# Attention-Guided Multitask Learning for Surface Defect Identification

Vignesh Sampath, Iñaki Maurtua, Juan José Aguilar Martín, Andoni Rivera, Jorge Molina, and Aitor Gutierrez

***Abstract*—Surface defect identification is an essential task in the industrial quality control process, in which visual checks are conducted on a manufactured product to ensure that it meets quality standards. The convolutional neural network (CNN)-based surface defect identification method has proven to outperform traditional image processing techniques. However, the real-world surface defect datasets are limited in size due to the expensive data generation process and the rare occurrence of defects. To address this issue, this article presents a method for exploiting auxiliary information beyond the primary labels to improve the generalization ability of surface defect identification tasks. Considering the correlation between pixel-level segmentation masks, object-level bounding boxes, and global image-level classification labels, we argue that jointly learning features of the related tasks can improve the performance of surface defect identification tasks. This article proposes a framework named Defect-Aux-Net, based on multitask learning with attention mechanisms that exploit the rich additional information from related tasks with the goal of simultaneously improving robustness and accuracy of the CNN-based surface defect identification. We conducted a series of experiments with the proposed framework. The experimental results showed that the proposed method can significantly improve the performance of state-of-the-art models while achieving an overall accuracy of 97.1%, Dice score of 0.926, and mean average precision of 0.762 on defect classification, segmentation, and detection tasks.**

***Index Terms*—Deep learning, defect classification, defect detection, defect segmentation, machine vision, multitask learning (MTL), quality control, surface defect detection.**

Manuscript received 3 January 2022; revised 20 June 2022 and 25 September 2022; accepted 22 December 2022. Date of publication 4 January 2023; date of current version 24 July 2023. This work was supported in part by DIGIMAN4.0 Project ("Digital Manufacturing Technologies for Zero? defect", https://www.digiman4?0.mek.dtu.dk/) which is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation under Project 814225, and in part by the ELKARTEK Project KK?2020/00049 3KIA of the Basque Government. Paper no. TII-21-5940. *(Corresponding author: Vignesh Sampath.)*

Vignesh Sampath, Iñaki Maurtua, Andoni Rivera, Jorge Molina, and Aitor Gutierrez are with the Autonomous and Intelligent Systems Unit, Tekniker, 20600 Gipuzkoa, Spain (e-mail: am14m016@gmail.com; inaki.maurtua@tekniker.es; andoni.rivera@tekniker.es; jorge.molina@tekniker.es; aitor.gutierrez@tekniker.es).

Juan José Aguilar Martín is with the Department of Design and Manufacturing Engineering, University of Zaragoza, 50009 Zaragoza, Spain (e-mail: jaguilar@unizar.es).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3234030.

Digital Object Identifier 10.1109/TII.2023.3234030

## I. Introduction

AUTOMATED visual inspection plays an important role in industrial-informatics-based decision-making systems in various industries, including steel manufacturing companies, automotive industries, electronic manufacturing, and pharmaceutical companies. The correct, consistent, and early detection of surface defects can make it possible to detect defective products early in the manufacturing process, which leads to time and cost savings. Inspection procedures for detecting such defects are usually performed using nondestructive testing (NDT) methods. NDT procedure is a combination of various inspection steps used to identify discontinuities or defects in a product without causing damage to its usability. The most frequently used industrial NDT methods are visual optic testing, radiography, X-ray vision, ultrasonic imaging, dye penetrant testing, magnetic particle testing, and infrared thermal imaging. The testing procedure for each of these methods involves several steps, all of which can be easily automated. However, the final step of visual inspection is more complex in terms of automation and remains primarily a manual process performed by operators.

The traditional machine-vision system relies on hand-crafted features, such as color, contrast, texture, edges, foreground–background statistics, etc., followed by machine learning classifiers, such as support vector machines, decision tree, or $K$-nearest neighbors. Consequently, hand-crafted feature extraction plays an important role in classical approaches. However, these features are not robust and suited for different tasks, which leads to long development cycles. Deep learning methods, on the other hand, learn the relevant features directly from the raw data, without the need for handcrafted feature representations. In recent years, convolutional neural network (CNN) has achieved and even surpassed human-level performance on computer vision tasks such as image classification. The key difference between CNN and traditional machine-vision algorithms is that CNN automatically detects significant features without any human supervision, which made it the most widely used. A fascinating feature of CNN is its ability to take advantage of the spatial or temporal correlation of image data. There are three main problem categories for image recognition tasks using CNN: 1) classification, 2) segmentation, and 3) object detection. The classification task aims to classify an image into a certain category. Starting with the ImageNet Large Scale Visual Recognition Challenge winning architecture of AlexNet [1], a series of increasingly complex architectures including ResNet [2],

Inception [3], Densenet [4], and EfficientNet [5] have been proposed in the literature for the classification task. Object detection is a task that localizes an object using a bounding box. Some of the notable object detection algorithms include Fast R-CNN [6], Faster R-CNN, Mask R-CNN [7], single shot detection (SSD) [8], You Only Look Once (YOLO) [9], etc. Segmentation is the task of performing pixel-by-pixel classification. Several segmentation algorithms have been proposed in the literature including fully convolutional networks, encoder–decoder-based approaches [10], multiscale and pyramid architectures [11], etc.

However, industrial visual inspection systems barely utilized the potential of those complex architectures due to several reasons [12]. One of the main reasons is that the continuous improvement in industrial processes has resulted in fewer and fewer defective samples or the number of defective samples is very limited [13]. This problem of learning from a limited number of samples is usually referred to as the small sample problem, which can easily lead to poor generalization ability of the trained model [14]. In addition, the target surface defects have different scales, making the deep learning models even more challenging to identify the small-sized defects. On the one hand, the visual appearance of the real-world surfaces defects varies with the type of materials, imaging conditions, and camera position. On the other hand, it is challenging to distinguish tiny defects from the noise or non-defect components within an image (as shown in Fig. 1). Hence, the appearance of false positives in a defect-free image is an inevitable circumstance. Furthermore, real-time applications of complex CNN models are extremely limited due to the long inference time and the resulting higher computational resource and power consumption.

