture that function with consideration of residual connections [58]. The residual connections resolve the issue of vanishing gradients to boost training of deeper networks. ResNet50 has strength to capture detailed information in images with improved adaptability for item recognition across the images [59]. DenseNet121 is a CNN architecture that work with the reuse of dense connections and features [60]. Fundamentally, in DenseNet, every layer is linked to every other layer in feed-forward settings. While in traditional deep learning setup, output

layer usually acts as the only input to subsequent layers in DenseNet configuration, dense connections are introduced every layer to get feature, and they allow maps of layers before it [61] [62]. This special arrangement control growth of network and computational complexity of the algorithms which support improved performance [63]. VGG is also a CNN model developed by Visual Geometry Group (VGG) at the University of Oxford. The VGG16 architecture is the one with 16 layers with learnable weights [64] [65]. This architecture reduces the spatial dimensions of the input image while increasing the depth of the feature maps [66]. VGG16 produces better classification due to its ability to focus on targeted image area [67].

As justified by many evidence of their strengths with handling of image classification, EfficientNetB1, ResNet50, DenseNet121 and VGG16 were implemented.

The experiment showed DenseNet121 performing better than the rest of the deep learning (Table 5). Thus, denseNet121 was selected for classification in the final model as specified in the project workflow. DenseNet-121 has shown in various computer vision tasks, such as image classification, object detection, and image segmentation. The architecture of DenseNet-121 allows for better gradient flow throughout the network, reducing the vanishing gradient problem. Also, its dense connections promote feature reuse, allowing the network to learn more compact and efficient representations. This makes it suitable for the task of defect detection in steel plate images. Specifically, the DenseNet-121 algorithm was used for binary and multi-class classification in main analysis.

Table 5. Result of initial experimentation (Classification).

	EfficientNetB1	ResNet50	DenseNet121	VGG16
Accuracy	0.9128	0.9201	0.9234	0.7259

4.2. Initial Experimentation: Segmentation

Similarly, U-Net and DeepLabV3 were tested for segmentation. The architecture of U-Net is characterized by its U-shaped structure. The “U” shape consists of a contracting path and an expansive path that allow the model to capture both context and precise localization of images during segmentation [68]. U-Net is adaptable, efficient and high performing [69] [70]. DeepLab is combined neural networks specifically designed for semantic image segmentation [71] [72]. Implementing these two models achieved results in Table 6.

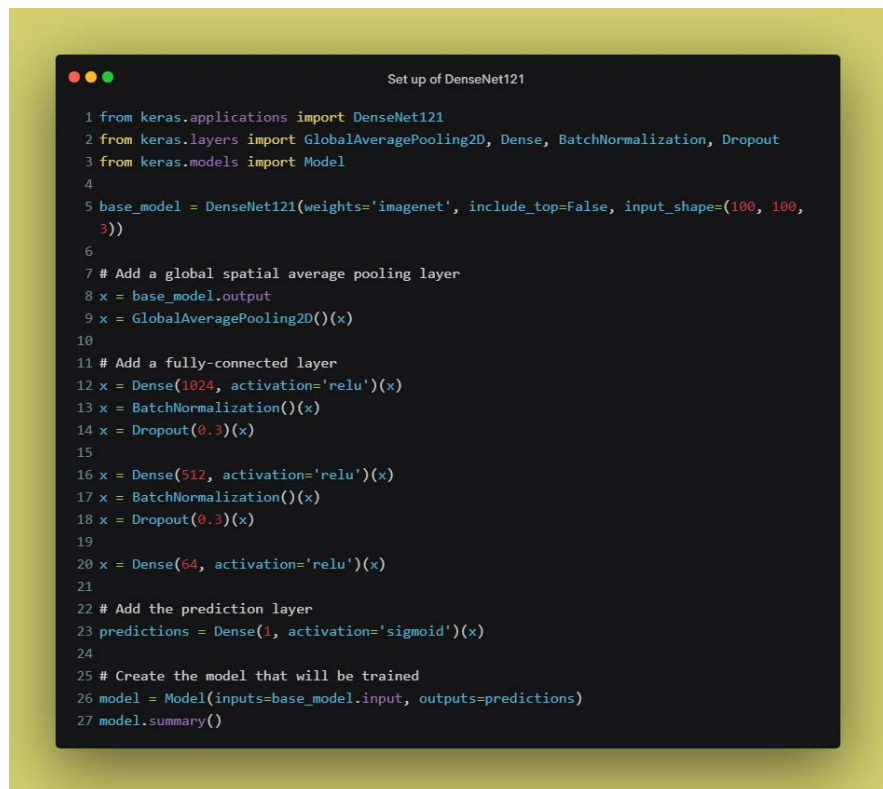
Table 6. Result of initial experimentation (Segmentation).