To address these limitations, we present a novel universal architecture that integrates classification, segmentation, and detection of surface defects in a single network. Our architecture, Defect-Aux-Net, is primarily motivated by a multitask learning (MTL) scheme that exploits useful information from related learning tasks to help mitigate the problem of data scarcity. The proposed architecture is based on FPN-semantic-segmentation [11] with the additional tasks of defect classification and detection to improve the generalization ability by utilizing the image-level information as an inductive bias. Specifically, we developed a new MTL network based on FPN, where the classification task is carried out in the bottom-up pathway of the network and segmentation is performed in the top-down pathway of the network. To create a bounding box, we employ two subnetworks in the top-down pathway, where one subnet determines the class associated with the bounding box and the other performs the regression to adjust the bounding box position.

The FPN-based feature extractor in the proposed network allows surface defects to be recognized at vastly different scales by efficiently sharing features between image regions. We further introduce the positional and the channel attention mechanisms that focus on learning the features of small surface defects to improve the robustness of detecting small defects surrounded by a complex background.

We evaluate our model on TekErreka, and Severstal [15] surface defect datasets, with defect classification, segmentation, and detection tasks. Experimental results demonstrate that jointly learning features of related tasks can improve the performance of all tasks.

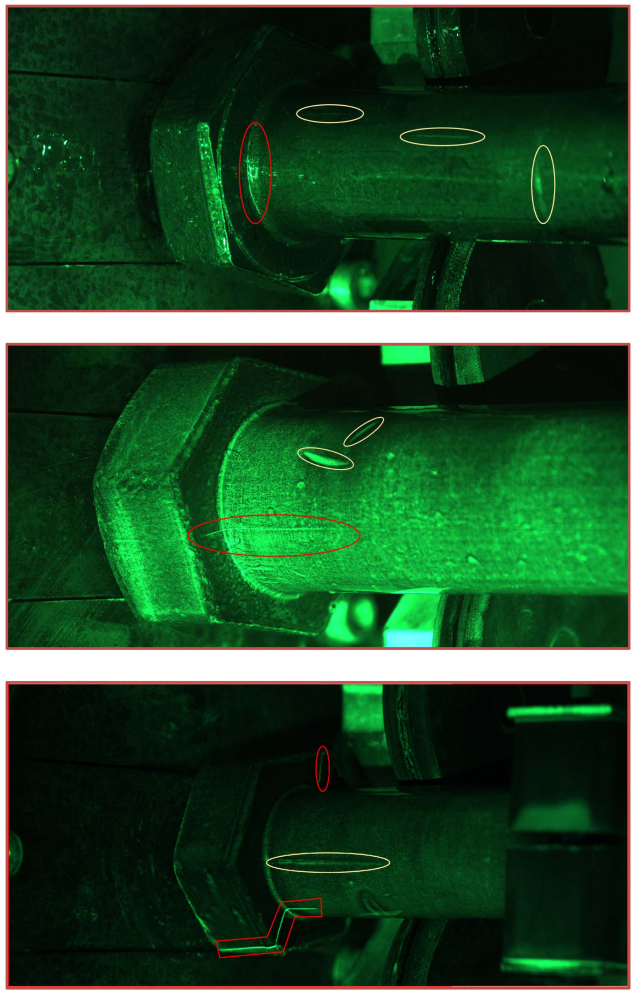

Fig. 1. Magnetic particle inspection on threaded fasteners of different surface finish (TekErreka dataset). Surface defects are marked by red circles and noise due to magnetic particle depositions are marked in yellow.

Overall, the contributions of our work are as follows.

1) First, we propose a Defect-Aux-Net model architecture, which can perform classification, segmentation, and detection of surface defects in a single network. Compared with the existing state-of-the-art CNN models, this architecture is lightweight and compact in terms of model parameters. From the model training point of view employing fewer parameters in the architecture enables the model to efficiently learn potential surface defects from a smaller number of labeled examples.
2) In contrast to existing single-task learning, our proposed MTL in surface defect detection facilitates the model to learn useful representations of the data by exploiting shared information from related tasks.
3) Considering surface defect detection with a complex background, the positional and the channel attention mechanisms are incorporated to amplify target features and to reduce the influence of background noise.

4) The proposed model is compact and efficient with the state-of-the-art performance that meets the computational resource requirements of the real-time inference speed.

## II. Related Work

A large and growing body of literature has explored the use of CNN for surface defect identification. Kim et al. [16] adopted a few-shot learning technique with a Siamese neural network using CNN, which aims to classify surface defects with a limited number of training images. Lin et al. [17] employed a class activation mapping technique in CNN to simultaneously achieve defect classification and localization tasks in the LED chip defect inspection process. Tao et al. [18] designed cascaded autoencoder (CASAE) architecture to segment and localize defect region. The proposed architecture transforms the input image into a mask prediction, and then, the defect region of the segmented mask is classified into their specific classes. Jing et al. [19] combined autoencoder with a fully connected network to detect keyboard light leakage defects from mere dust. Jian et al. [20] leveraged generative adversarial network to exaggerate the tiny defects within the images to improve the accuracy of different classifiers. Zheng et al. [21] proposed a three-stage model for rail surface and fastener defect detection. In the first stage, the YOLOV5 framework is employed to localize the rail and fasteners. Then, an object detection model based on Mask-RCNN is used to detect the surface defect of the rail surface. At the final stage, the ResNet architecture is utilized to classify defects of the fasteners. To detect defects at a different scale, Xu et al. [22] used a pretrained ResNet model to extract the multiscale features and fuse them using a multilevel feature fusion network. In [23], U-Net and residual U-Net architectures were used for the fine-grained segmentation of surface defects on a steel sheet. The main drawback of these methods is that the model needs a large amount of annotated data and hence the localization of defects is very coarse in the real-time scenario.

## III. Proposed Method

### A. Network Architecture

Our proposed network is inspired by two deep learning architectures that are widely used: 1) feature pyramid network (FPN) and 2) ResNet-50. Recognizing surface defects at vastly different scales is a fundamental challenge in the industrial machine vision system. For this reason, we use FPN that uses a pyramidal hierarchy of convolutional filters to extract feature pyramids at different scales. FPN consists of two pathways: 1) bottom-up and 2) top-down. The bottom-up pathway also known as the encoder is the typical CNN, which can be any image classifier for feature extraction. As we go up, the encoder gradually decreases the spatial resolution while building high-level feature maps. The top-down pathway is connected to the bottom-up pathway through lateral connections for efficient multiscale feature fusion. It is designed to enhance the feature maps from the bottom-up pathway and build semantically strong feature maps at multiple scales by double upscaling. As a result, the feature pyramid has rich semantics at all levels because the lower semantic features are interconnected to the higher semantics.

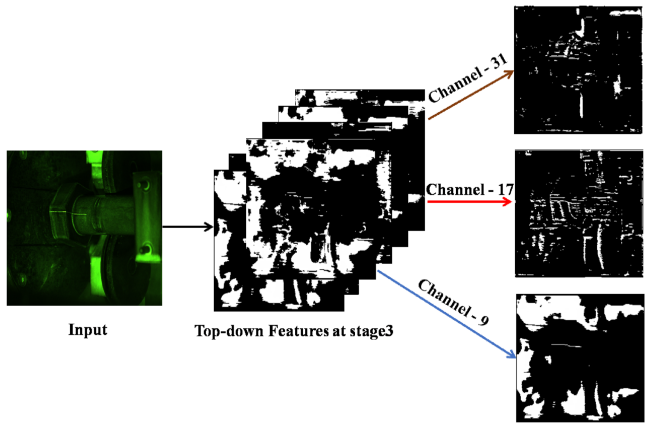


Fig. 2. Sample features in different channels of top-down pathway at stage 3.

*1) Bottom-Up Pathway:* We tested several standard image classification architectures to select the core model and finally chose ResNet-50 as the backbone. ResNet-50 has shown great performance for surface defect classification, segmentation, and detection tasks. ResNet-50 architecture has the advantage of using a stride of two for each scale reduction, which makes it easier to incorporate ResNet-50 into FPNs when we need to upscale feature maps in a top-down pathway. Furthermore, Resnet-50 is a relatively small network based on modern standards; therefore, it is suitable for our limited labeled data problem. However, existing ResNet-50 feature pyramids have two problems in the way they apply convolution operations to the input features. First, the receptive field of the encoder has the information only about the local region, so the global information is lost. Second, the feature maps constructed from the learned weights are given an equal magnitude of importance but some feature maps are more important for the next layers than others. For instance, a feature map that contains edge information of the defects might be more important than another feature map that has background texture information (as shown in Fig. 2). Thus, to incorporate channel attention we adopt Squeeze-and-excitation (SE) module [24] in the encoder. SE module consists of three components: 1) squeeze, 2) excite, and 3) scale components.

The main goal of the squeeze component is to extract global information from each of the channels c in a feature block U. The global information is acquired by applying a global average pooling operation across their spatial dimensions $(H \times W)$ for each channel $\mathrm{U_c}$ of U to obtain global statistics $(1 \times 1 \times C)$. Mathematically, squeeze operation can be represented as

$$z_c = F_{\text{squeeze}}(U_c) = \frac{1}{H \times W} \sum_{m=1}^{H} \sum_{n=1}^{W} U_c(m, n). \quad (1)$$

After obtaining global information from the squeeze component, the excite component generate a set of weights for each channel. It uses a fully connected multilayer perceptron (MLP) bottleneck structure to dynamically calibrate the weights. This

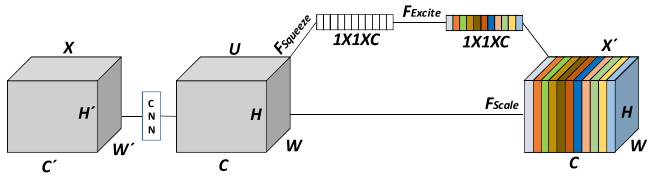


Fig. 3. Structure of squeeze and excite module.

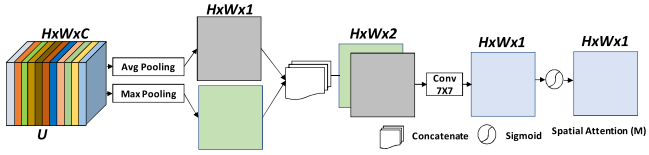


Fig. 4. Structure of spatial attention module.

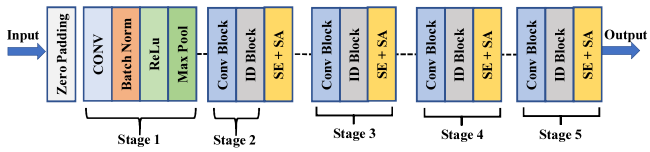


Fig. 5. FPN bottom-up structure with attention module.

MLP bottleneck has two fully connected layers with sigmoid activation as the output layer. The output of the excitation component can formally be represented by the following equation:

$$s = F_{\text{excite}}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \rho(W_1, z)) \quad (2)$$

where $\sigma$ is a Sigmoid operation, $\rho$ is ReLU operation, $z$ is the output from the squeeze component, $W_1$ and $W_2$ refers to weights of the two fully connected layers. Subsequently, each channel in the feature map is scaled by a simple elementwise multiplication of the input feature map and weights obtained from the excite component (as shown in Fig. 3).

Surface defects only appear in some parts of the image but not the whole image. Unlike the conventional Resnet-50 architecture, which gives equal importance to each region in an image, the spatial attention reduces background interferences by assigning a weight to each pixel in the feature map.