	U-Net	DeepLab
Dice Coefficient	0.7220	0.8518

4.3. Main Modelling: Model Training

Based on the outcomes of initial experimentation, DenseNet121 and DeepLab is

selected to build the classification and segmentation models in the system workflow. The input of DenseNet121 is configured with shape of 100 by 100 pixels with 3 colour channels (RGB) and set up with five layers (Figure 5). The whole layers are divided into DenseBlocks where average pooling layer average each of the feature map and reduce spatial dimensions. Each of the fully-connected (dense) layers is followed by ReLu activation function. Batch normalization is applied to normalize the activations to stabilize the model and improve speed.



```

Set up of DenseNet121
1 from keras.applications import DenseNet121
2 from keras.layers import GlobalAveragePooling2D, Dense, BatchNormalization, Dropout
3 from keras.models import Model
4
5 base_model = DenseNet121(weights='imagenet', include_top=False, input_shape=(100, 100,
6 3))
7 # Add a global spatial average pooling layer
8 x = base_model.output
9 x = GlobalAveragePooling2D()(x)
10
11 # Add a fully-connected layer
12 x = Dense(1024, activation='relu')(x)
13 x = BatchNormalization()(x)
14 x = Dropout(0.3)(x)
15
16 x = Dense(512, activation='relu')(x)
17 x = BatchNormalization()(x)
18 x = Dropout(0.3)(x)
19
20 x = Dense(64, activation='relu')(x)
21
22 # Add the prediction layer
23 predictions = Dense(1, activation='sigmoid')(x)
24
25 # Create the model that will be trained
26 model = Model(inputs=base_model.input, outputs=predictions)
27 model.summary()

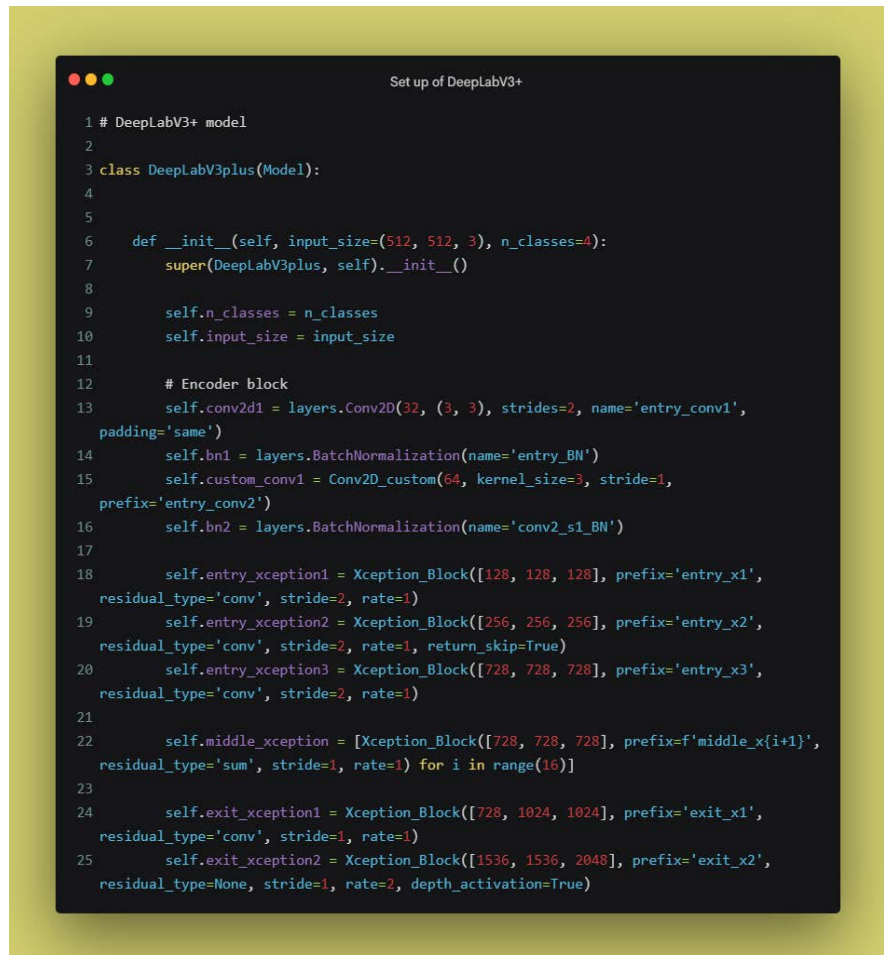
```

Figure 5. Python code to execute DenseNet121 (Google Colab).

A drop-out of 0.3 was applied after the first two dense layers thus during model training, 30% of the units are randomly fix to zero in order to stop overfitting. Sigmoid activation function which is perfect for binary classification and is added at final dense layer. The DenseNet-121 model was trained on the pre-processed dataset. The training process involved backpropagation and gradient descent to update the model's parameters. The model was trained to learn to identify and categorize the defects present in the images of steel plates.

The DeepLabV3+ model was setup after initializing the input size as (512, 512, 3) and the number of output classes as **n_classes**. The **n_classes** in the DeepLabV3+ model is applied to make the model flexible and adaptable to various defect classes in the dataset (Figure 6). This design makes the model generalizable. The encoder block was configured with a Conv2D layer with 32 filters, a kernel size of (3, 3), and stride 2, followed by a custom convolutional layer (Conv2D_custom) with 64 filters, a kernel size of (1, 3), and stride 1, and a BatchNormalization layer.

Xception blocks are applied for different sizes like (128, 128, 128), (256, 256, 256), and (728, 728, 728), with strides depending on the layer. The SeparableConv2D layers include filters of 256, with dilation rates of 6, 12, and 18.



```

Set up of DeepLabV3+

1 # DeepLabV3+ model
2
3 class DeepLabV3plus(Model):
4
5
6 def __init__(self, input_size=(512, 512, 3), n_classes=4):
7     super(DeepLabV3plus, self).__init__()
8
9     self.n_classes = n_classes
10    self.input_size = input_size
11
12    # Encoder block
13    self.conv2d1 = layers.Conv2D(32, (3, 3), strides=2, name='entry_conv1',
padding='same')
14    self.bn1 = layers.BatchNormalization(name='entry_BN')
15    self.custom_conv1 = Conv2D_custom(64, kernel_size=3, stride=1,
prefix='entry_conv2')
16    self.bn2 = layers.BatchNormalization(name='conv2_s1_BN')
17
18    self.entry_xception1 = Xception_Block([128, 128, 128], prefix='entry_x1',
residual_type='conv', stride=2, rate=1)
19    self.entry_xception2 = Xception_Block([256, 256, 256], prefix='entry_x2',
residual_type='conv', stride=2, rate=1, return_skip=True)
20    self.entry_xception3 = Xception_Block([728, 728, 728], prefix='entry_x3',
residual_type='conv', stride=2, rate=1)
21
22    self.middle_xception = [Xception_Block([728, 728, 728], prefix=f'middle_x{i+1}',
residual_type='sum', stride=1, rate=1) for i in range(16)]
23
24    self.exit_xception1 = Xception_Block([728, 1024, 1024], prefix='exit_x1',
residual_type='conv', stride=1, rate=1)
25    self.exit_xception2 = Xception_Block([1536, 1536, 2048], prefix='exit_x2',
residual_type=None, stride=1, rate=2, depth_activation=True)

```

Figure 6. The Python code extract for DeepLabV3+ Set up (Google Colab).

4.4. Integration: Classifier Probability and Thresholding Masks

The predicted class of defects from classifiers are fed into segmentation through decision flow of 2nd/98th percentile rule. This allow human decisions as areas outside the range between the 2nd percentile and the 98th percentile in the predicted masks are neglected. With application of thresholding, model performance is improved (reliability and robustness) by focusing on the area that is more concern with the study predictions and discard other regions.

4.5. Model Evaluation

After training, the classification model's performance was evaluated using various metrics, including precision, recall, F1-score, and accuracy (**Figure 7**). These metrics provided a quantitative measure of the model's ability to accurately identify and categorize the defects in the steel plate images.