The spatial attention focuses on the most relevant parts of the feature maps in the spatial dimension. The working principle of our spatial attention mechanism is as follows.

Given feature block $U$, we use average and max-pooling operations along the channel axis and concatenate them to generate an efficient feature map summary M. A convolutional layer followed by sigmoid operation is then performed on the feature M to produce a spatial attention map (as shown in Fig. 4).

ResNet uses four modules consisting of residual blocks, each of which uses two blocks, 1) Identity (ID) blocks and 2) convolution blocks, depending on whether the input / output dimensions are the same or different. We arrange SE and SA modules in series and integrate into a residual block (as shown in Fig. 5).

*2) Top-Down Pathway:* Deep features from a bottom-up pathway are upsampled by convolutions and bilinear upsampling operations until all the feature maps reach one-fourth scale. Attention module outputs from a bottom-up pathway $\{C_2, C_3, C_4, C_5\}$ are fused to a top-down pathway through lateral connections for an efficient multiscale feature fusion. First, $1 \times 1$ convolutional filter is applied to the feature maps $\{C_2, C_3, C_4, C_5\}$ to get a fixed number of channels and then merged with the corresponding top-down feature map by elementwise addition. Finally, the outputs are summed and then transformed into a pixelwise output (as shown in Fig. 6).

*3) Segmentation Branch:* The segmentation branch from a top-down pathway aims at classifying pixels into a set of predefined classes. The pixels corresponding to background are far more numerous than the pixels of surface defects in the real-world dataset, which causes the model to be biased toward the background element. To address the pixelwise class imbalance, we employ Dice loss, which uses the Dice coefficient to calculate overlapping of the pixels of the predicted mask with the ground truth label. Mathematically, the Dice loss function is defined as

$$L_{\text{seg}} = 1 - \frac{2y\hat{y} + 1}{y + \hat{y} + 1} \quad (3)$$

where $y_i$ is the ground truth label and $\hat{y}_i$ is the predicted label. The value of the Dice coefficient ranges from 0 to 1, where 1 indicates the perfect and complete overlap of pixels.

*4) Classification Branch:* The output of the bottom-up pathway encodes the rich abstract feature representations of the input image. Hence, we utilize the spatial average of the feature maps from the bottom-up pathway via a global average pooling layer, and then, the resulting feature vector is fed into the sigmoid or softmax layer depending on the classification type. We employ binary cross-entropy (BCE) as a classification loss function. Mathematically, our classification loss is defined as

$$L_{\text{class}} = \frac{1}{k} \sum_{1}^{k} CE(y_i, \hat{y}_i) \quad (4)$$

where $y_i$ is the ground truth label, $\hat{y}_i$ is the predicted label of $i$th sample, and $k$ is the total number of samples. CE is the binary cross entropy function.

*5) Object Detection Branch:* We extract bounding boxes and its associated classes by employing box regression and classification subnets at each level of top-down pathway. The classification subnet predicts the probability of defect presence at each spatial location of an input image. The box regression subnet is attached to a top-down pathway in parallel to the classification subnet for the purpose of regressing offset from each anchor box to the ground truth bounding boxes. To handle class imbalance problems, we adopt focal loss [25], an improved version of cross entropy to focus learning on hard negative examples. It is defined as

$$L_{\text{detection}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \quad (5)$$

where $\alpha_t$ is the weight parameter per class and $\gamma$ is the hyperparameter focuses on hard negative samples. We choose $\alpha_t = 0.25$ and $\gamma = 4$ as suggested in [26].

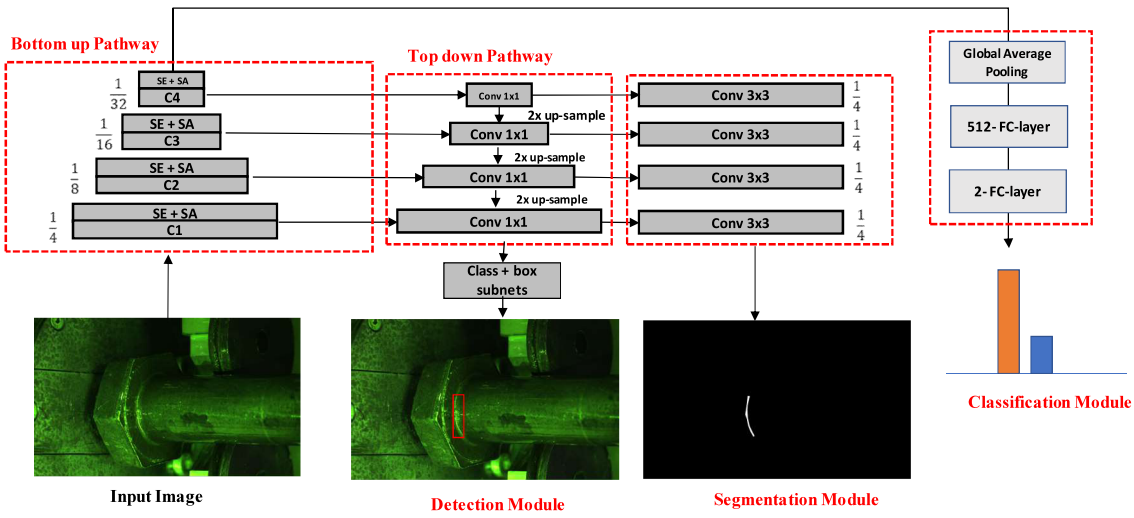


Fig. 6. Overview of the proposed Defect-Aux-Net architecture. It is mainly composed of classification, segmentation, and detection module that incorporates multitask loss function.

### B. Loss Function

Our proposed method combines three loss functions from the classification, segmentation, and detection tasks, which provide mutual sources of inductive bias for each task. Specifically, the segmentation and detection loss functions signal back to the entire model (bottom-up and top-down pathway) while the classification loss signals back only to bottom-up pathway. We combine and weight the three losses into a multitask loss $L_M$ to leverage the heterogeneous annotations and jointly optimize multiple tasks as follows:

$$L_M = \beta L_{\text{class}} + \beta_1 L_{\text{seg}} + \beta_2 L_{\text{detection}} \tag{6}$$

where $\beta$, $\beta_1$, and $\beta_2$ are weight parameters. We tested with different combinations of weight parameters and found that $\beta = \beta_1 = \beta_2 = 1$ yields the best result for all the tasks.

## IV. Experiments

### A. Datasets

In this article, we evaluate our framework on real-world surface defect identification problems. We use two challenging datasets with increasing resolutions and complexities, 1) Severstal steel sheet [15] and 2) TekErreka steel fastener defect datasets. Severstal, the largest steel and steel-related mining company, has recently published the largest industrial steel sheet surface defect dataset, which contains pixelwise masks annotated by their technical experts. The dataset contains 12 568 grayscale images of size 1600×256. Each image in the dataset has the possibility of having either no defects, a single defect, or multiple defects divided into four classes. Fig. 7 shows the example of steel defect images on Severstal datasets. We randomly select 10% and 20% of the 12 568 original images as the validation and test data. The main challenge with this dataset is that the interclass similarities between defective and defect-free examples are very high.

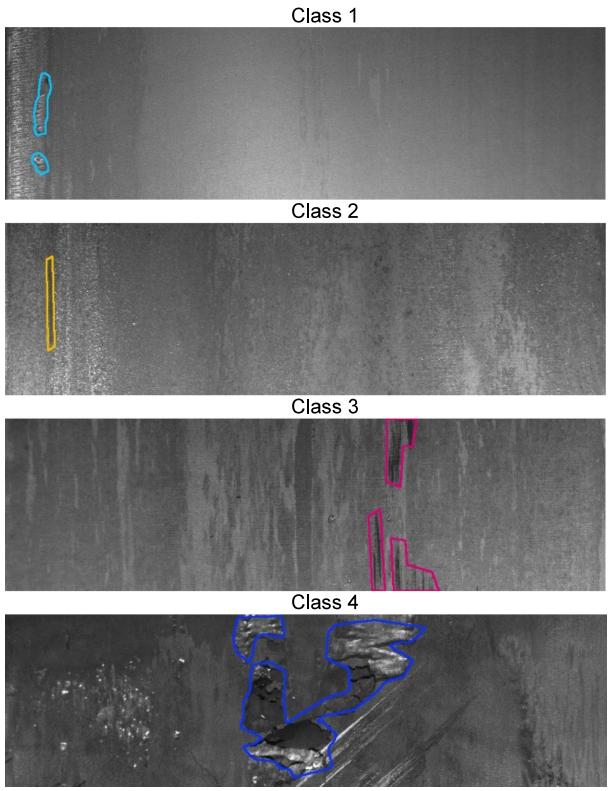


Fig. 7. Sample images of Severstal steel with four classes of defect.

The TekErreka dataset is a self-collected steel fastener surface defect dataset based on a magnetic particle inspection procedure. The magnetic particle inspection is an excellent method to investigate near-surface defects in steel fasteners. The basic principle is to magnetize a steel fastener parallel to its surface. If the fastener is free from defects the magnetic field lines run within the fastener and parallel to its surface. In case of magnetic inhomogeneity, for instance, near cracks, the magnetic field lines

will locally leave the surface and a leakage field occurs. When a suspension of ferromagnetic particles is applied to the test piece surface the magnetic particles will run off at defect-free areas. In the places of leakage fields, the magnetic particles are attracted and clustered together thus indicating the location of the defect. The surface defects can be visible under ultraviolet light. We acquired the TekErreka dataset from a magnetic particle inspection apparatus located at the Erreka fastening solutions. The defects in the TekErreka dataset differ in their size, shape, location, and materials type and thus cover several scenarios in real-time defect detection. The difficulty in this dataset lies in the similarity of defects and noise due to magnetic particles deposition on the defect-free surface of the fasteners. There are many factors responsible for the noise component, which include magnetic particle size, the amount of magnetic particles used, ultraviolet light present, etc. The original examples are directly stored in a database as RGB images of size $2464 \times 2056$. It has 450 positive and 1200 negative examples. We split the TekErreka dataset into training and testing sets: 80% for training and 20% for evaluation of the model performance.

### B. Preprocessing

We resized the images of the Severstal dataset to $128 \times 800$ and the TekErreka dataset to $600 \times 600$. To keep the pixel values in the same scale, we normalized the images using min–max standardization. It rescales raw pixel values to a range of 0 and 1. This helps the optimizer not get stuck taking steps that are too large in one dimension, or too small in another.

### C. Data Augmentation

To improve the diversity of the training set, we apply random but realistic data augmentation such as rotation, vertical/horizontal flips, zoom, shear, and channel shifts.

### D. Training Details

The Defect-Aux-Net is implemented using the Tensorflow framework. All the experiments are run on Google-cloud TPU V2 infrastructure, which contains 8 cores with 64 GB memory. The network is optimized with the Adam optimizer and trained with a batch size of 128 for 50 epochs. We adopt one cycle policy [27] to find an optimal learning rate.

### E. Evaluation Metrics

The classification results are evaluated using precision, recall, F1-score, and binary accuracy

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{F1Score} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{9}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{10}$$

TABLE I
PERFORMANCE OF THE PROPOSED APPROACH ON LOSS VARIANTS FOR THE DEFECT SEGMENTATION TASK

| Loss Function | IoU | Dice |
|---|---|---|
| BCE | 0.892 | 0.911 |
| Dice | **0.903** | **0.926** |
| Jaccard | 0.900 | 0.913 |
| Dice + BCE | 0.901 | 0.920 |
| Jaccard + BCE | 0.899 | 0.912 |

The bold values signify our proposed method.

where TP, TN, FP, and FN denote true positive (correctly identified surface defects), true negative (correctly identified non-defect images), false positive (erroneously classified images as surface defect), and false negative (erroneously classified images as non-defect). Precision measures the percentage of images with surface defects that are correctly classified while recall is the ratio of correctly classified images with surface defects to all images with surface defects. F1-score can be interpreted as a harmonic mean of precision and recall. The overall performance of the classification task is measured by its accuracy.