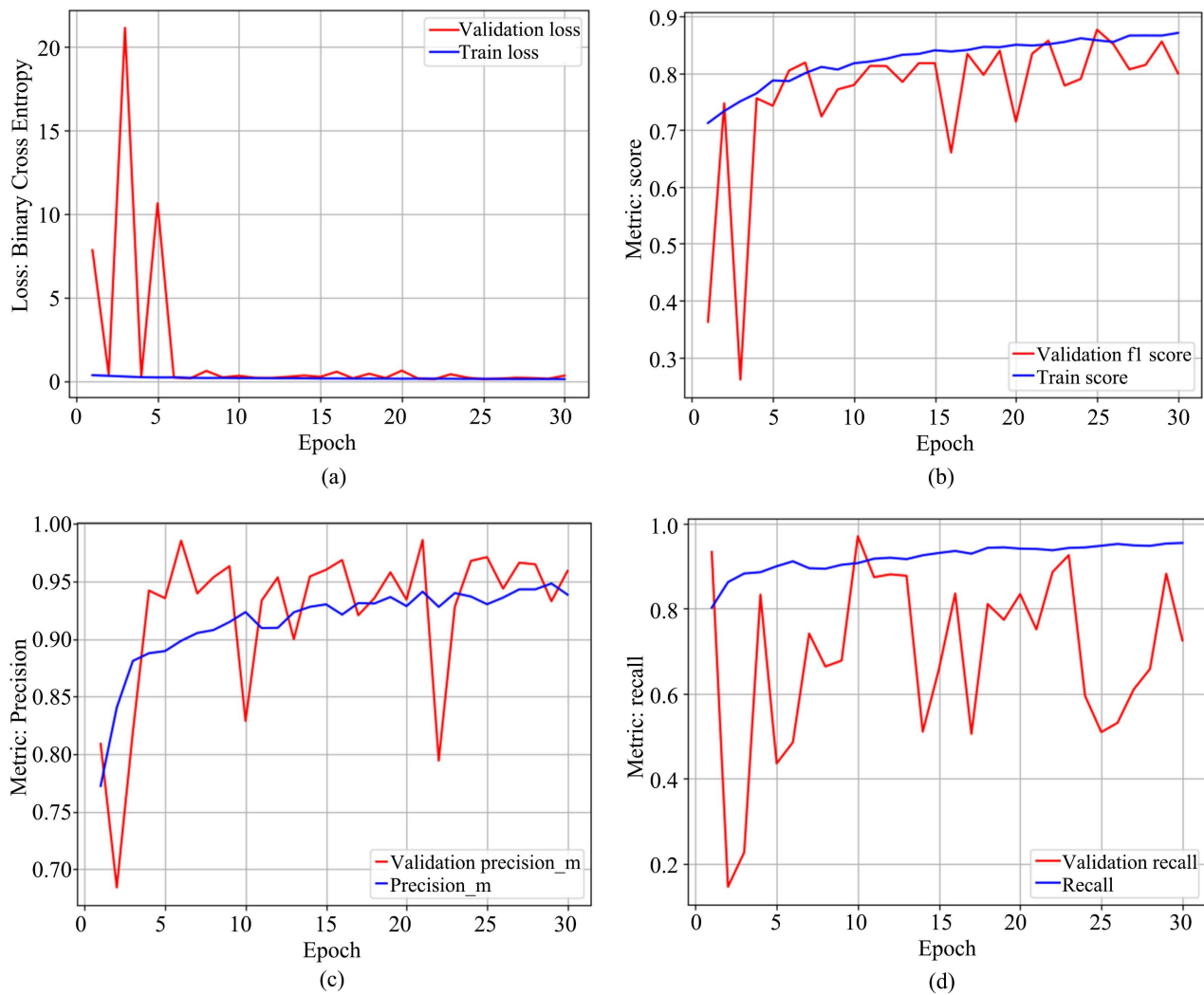


Figure 7. Result of DenseNet121 classification.

The segmentation models are trained using the Adam optimizer and the Dice loss function. The Dice coefficient is used as a metric for model evaluation. The models are trained for a specified number of epochs, with the best model saved based on the maximum Dice coefficient achieved on the validation set. Each trained model is then evaluated on the train, validation, and test sets. The evaluation scores (Dice loss and Dice coefficient) for each defect class are displayed (**Figure 8**). Additionally, the model's predictions on a subset of the train, validation, and test datasets are visualized by displaying the original image, the ground truth mask, and the predicted mask side by side (**Figures 9-12**).

These models were trained and evaluated separately for each type of defect.

5. Result and Discussion

The DenseNet-121 model was trained and evaluated on the Severstal steel defect detection task. The model demonstrated promising results, successfully identifying and categorizing defects in the steel plate images. The performance of the