The segmentation results are evaluated using Dice score and Intersection-over-Union (IoU), which quantify the percentage overlap between the predicted and target binary masks. To evaluate defect detection results, we used the mean average precision (mAP) that compares the detected bounding box to the ground truth bounding box and returns a score.

### F. Experiments on Defect Segmentation

We performed a series of experiments on the TekErreka dataset to test the effectiveness of different loss functions. First, we trained Defect-Aux-Net using BCE, and Dice loss alone as the segmentation loss. Then, it was trained using a combination of loss functions. The results are shown in Table I.

Using Dice loss alone yielded more accurate results than using a combination of losses. Additionally, the Dice loss function assisted our model to converge faster. We use the Dice loss function throughout rest of the experiments.

To verify the effectiveness of the segmentation task using the MTL strategy, we compared the proposed MTL network (Defect-Aux-Net) against the following network with the same bottom-up backbone (Resnet50 + SE + SA attention module).

1) FPN [11]: This is the original FPN architecture without the MTL strategy and serves as our baseline.
2) UNet [10]: This network uses an encoder for multilevel feature extraction and a decoder that scales them up and combines multilevel features through stacking.
3) LinkNet [28]: This is similar to UNet with the difference of replacing stacking operation with addition in skip connections.
4) PSPNet [28]: Pyramid scene parsing network uses a pyramid pooling module for multiscale feature extraction.

Based on the experimental results, we observed that the proposed multitask learning strategy achieves better segmentation performance as compared to the state-of-the-art segmentation models. The Dice and IoU scores of the various segmentation models on the Severstal dataset are depicted in Figs. 8 and 9.

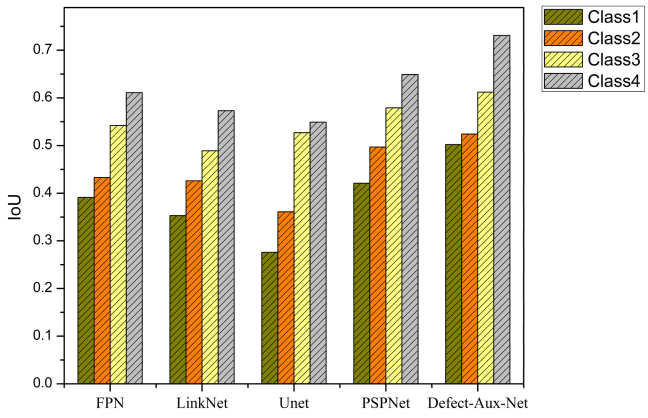


Fig. 8. IOU comparison between the state-of-the-art segmentation methods and the proposed approach on each type of defect classification.

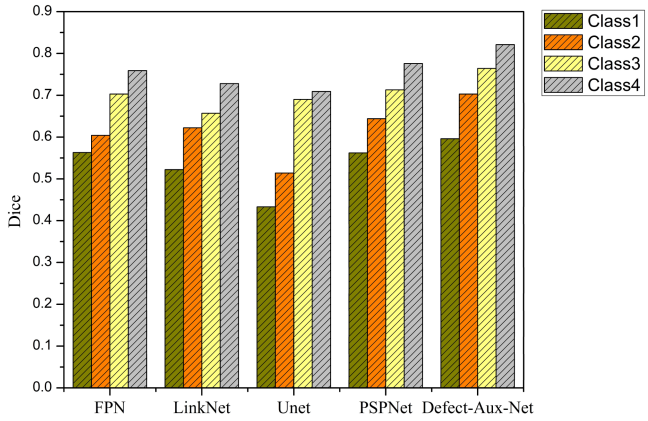


Fig. 9. Dice score comparison between the state-of-the-art segmentation methods and the proposed approach on each type of defect classification.

TABLE II
PERFORMANCE OF THE COMPETING MODELS ON THE TEKERREKA DATASET

| Model | IoU | Dice |
|---|---|---|
| FPN [11] | 0.881 | 0.902 |
| LinkNet [28] | 0.876 | 0.895 |
| Unet [10] | 0.832 | 0.856 |
| PSPNet [29] | 0.885 | 0.917 |
| Defect-Aux-Net | **0.903** | **0.926** |

The bold values signify our proposed method.

We observe that Defect-Aux-Net is able to achieve higher scores for all classes as compared to the other segmentation models. Table II shows the performance of the various networks on the TekErreka dataset. Experimental results from Table II showed that the proposed multitask learning can improve the performance of its corresponding single-task model. Taking advantage of the classification-guidance module, Defect-Aux-Net avoids the oversegmentation of defects in a complex background.