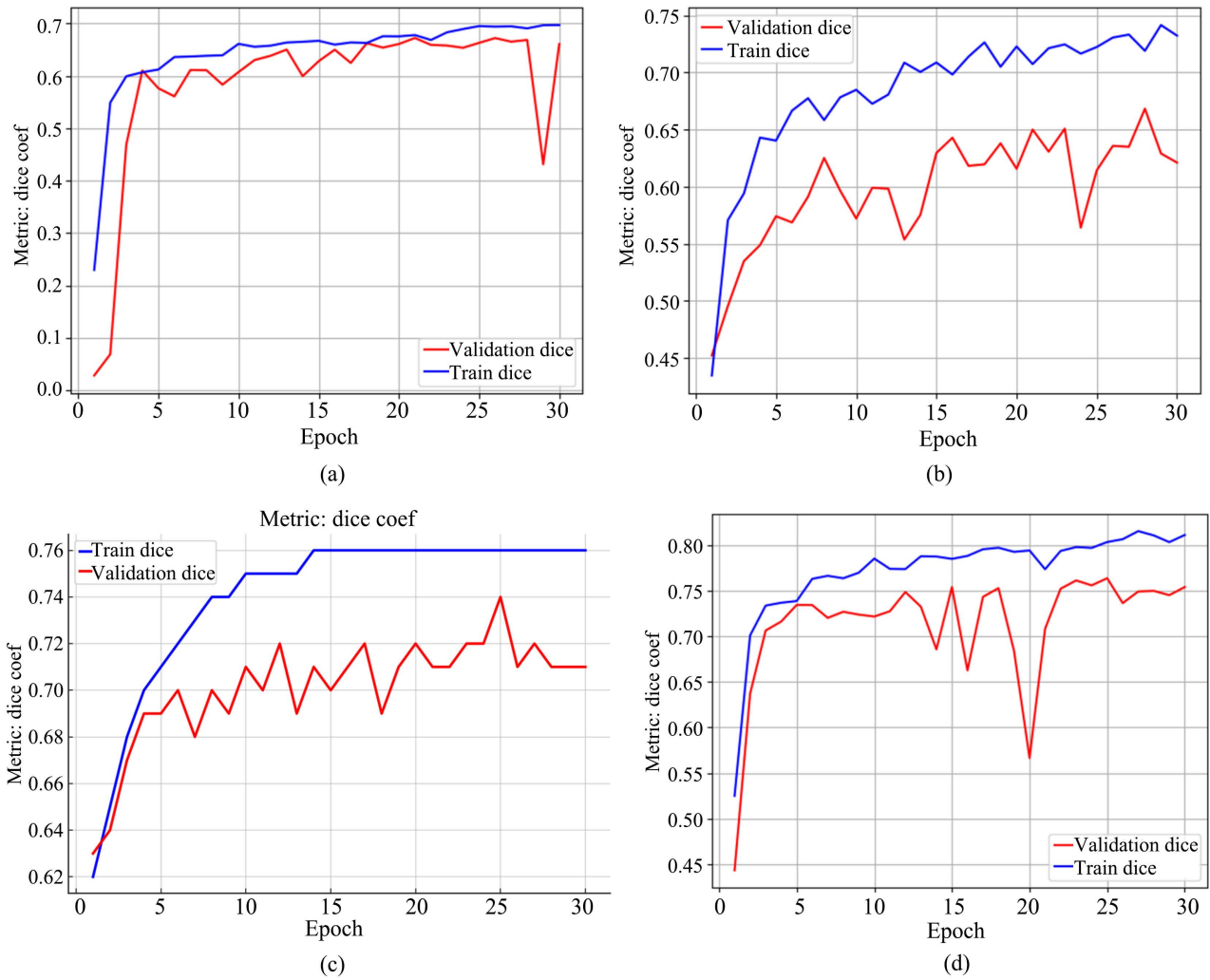


Figure 8. Performance for segmentation models for Defect 1, 2, 3 and 4.

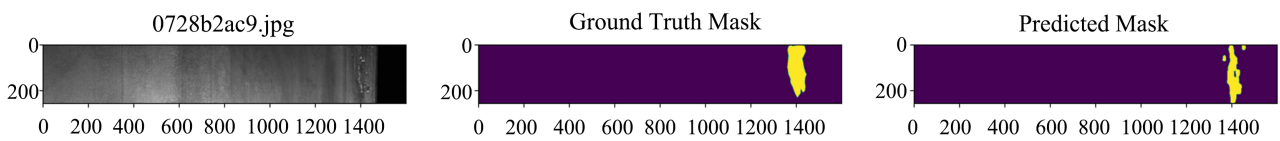


Figure 9. Original image, the ground truth mask, and the predicted mask (Defect 1).

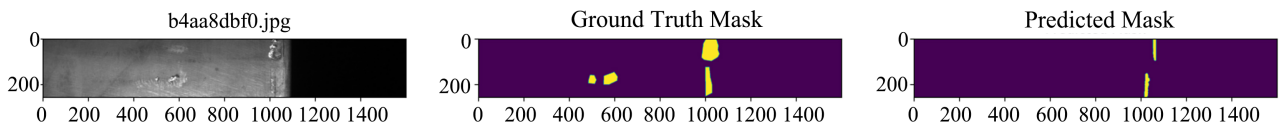


Figure 10. Original image, the ground truth mask, and the predicted mask (Defect 2).

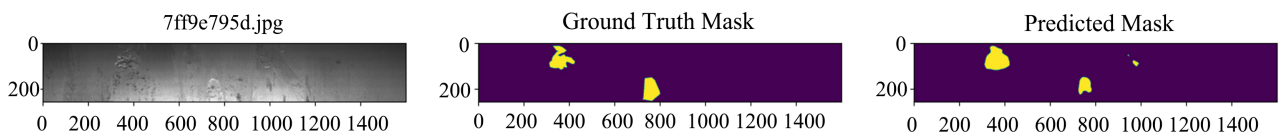


Figure 11. Original image, the ground truth mask, and the predicted mask (Defect 3).

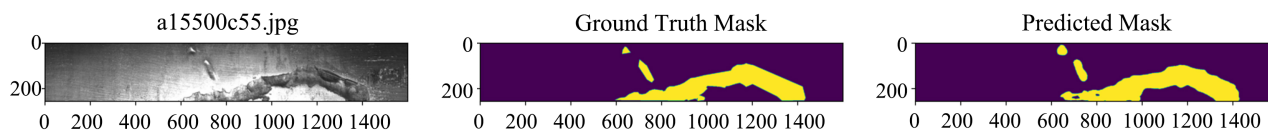


Figure 12. Original image, the ground truth mask, and the predicted mask (Defect 4).

model was evaluated using several metrics, including precision, recall, F1-score, and accuracy. These metrics were calculated for each class of defects, providing a detailed view of the model's performance across different types of defects. The precision, recall, and F1-score provided insights into the model's ability to correctly identify the presence of a defect and correctly categorize it. The accuracy metric provided an overall view of the model's performance across all classes. The results showed that the model was able to achieve a high level of accuracy, demonstrating its effectiveness in identifying and categorizing defects in steel plate images.

The model performs well across all three data sets (**Table 7**), with slightly lower accuracy on the validation and testing sets. The values of F1 are consistent with the accuracy metrics, with a slight drop in performance from training to testing. The models have high precision, especially in the validation set. The recall is slightly lower in validation and testing sets, indicating that the model might miss some positive cases. The DeepLab for segmentation demonstrated promising results (**Table 8**) in identifying, categorizing, and localizing defects in the steel plate images.

Table 7. Classification Results for Train, Validation and Test dataset.

	Accuracy	Binary Cross_Entropy	F1_Score	Precision	Recall
Training	0.9135	0.2042	0.9152	0.9177	0.9231
Validation	0.9032	0.2383	0.9002	0.9503	0.8643
Testing	0.8990	0.2301	0.8910	0.9376	0.8591

Table 8. Segmentation Results for Train, Validation and Test dataset.

	Dice_Coefficient	Dice_loss
Training	0.8421	0.1696
Validation	0.6977	0.3331

5.1. Dice Coefficient for Defect Class

The coefficients range from 64.81% to 87.69% across different datasets and defect classes, indicating varying performance for different defects (**Table 9**). Some insights can be drawn. Defect 4 consistently has the highest Dice Coefficients across all sets, suggesting that the model is best at detecting this defect. Defect 1 has the lowest Dice Coefficient in the testing set, hinting at potential challenges in detecting this specific defect.

Table 9. Dice Coefficient for each defect class.

	Defect 1	Defect 2	Defect 3	Defect 4
Training	0.7262	0.8596	0.7366	0.8112
Validation	0.6721	0.8451	0.7160	0.7639
Testing	0.6481	0.8769	0.7109	0.7877

5.2. Models Comparison

Comparing this model with other commonly used models in the field of image recognition, such as VGG16, ResNet, DenseNet-121 showed superior performance. The superior performance of DenseNet-121 can be attributed to its unique architecture, where each layer is connected to every other layer in a feed-forward fashion, allowing for better gradient flow and feature reuse. This architecture enables DenseNet-121 to learn more compact and efficient representations, making it more suitable for complex tasks like defect detection in steel plate images. Different variations of the DeepLab models. The use of DeepLab V3 provides the benefits of a powerful, efficient convolutional network that scales well with increasing amounts of data and computational resources. Each model was trained separately to detect a specific type of defect in the steel plates. The model's performance was evaluated using the Dice coefficient, a popular metric for image segmentation tasks that measures the overlap between the predicted and actual results. High Dice coefficient value (**Table 10**) indicate a high degree of overlap and, thus, successful defect detection. The models demonstrated effective learning, with their performance improving over successive training epochs. The visualization of the model's predictions further confirmed their ability to accurately detect and categorize defects.

Table 10. Comparing study models with benchmarked literature results (Accuracy and Dice Coefficient).

	EfficientNetB1 (Acc)	ResNet50 (Acc)	DenseNet121 (Acc)	VGG16 (Acc)	DeepLab (Dice)
Study Model	0.9128	0.9201	0.9234	0.7259	0.8421
Amin & Akhter 2020 [73]					0.5430
Abu <i>et al.</i> 2021 [25]		0.7050 (ResNet101-CPU)	0.7034 (CPU)	0.5000	
		0.7235 (ResNet101-GPU)	0.7027 (GPU)		

VGG16 showed lowest accuracy due to its smallest number of layers embedded in its configuration. Though observation indicates that larger layer sizes do not ensure improve accuracy value. The study models achieved higher accuracies compared to [25] outcomes as configurations were specifically tailored towards

improved performance with optimization of models' parameters (pooling, activation functions, batch normalization).

The study segmentation model performed better than benchmarked paper from literature review, Amin and Akhter 2020. DeepLab V3 delivered higher dice coefficients across classes of the defects while Amin and Akhter 2020 model exhibited imbalanced performance across defect classes and failed to predict defects for classes 1, 2 and 4.

The proposed methodology offers several improvements over existing benchmarked works in defect detection. Deep learning models like VGG16 and ResNet101 used in research by [25], yielded lower accuracies of 50% and 70.5%, respectively. In contrast, the proposed system utilizing DenseNet121 achieves a significantly higher accuracy of 92.34% (Table 11). This demonstrates its superior capability with learning and identification of defect patterns. The DenseNet121 architecture outperforms other deep learning techniques by leveraging dense connections between layers, utilization of feature reuse, and improvement of gradient flow. All these allow DenseNet121 to mitigate vanishing gradient problem better. Also, the overall system in this research integrates multi-label classification and segmentation, providing both identification and precise localization of defects. This combination surpasses models that focus solely on classification like those by [22] and [25] which they achieved accuracies of 79.91% with MobileNet and 96.9% with the FDD model. Their models lacked segmentation capabilities, and this limits the practical application when localization of defect is a top target. Based on the initial experimentation and final modelling, the use of DeepLab V3 for segmentation in the current research further improves performance. The Dice coefficient for segmentation ranges from 64.81% to 87.69% across different defect classes. This provides more robust and reliable detection compared to previous approaches in research by [73] where their model only achieved a Dice coefficient of 54.30%. DenseNet121 maintains a higher feature map resolution in its configuration while the is densely connected. This allows DenseNet121 to capture more detailed spatial information and produced higher precision of 92.31% compared to YOLO models that downsample the input image multiple times and achieved highest precision of 65.20% in the work of Akhyar *et al.* (2023) [22]. Generally, the proposed methodology adopted in this research excels by offering a more integrated approach with the combination of classification and segmentation to improve accuracy and precision that surpasses models in existing works.

Table 11. Comparing study models with benchmarked literature results (Precision).

	YOLOv4 (Precision)	YOLOv5 (Precision)	YOLOX (Precision)	Ca scade R-CNN (Precision)	FDD (Precision)	DenseNet (Precision)
Akhyar <i>et al.</i> 2023 [22]	0.6080	0.6010	0.6520	0.6750	0.7830	
Study Model						0.9231

6. Conclusion and Future Work

The system adopted in this research addresses the issues of subjectivity and human error prevalent in traditional defect detection methods with the incorporation of automation through deep learning techniques. As detailed from background review, traditional defect detection methods like manual measurement and physical inspection are highly dependent on human judgment. This leads to inconsistencies particularly when well-hidden defects are targeted. Also, manual techniques are prone to operator fatigue, skill variability, and bias. These significantly reduce accuracy and repeatability. The system developed in this research utilizes combination of DenseNet121 for classification and DeepLabV3 for segmentation to automate the entire defect detection process. This deep learning system is trained on large datasets by Severstal dataset. This allows the system to learn and identify defects autonomously without the need for subjective human judgment. The automation offered by these models eliminates human-related errors and introduces improved level of consistency that is impossible with manual methods. DenseNet121, with its dense connections, allows for better feature reuse and gradient flow and produces higher accuracy with the identification and classification of defects. Similarly, segmentation using DeepLabV3 provides precise localization of defects and further improves accuracy with the visualization of the exact position of the defects within an image. This two-stage approach, integrating classification with segmentation, improves the system's ability to accurately detect and localize defect while reducing false positives and negatives that are common in manual inspections. Also, the system applies thresholding techniques that discard low-confidence predictions and improve the reliability of defect identification.

Future Work:

1) Multimodal Learning: in a situation where more diverse datasets are available, future work should delve into multimodal learning such that models are trained to utilize information from different data sources (like images, sensors, and texts) simultaneously to create more robust and reliable defect detection ML system.