TABLE III
COMPARISON OF PERFORMANCE OF DEFECT-AUX-NET AND STATE-OF-THE-ART CLASSIFICATION MODELS

| Model | Dataset | Class | Recall | Precision | F1-Score | Accuracy |
|---|---|---|---|---|---|---|
| Resnet-50 [2] | Severstal | Class1 | 0.454 | 0.403 | 0.427 | 0.831 |
| | | Class2 | 0.591 | 0.533 | 0.561 | 0.958 |
| | | Class3 | 0.918 | 0.847 | 0.881 | 0.811 |
| | | Class4 | 0.857 | 0.852 | 0.854 | 0.963 |
| | TekErreka | Class1 | 0.759 | 0.979 | 0.855 | 0.949 |
| SEResnet-50 [24] | Severstal | Class1 | 0.508 | 0.556 | 0.531 | 0.875 |
| | | Class2 | 0.617 | 0.580 | 0.598 | 0.970 |
| | | Class3 | 0.980 | 0.816 | 0.891 | 0.817 |
| | | Class4 | 0.559 | 0.940 | 0.701 | 0.940 |
| | TekErreka | Class1 | 0.803 | 0.968 | 0.878 | 0.955 |
| Effecient net-B0 [5] | Severstal | Class1 | 0.891 | 0.859 | 0.875 | 0.964 |
| | | Class2 | 0.872 | 0.732 | 0.796 | 0.984 |
| | | Class3 | 0.943 | 0.963 | 0.953 | 0.929 |
| | | Class4 | 0.946 | 0.924 | 0.935 | 0.983 |
| | TekErreka | Class1 | 0.858 | 0.928 | 0.892 | 0.958 |
| Defect-Aux-Net (ours) | Severstal | Class1 | 0.891 | 0.926 | 0.908 | 0.975 |
| | | Class2 | 0.957 | 0.900 | 0.928 | 0.994 |
| | | Class3 | 0.982 | 0.929 | 0.955 | 0.929 |
| | | Class4 | 0.946 | 0.940 | 0.943 | 0.985 |
| | TekErreka | Class1 | 0.887 | 0.939 | 0.912 | 0.971 |

## G. Experiments on Defect Classification

We evaluated and compared the classification task performance of the proposed approach with the state-of-the-art deep learning architectures. While evaluating the classification task, the other two modules, segmentation and detection, are removed from the network. The results of the experiments are summarized in Table III. It can be noted that most errors are due to false positives. The visual similarity between defects and surface noise leads to false positive errors. Notably, Defect-Aux-Net obtains overall accuracy of at least 92.9% and at most 99.4% across all defect types on the Severstal dataset. Based on the experimental results, we observe that the proposed MTL approach achieves a surpassing performance over the other models. Also, it is evident that incorporating the segmentation task improves the performance of the classification task and vice-versa.

To assess the effectiveness of the proposed approach against the limited data problem, we removed part of the training data and conducted a series of experiments leaving 90%, 75%, and 50% from the training data. The effect of training data size on its accuracy is shown in Fig. 10. The proposed Defect-Aux-Net showed a consistent performance even when only 50% of the original training data is used in training. As seen, the proposed multitask loss function greatly improves the performance of the classification task by taking image, pixel, and map level optimization into consideration.

To verify the importance of the attention mechanisms in Defect-Aux-Net, we compared the accuracy of the network with

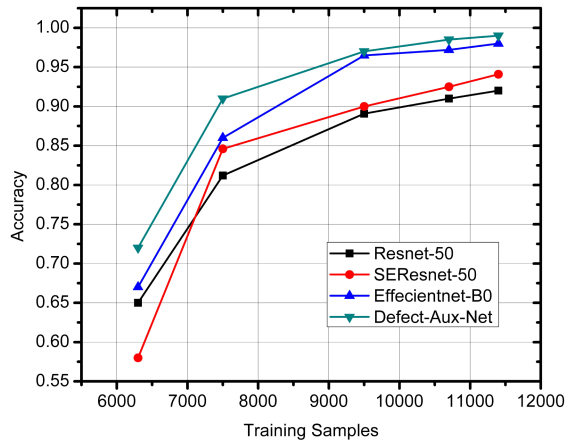


Fig. 10. Training data size versus classification accuracy of the Severstal dataset.

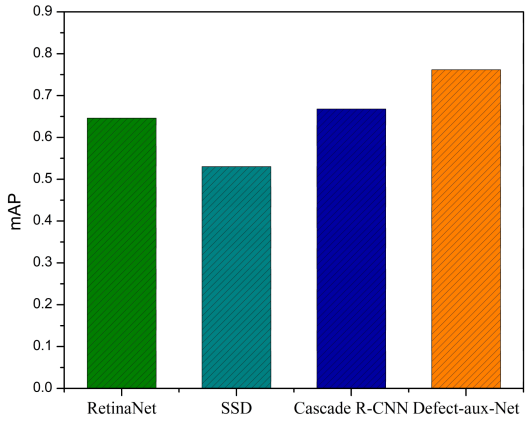


Fig. 11. mAP comparison between the state-of-the-art detection models and the proposed model.

TABLE IV
EFFECT OF USING ATTENTION MECHANISMS ON TEKERREKA DATASET

| Model | Accuracy | Parameters (M) |
|---|---|---|
| Defect-Aux-Net (without attention) | 0.962 | 33.2 |
| Defect-Aux-Net (with SE attention) | 0.968 | 35.7 |
| Defect-Aux-Net (Spatial attention) | 0.963 | 33.5 |
| **Defect-Aux-Net (with SE + Spatial attention)** | **0.971** | **36.2** |

The bold values signify our proposed method.

and without spatial and channel attention mechanism (squeeze and excite) on the TekErreka dataset, as shown in Table IV. Furthermore, we experimented with inserting a combination of both spatial and channel attention mechanisms.

### H. Experiments on Defect Detection

The proposed model is compared with other object detection algorithms on the TekErreka dataset. The comparative models include SSD [8], RetinaNet [25], and cascade R-CNN [30]. Fig. 11 shows the mAP scores of the various detection models for the TekErreka dataset. We observe that Defect-Aux-Net is able to achieve a higher mAP score as compared to the alternative networks. The mAP of the proposed algorithm is 17.95%, 43.77%, and 26.03% higher than that of RetinaNet, SSD, and Cascade RCNN.

### I. Inference Time

In addition to the model performance, we attempt to determine the effectiveness of the MTL framework on the inference time. We compared the inference time of the proposed approach with a conventional single-task network where each task requires a separate pass through the network during inference. All the inference time was measured using a computer with an Intel Core processor. The CPU specification is summarized in Table V.