2) System with Feedback Loops: Implementation of systems such that outputs of the machine learning models are continuously inputted for retraining to ensure that the models will evolve and adapt to changing production dynamics and new types of defects.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Mägi, R. (2007) Engineering Education and Errors. In: Elleithy, K., Sobh, T., Mahmood, A., Iskander, M. and Karim, M., Eds., *Advances in Computer, Information, and Systems Sciences, and Engineering*, Springer, 387-391.
https://doi.org/10.1007/1-4020-5261-8_60

- [2] Liu, G.J., Zhang, Y., Qiu, J. and Lu, K.H. (2009) Automatic Image Measurement Technology in Endoscopy NDT. *Key Engineering Materials*, **413**, 803-809. <https://doi.org/10.4028/www.scientific.net/kem.413-414.803>
- [3] Karpinski, K., Akguen, D., Gebauer, H., Paksoy, A., Lupetti, M., Markova, V., *et al.* (2024) Reliability of Manual Measurements versus Semiautomated Software for Glenoid Bone Loss Quantification in Patients with Anterior Shoulder Instability. *Orthopaedic Journal of Sports Medicine*, **12**, 1-7. <https://doi.org/10.1177/23259671231222938>
- [4] Giap, D.B., Ngoc Le, T., Wang, J. and Wang, C. (2021) Wavelet Subband-Based Tensor for Smartphone Physical Button Inspection. *IEEE Access*, **9**, 107399-107415. <https://doi.org/10.1109/access.2021.3099965>
- [5] Xie, Y., Ye, Y., Zhang, J., Liu, L. and Liu, L. (2014) A Physics-Based Defects Model and Inspection Algorithm for Automatic Visual Inspection. *Optics and Lasers in Engineering*, **52**, 218-223. <https://doi.org/10.1016/j.optlaseng.2013.06.006>
- [6] Charlesworth, J.P. and Temple, J.A. (2001) Engineering Applications of Ultrasonic Time of Flight Diffraction. Research Studies Press.
- [7] Drinkwater, B.W. and Wilcox, P.D. (2006) Ultrasonic Arrays for Non-Destructive Evaluation: A Review. *NDT & E International*, **39**, 525-541. <https://doi.org/10.1016/j.ndteint.2006.03.006>
- [8] Gong, X., Bai, Y., Liu, Y. and Mu, H. (2020) Application of Deep Learning in Defect Detection. *Journal of Physics: Conference Series*, **1684**, Article ID: 012030. <https://doi.org/10.1088/1742-6596/1684/1/012030>
- [9] Sankhye, S. and Hu, G. (2020) Machine Learning Methods for Quality Prediction in Production. *Logistics*, **4**, Article 35. <https://doi.org/10.3390/logistics4040035>
- [10] Khalfaoui, S., Manouvrier, E., Briot, A., Delaux, D., Butel, S., Ibrahim, J., *et al.* (2021) Defect Prediction on Production Line. In: Jansen, T., Jensen, R., Mac Parthaláin, N. and Lin, C.M., Eds., *Advances in Intelligent Systems and Computing*, Springer International Publishing, 532-544. https://doi.org/10.1007/978-3-030-87094-2_47
- [11] Schuh, G., Gützlaff, A., Thomas, K. and Welsing, M. (2021) Machine Learning Based Defect Detection in a Low Automated Assembly Environment. *Procedia CIRP*, **104**, 265-270. <https://doi.org/10.1016/j.procir.2021.11.045>
- [12] Janiesch, C., Zschech, P. and Heinrich, K. (2021) Machine Learning and Deep Learning. *Electronic Markets*, **31**, 685-695. <https://doi.org/10.1007/s12525-021-00475-2>
- [13] Neeraj, N., Kumar, N. and Maurya, V.K. (2020) A Review on Machine Learning (Feature Selection, Classification and Clustering) Approaches of Big Data Mining in Different Areas of Research. *Journal of Critical Reviews*, **19**, 2610-2626.
- [14] Psarommatis, F., Sousa, J., Mendonça, J.P. and Kiritsis, D. (2021) Zero-Defect Manufacturing the Approach for Higher Manufacturing Sustainability in the Era of Industry 4.0: A Position Paper. *International Journal of Production Research*, **60**, 73-91. <https://doi.org/10.1080/00207543.2021.1987551>
- [15] Mumuni, A. and Mumuni, F. (2022) Data Augmentation: A Comprehensive Survey of Modern Approaches. *Array*, **16**, Article ID: 100258. <https://doi.org/10.1016/j.array.2022.100258>
- [16] Paullada, A., Raji, I.D., Bender, E.M., Denton, E. and Hanna, A. (2021) Data and Its (Dis)contents: A Survey of Dataset Development and Use in Machine Learning Research. *Patterns*, **2**, Article ID: 100336. <https://doi.org/10.1016/j.patter.2021.100336>
- [17] Althnian, A., AlSaeed, D., Al-Baity, H., Samha, A., Dris, A.B., Alzakari, N., *et al.* (2021) Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the

- Medical Domain. *Applied Sciences*, **11**, Article 796. <https://doi.org/10.3390/app11020796>
- [18] Islam, T., Hafiz, M.S., Jim, J.R., Kabir, M.M. and Mridha, M.F. (2024) A Systematic Review of Deep Learning Data Augmentation in Medical Imaging: Recent Advances and Future Research Directions. *Healthcare Analytics*, **5**, Article ID: 100340. <https://doi.org/10.1016/j.health.2024.100340>
- [19] Ferguson, M., Ak, R., Lee, Y.T. and Law, K.H. (2018) Detection and Segmentation of Manufacturing Defects with Convolutional Neural Networks and Transfer Learning. *Smart and Sustainable Manufacturing Systems*, **2**, 137-164. <https://doi.org/10.1520/ssms20180033>
- [20] Manakitsa, N., Maraslidis, G.S., Moysis, L. and Fragulis, G.F. (2024) A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision. *Technologies*, **12**, Article 15. <https://doi.org/10.3390/technologies12020015>
- [21] Zhang, Z., Wen, G. and Chen, S. (2019) Weld Image Deep Learning-Based Online Defects Detection Using Convolutional Neural Networks for Al Alloy in Robotic Arc Welding. *Journal of Manufacturing Processes*, **45**, 208-216.
- [22] Akhyar, F., Liu, Y., Hsu, C., Shih, T.K. and Lin, C. (2023) FDD: A Deep Learning-based Steel Defect Detectors. *The International Journal of Advanced Manufacturing Technology*, **126**, 1093-1107. <https://doi.org/10.1007/s00170-023-11087-9>
- [23] He, T., Liu, Y., Xu, C., Zhou, X., Hu, Z. and Fan, J. (2019) A Fully Convolutional Neural Network for Wood Defect Location and Identification. *IEEE Access*, **7**, 123453-123462. <https://doi.org/10.1109/access.2019.2937461>
- [24] Son, L.H., Tripathy, H.K., Acharya, B.R., Kumar, R. and Chatterjee, J.M. (2018) Machine Learning on Big Data: A Developmental Approach on Societal Applications. In: Mittal, M., Balas, V., Goyal, L. and Kumar, R., Eds., *Big Data Processing Using Spark in Cloud*, Springer, 143-165. https://doi.org/10.1007/978-981-13-0550-4_7
- [25] Abu, M., Amir, A., Lean, Y.H., Zahri, N.A.H. and Azemi, S.A. (2021) The Performance Analysis of Transfer Learning for Steel Defect Detection by Using Deep Learning. *Journal of Physics: Conference Series*, **1755**, Article ID: 012041. <https://doi.org/10.1088/1742-6596/1755/1/012041>
- [26] Ferraz Júnior, F., Romero, R.A.F. and Hsieh, S. (2023) Machine Learning for the Detection and Diagnosis of Anomalies in Applications Driven by Electric Motors. *Sensors*, **23**, Article 9725. <https://doi.org/10.3390/s23249725>
- [27] Wu, H. and Zhou, Z. (2021) Using Convolution Neural Network for Defective Image Classification of Industrial Components. *Mobile Information Systems*, **2021**, Article ID: 9092589. <https://doi.org/10.1155/2021/9092589>
- [28] Yang, J., Li, S., Wang, Z., Dong, H., Wang, J. and Tang, S. (2020) Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges. *Materials*, **13**, Article 5755. <https://doi.org/10.3390/ma13245755>
- [29] Marbade, A.R. and Bhelkar, A.M. (2022) Intelligent Part Recognition, Classification and Surface Defect Detection by Using Computer Vision (CV) & Machine Learning (ML). *International Journal of Creative Research Thoughts (IJCRT)*, **10**, c361-c367.
- [30] Wang, L., Wu, J. and Wu, D. (2020) Research on Vehicle Parts Defect Detection Based on Deep Learning. *Journal of Physics: Conference Series*, **1437**, Article ID: 012004. <https://doi.org/10.1088/1742-6596/1437/1/012004>
- [31] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv: 200410934.

- [32] GitHub (2021) YOLOV5-Master. <https://github.com/ultralytics/yolov5.git/>
- [33] Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021) YOLOX: Exceeding YOLO Series in 2021. arXiv: 210708430.
- [34] Qiao, S., Chen, L. and Yuille, A. (2021) DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, 20-25 June 2021, 10208-10219. <https://doi.org/10.1109/cvpr46437.2021.01008>
- [35] Njeri, N.R. (2022) Data Preparation for Machine Learning Modelling. *International Journal of Computer Applications Technology and Research*, **11**, 231-235. <https://doi.org/10.7753/ijcatr1106.1008>
- [36] Fernandes, A.A.A., Koehler, M., Konstantinou, N., Pankin, P., Paton, N.W. and Sakellariou, R. (2023) Data Preparation: A Technological Perspective and Review. *SN Computer Science*, **4**, Article No. 425. <https://doi.org/10.1007/s42979-023-01828-8>
- [37] Grishin, A. (2019) Severstal: Steel Defect Detection. Kaggle. <https://kaggle.com/competitions/severstal-steel-defect-detection>
- [38] Jayatilake, S.M.D.A.C. and Ganegoda, G.U. (2021) Involvement of Machine Learning Tools in Healthcare Decision Making. *Journal of Healthcare Engineering*, **2021**, Article ID: 6679512. <https://doi.org/10.1155/2021/6679512>
- [39] Sarker, I.H. (2021) Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, **2**, Article No. 420. <https://doi.org/10.1007/s42979-021-00815-1>
- [40] Alnuaimi, A.F.A.H. and Albaldawi, T.H.K. (2024) An Overview of Machine Learning Classification Techniques. *BIO Web of Conferences*, **97**, Article No. 00133. <https://doi.org/10.1051/bioconf/20249700133>
- [41] Morán-Fernández, L. and Bolón-Canedo, V. (2023) Finding a Needle in a Haystack: Insights on Feature Selection for Classification Tasks. *Journal of Intelligent Information Systems*, **62**, 459-483. <https://doi.org/10.1007/s10844-023-00823-y>
- [42] Seo, H., Badiel Khuzani, M., Vasudevan, V., Huang, C., Ren, H., Xiao, R., *et al.* (2020) Machine Learning Techniques for Biomedical Image Segmentation: An Overview of Technical Aspects and Introduction to State-of-Art Applications. *Medical Physics*, **47**, e148-e167. <https://doi.org/10.1002/mp.13649>
- [43] Rizwan, I., Haque, I. and Neubert, J. (2020) Deep Learning Approaches to Biomedical Image Segmentation. *Informatics in Medicine Unlocked*, **18**, Article ID: 100297. <https://doi.org/10.1016/j.imu.2020.100297>
- [44] Rayed, M.E., Islam, S.M.S., Niha, S.I., Jim, J.R., Kabir, M.M. and Mridha, M.F. (2024) Deep Learning for Medical Image Segmentation: State-Of-The-Art Advancements and Challenges. *Informatics in Medicine Unlocked*, **47**, Article ID: 101504. <https://doi.org/10.1016/j.imu.2024.101504>
- [45] Archana, R. and Jeevaraj, P.S.E. (2024) Deep Learning Models for Digital Image Processing: A Review. *Artificial Intelligence Review*, **57**, Article No. 11. <https://doi.org/10.1007/s10462-023-10631-z>
- [46] Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S. and Jorge Cardoso, M. (2017) Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In: Cardoso, M., *et al.*, Eds., *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 240-248. https://doi.org/10.1007/978-3-319-67558-9_28
- [47] Rezaei, F., Izadi, H., Memarian, H. and Baniassadi, M. (2019) The Effectiveness of