TABLE V
SYSTEM SPECIFICATION

| CPU Specification | |
|---|---|
| CPU Processor type | Intel(R) Xeon(R) |
| Processor Base Frequency | 2.20 GHz |
| Total Cores | 1 |

TABLE VI
COMPARISON OF THE INFERENCE TIME OF DEFECT-AUX-NET AND BASELINE MODEL

| Model | Task | Task Name | Inference time CPU (s) | Parameters (M) |
|---|---|---|---|---|
| Single Task Networks | Task 1 | Classification (ResNet-50) | 0.0654 | 23.5 |
| | Task 2 | Segmentation (ResNet-50 FPN) | 0.1106 | 26.9 |
| | Task 3 | Detection (ResNet-50 RetinaNet) | 0.1780 | 34.0 |
| | Total | Classification + Segmentation + Detection | 0.3540 | 84.4 |
| Multitask Network | Multitask | Classification + Segmentation + Detection (Defect- Aux-Net) | **0.1927** | **36.2** |

The bold values signify our proposed method.

From Table VI, we can see that our proposed framework allows for a 57.1% reduction in the model size by solving different tasks jointly rather than independently. Compared to the single-task network, the inference time of our proposed network reduces by 45.5%.

## V. DISCUSSION

By incorporating the MTL strategy, our proposed Defect-Aux-Net improves the performance of defect classification, segmentation, and detection tasks. Intuitively, the multitask deep learning system can provide regularization effects to the

multiscale feature learning and thus improve the performance as opposed to the single-task algorithms. Also, the MTL framework can save computational inference time as only a single network needs to be evaluated for three different tasks. The experimental results show that our proposed algorithm greatly improves the performance of the surface defect identification tasks compared to other state-of-the-art deep learning algorithms.

## VI. Conclusion

In this article, we described an attention-guided MTL scheme, which combines classification, segmentation, and defection for automated surface defect detection. Specifically, we proposed an extended FPN architecture with Resnet-50 incorporated as the encoder section of the model. The hybrid loss function is introduced to enhance the performance of the model. An overall accuracy of 97.1%, Dice score of 0.926, and mAP of 0.762 on classification, segmentation, and detection tasks of the TekErreka dataset were achieved with Defect-Aux-Net.

## Acknowledgment

This work was undertaken in the context of DIGIMAN4.0 project ("Digital Manufacturing Technologies for Zero-Defect," https://www.digiman4-0.mek.dtu.dk/). DIGIMAN4.0 is a European Training Network supported by Horizon 2020, the EU Framework Programme for Research and Innovation under Project 814225.

## References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[5] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 6105–6114.

[6] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[7] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020, doi: 10.1109/TPAMI.2018.2844175.

[8] W. Liu et al., "SSD: Single shot multiBox detector," in *Proc. Eur. Conf. Comput. Vis.* (Lecture Notes in Computer Science Series), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.

[11] S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets, "Feature pyramid network for multi-class land segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 272–2723, doi: 10.1109/CVPRW.2018.00051.

[12] X. Ni, Z. Ma, J. Liu, B. Shi, and H. Liu, "Attention network for rail surface defect detection via consistency of intersection-over-union (IoU)-guided center-point estimation," *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 1694–1705, Mar. 2022, doi: 10.1109/TII.2021.3085848.

[13] D. Zhang, K. Song, Q. Wang, Y. He, X. Wen, and Y. Yan, "Two deep learning networks for rail surface defect inspection of limited samples with line-level label," *IEEE Trans. Ind. Inform.*, vol. 17, no. 10, pp. 6731–6741, Oct. 2021, doi: 10.1109/TII.2020.3045196.

[14] L. Wen, Y. Wang, and X. Li, "A new cycle-consistent adversarial networks with attention mechanism for surface defect classification with small samples," *IEEE Trans. Ind. Inform.*, vol. 18, no. 12, pp. 8988–8998, Dec. 2022, doi: 10.1109/TII.2022.3168432.

[15] Kaggle, "Severstal: Steel defect detection. Can you detect and classify defects in steel?," 2019.

[16] M. S. Kim, T. Park, and P. Park, "Classification of steel surface defect using convolutional neural network with few images," in *Proc. IEEE 12th Asian Control Conf.*, 2019, pp. 1398–1401.

[17] H. Lin, B. Li, X. Wang, Y. Shu, and S. Niu, "Automated defect inspection of LED chip using deep convolutional neural network," *J. Intell. Manuf.*, vol. 30, no. 6, pp. 2525–2534, Aug. 2019, doi: 10.1007/s10845-018-1415-x.

[18] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Appl. Sci.*, vol. 8, no. 9, 2018, Art. no. 1575, doi: 10.3390/app8091575.

[19] J. Ren and X. Huang, "Defect detection using combined deep autoencoder and classifier for small sample size," in *Proc. IEEE 6th Int. Conf. Control Sci. Syst. Eng.*, 2020, pp. 32–35, doi: 10.1109/ICCSSE50399.2020.9171953.

[20] J. Lian et al., "Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network," *IEEE Trans. Ind. Inform.*, vol. 16, no. 2, pp. 1343–1351, Feb. 2020, doi: 10.1109/TII.2019.2945403.

[21] D. Zheng et al., "A defect detection method for rail surface and fasteners based on deep convolutional neural network," *Comput. Intell. Neurosci.*, vol. 2021, Jul. 2021, Art. no. 2565500, doi: 10.1155/2021/2565500.

[22] P. Xu, Z. Guo, L. Liang, and X. Xu, "MSF-Net: Multi-scale feature learning network for classification of surface defects of multifarious sizes," *Sensors*, vol. 21, no. 15, Jul. 2021, Art. no. 5125, doi: 10.3390/s21155125.

[23] D. Amin and S. Akhter, "Deep learning-based defect detection system in steel sheet surfaces," in *Proc. IEEE Region 10 Symp.*, 2020, pp. 444–448, doi: 10.1109/TENSYMP50017.2020.9230863.

[24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[27] L. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*.

[28] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process.*, 2017, pp. 1–4, doi: 10.1109/VCIP.2017.8305148.

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.

[30] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.