- Different Thresholding Techniques in Segmenting Micro CT Images of Porous Carbonates to Estimate Porosity. *Journal of Petroleum Science and Engineering*, **177**, 518-527. <https://doi.org/10.1016/j.petrol.2018.12.063>
- [48] Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T. and Ronneberger, O. (2016) 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M., Unal, G. and Wells, W., Eds., *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2016*, Springer, 424-432. https://doi.org/10.1007/978-3-319-46723-8_49
- [49] Ibrahim, A.M. and Bennett, B. (2014) The Assessment of Machine Learning Model Performance for Predicting Alluvial Deposits Distribution. *Procedia Computer Science*, **36**, 637-642. <https://doi.org/10.1016/j.procs.2014.09.067>
- [50] Antoniou, T. and Mamdani, M. (2021) Evaluation of Machine Learning Solutions in Medicine. *Canadian Medical Association Journal*, **193**, E1425-E1429. <https://doi.org/10.1503/cmaj.210036>
- [51] Erickson, B.J. and Kitamura, F. (2021) Magician’s Corner: 9. Performance Metrics for Machine Learning Models. *Radiology: Artificial Intelligence*, **3**, e200126. <https://doi.org/10.1148/ryai.2021200126>
- [52] Rainio, O., Teuho, J. and Klén, R. (2024) Evaluation Metrics and Statistical Tests for Machine Learning. *Scientific Reports*, **14**, Article No. 6086. <https://doi.org/10.1038/s41598-024-56706-x>
- [53] Powers, D. and Ailab, H. (2011) Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technology*, **2**, 2229-3981.
- [54] Zou, K.H., Warfield, S.K., Bharatha, A., Tempany, C.M.C., Kaus, M.R., Haker, S.J., *et al.* (2004) Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index1. *Academic Radiology*, **11**, 178-189. [https://doi.org/10.1016/s1076-6332\(03\)00671-8](https://doi.org/10.1016/s1076-6332(03)00671-8)
- [55] Wong, Y.M., Yeap, P.L., Ong, A.L.K., Tuan, J.K.L., Lew, W.S., Lee, J.C.L., *et al.* (2024) Machine Learning Prediction of Dice Similarity Coefficient for Validation of Deformable Image Registration. *Intelligence-Based Medicine*, **10**, Article ID: 100163. <https://doi.org/10.1016/j.ibmed.2024.100163>
- [56] Wongpanich, A., Pham, H., Demmel, J., Tan, M., Le, Q., You, Y., *et al.* (2021) Training Efficient Nets at Supercomputer Scale: 83% ImageNet Top-1 Accuracy in One Hour. 2021 *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, Portland, 17-21 June 2021, 947-950. <https://doi.org/10.1109/ipdpsw52791.2021.00146>
- [57] Tajane, K., Rathkanthiwar, V., Chava, G., Dhavale, S., Chawda, G. and Pitale, R. (2023) EffiConvRes: An Efficient Convolutional Neural Network with Residual Connections and Depthwise Convolutions. 2023 *7th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, 18-19 August 2023, 1-5. <https://doi.org/10.1109/iccubea58933.2023.10392177>
- [58] Hossain, M.B., Iqbal, S.M.H.S., Islam, M.M., Akhtar, M.N. and Sarker, I.H. (2022) Transfer Learning with Fine-Tuned Deep CNN ResNet50 Model for Classifying COVID-19 from Chest X-Ray Images. *Informatics in Medicine Unlocked*, **30**, Article ID: 100916. <https://doi.org/10.1016/j.imu.2022.100916>
- [59] Singla, M., Gill, K.S., Chauhan, R. and Pokhariya, H.S. (2024) ResNet50 Utilization for Bag Classification: A CNN Model Visualization Approach in Deep Learning. 2024 *IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, Bangalore, 28-29 June 2024, 1-4.

- <https://doi.org/10.1109/iciteics61368.2024.10624847>
- [60] Zhou, T., Ye, X., Lu, H., Zheng, X., Qiu, S. and Liu, Y. (2022) Dense Convolutional Network and Its Application in Medical Image Analysis. *BioMed Research International*, **2022**, Article ID: 2384830. <https://doi.org/10.1155/2022/2384830>
- [61] Ju, R., Lin, T., Jian, J., Chiang, J. and Yang, W. (2022) ThreshNet: An Efficient DenseNet Using Threshold Mechanism to Reduce Connections. *IEEE Access*, **10**, 82834-82843. <https://doi.org/10.1109/access.2022.3196492>
- [62] Jia, J., Lei, R., Qin, L., Wu, G. and Wei, X. (2023) iEnhancer-DCSV: Predicting Enhancers and Their Strength Based on Densenet and Improved Convolutional Block Attention Module. *Frontiers in Genetics*, **14**, Article 1132018. <https://doi.org/10.3389/fgene.2023.1132018>
- [63] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017) Densely Connected Convolutional Networks. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 21-26 July 2017, 2261-2269. <https://doi.org/10.1109/cvpr.2017.243>
- [64] Rawat, J., Logofătu, D. and Chiramel, S. (2020) Factors Affecting Accuracy of Convolutional Neural Network Using Vgg-16. In: Iliadis, L., Angelov, P., Jayne, C. and Pimenidis, E., Eds., *Proceedings of the International Neural Networks Society*, Springer, 251-260. https://doi.org/10.1007/978-3-030-48791-1_19
- [65] AKGÜN, D. (2020) An Evaluation of VGG16 Binary Classifier Deep Neural Network for Noise and Blur Corrupted Images. *Sakarya University Journal of Computer and Information Sciences*, **3**, 264-271. <https://doi.org/10.35377/saucis.03.03.725647>
- [66] Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K. (2018) Convolutional Neural Networks: An Overview and Application in Radiology. *Insights into Imaging*, **9**, 611-629. <https://doi.org/10.1007/s13244-018-0639-9>
- [67] Cao, Z., Wang, K., Wen, J., Li, C., Wu, Y., Wang, X., et al. (2024) Fine-grained Image Classification on Bats Using VGG16-CBAM: A Practical Example with 7 Horseshoe Bats Taxa (CHIROPTERA: Rhinolophidae: *Rhinolophus*) from Southern China. *Frontiers in Zoology*, **21**, Article No. 10. <https://doi.org/10.1186/s12983-024-00531-5>
- [68] Pun, N.S. and Agarwal, S. (2022) Modality Specific U-Net Variants for Biomedical Image Segmentation: A Survey. *Artificial Intelligence Review*, **55**, 5845-5889. <https://doi.org/10.1007/s10462-022-10152-1>
- [69] Siddique, N., Paheding, S., Reyes Angulo, A.A., Alom, M.Z. and Devabhaktuni, V.K. (2022) Fractal, Recurrent, and Dense U-Net Architectures with EfficientNet Encoder for Medical Image Segmentation. *Journal of Medical Imaging*, **9**, Article ID: 064004. <https://doi.org/10.1117/1.jmi.9.6.064004>
- [70] Tashk, A., Herp, J., Bjørsum-Meyer, T., Koulaouzidis, A. and Nadimi, E.S. (2022) Aid-U-Net: An Innovative Deep Convolutional Architecture for Semantic Segmentation of Biomedical Images. *Diagnostics*, **12**, Article 2952. <https://doi.org/10.3390/diagnostics12122952>
- [71] Chen, L., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H. (2018) Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV 2018*, Springer, 833-851. https://doi.org/10.1007/978-3-030-01234-2_49
- [72] Wang, Y., Wang, C., Wu, H. and Chen, P. (2022) An Improved Deeplabv3+ Semantic Segmentation Algorithm with Multiple Loss Constraints. *PLOS ONE*, **17**, e0261582. <https://doi.org/10.1371/journal.pone.0261582>

- [73] Amin, D. and Akhter, S. (2020) Deep Learning-Based Defect Detection System in Steel Sheet Surfaces. 2020 *IEEE Region 10 Symposium (TENSYP)*, Dhaka, 5-7 June 2020, 444-448. <https://doi.org/10.1109/tensymp50017.2020.9230863>