APPLIED RESEARCH

Deep Learning Based Classification of Military Cartridge Cases and Defect Segmentation

SERHAT TURAL [1], **REFIK SAMET** [1], **(Member, IEEE),**
SEMRA AYDIN[2], **AND MOHAMED TRAORE**[1]
[1]Department of Computer Engineering, Ankara University, 06100 Ankara, Turkey
[2]Department of Computer Engineering, National Defense University, 06420 Ankara, Turkey

Corresponding author: Serhat Tural (serhattural@hotmail.com)

ABSTRACT The final stage of the production process in the industry is quality control. Quality control answers the question of is there a defect on the surface of the products. Frequently the quality control is performed manually. The disadvantages of manual quality control are high error rate (low accuracy), low product rate (low performance) and high expense rate (high cost). The solution is automatic quality control using machine vision systems. These systems classify the products and segment the defects on their surfaces by processing the images taken by cameras during the production process in real-time. Some products like military cartridge cases have metallic, cylindrical, non-uniform texture and highly reflective surface. So, the quality of images is very important. Another factor that affects the accuracy is the non-uniform texture of the product surface. Distinguishing the product non-uniform texture from defect texture is a challenging problem. In previous works, this problem has been tried to be solved with image processing and deep learning techniques and the accuracy of 97% and 96% have been obtained, appropriately. According to NATO standards, the accuracy of the classification of the military cartridge cases should be above 99%. In this work, the methodology for classification of the military cartridge cases and segmentation of the defects on their surfaces with non-uniform texture is proposed to increase the accuracy. In scope of the proposed methodology the datasets with non-defective, defective, and labeled/masked image classes of the cartridge cases were created, the deep learning models to classify the military cartridge cases and segment the defects on their surfaces were proposed, implemented, and obtained results were evaluated using the metrics such as Accuracy, Precision, Recall, F1-Score, Jaccard Index (JI) and Mean Intersection over Union (mIoU). Obtained results showed that the proposed methodology increased the accuracy of classification to 100% with the DenseNet169 model and the F1-Score of segmentation to 92.1% with Improved U-Net and ResUnet models.

INDEX TERMS Deep learning, defect detection, defect segmentation, defect texture, object inspection, military cartridge case, object classification, non-uniform surface texture, quality control.

I. INTRODUCTION

Machine vision systems are very popular in industrial applications. Due to these systems the products' standards are guaranteed, accuracy and performance are increased, and the expense is decreased.

One area of machine vision system usage is the quality control of industrial products. The final stage of the production process in industry applications is quality control.

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Cusano [ID].

Quality control answers the question of is there a defect on the surface of the product. Frequently, the quality control is performed manually. The disadvantages of manual quality control are high error rate (low accuracy), low product rate (low performance) and high expense rate (high cost). The solution is automatic quality control using machine vision systems.

Machine vision systems classify the products and segment the defects on their surfaces by processing the images taken by cameras during the production process in real-time. So, the quality of images is very important. Another factor that

affects the accuracy is the property of the product surface. Some products like military cartridge cases have metallic, cylindrical, non-uniform texture and highly reflective surface. Metal surface has a high coefficient of reflection and non-uniform texture. This kind of texture includes random patterns similar to the real defects. Such features make difficult to distinguish real defects on the surface of products from the non-uniform product texture.

Many traditional algorithms have been developed to use in machine vision systems to detect surface quality. Traditional machine vision algorithms couldn't reach desired results for objects with highly reflective surfaces, random structures, and textures, making it hard to identify defects to use in a real environment. Some surface defects are very difficult to distinguish from product surface texture by using traditional image processing algorithms. Human sense can detect out these types of defects. Deep learning comes to behave like human sense and can detect such kinds of defects. However, the accuracy obtained in the existing studies related to the deep learning techniques is not sufficient for inspection of critical tasks like military cartridge cases. The image processing techniques have a better performance on structural defects like it is easy to distinguish from the product texture. On the other hand, the deep learning techniques imitate the human sense and have better performance by detecting very soft defects that are similar to the product texture. So, it can be concluded that by deep learning techniques the higher accuracy can be reached.

Distinguishing the product texture from defect texture is a challenging problem. In previous works [1], [2], this problem has been tried to be solved with image processing [1] and deep learning [2] techniques and the accuracy of 97% and 96% were obtained, appropriately. Proposed deep learning technique in [2] was used for classification of the defective cartridge cases. According to NATO standards, the accuracy of the classification of the military cartridge cases should be above 99%. The segmentation of defects on the surface of cartridge cases is also very important. However, there are no studies in the current literature that segment the defects of cartridge cases.

In this work, the methodology for classification of the military cartridge cases and segmentation of the defects on their surfaces is proposed to increase the accuracy. In scope of the proposed methodology the datasets with non-defective, defective, and labeled/masked image classes of the cartridge cases were created, the deep learning models to classify the military cartridge cases and segment the defects on their surfaces were proposed, implemented, and obtained results were evaluated using the metrics such as Accuracy, Precision, Recall, F1-Score, Jaccard Index and Mean Intersection over Union.

The rest part can be summarized as follows. In Chapter II the related works are evaluated. Chapter III presents the deep convolutional neural networks. Chapter IV describes the proposed methodology. Chapter V implements the proposed methodology on military cartridge cases. Finally,

Chapter VI concludes the obtained results and future works.

II. RELATED WORKS
This section discusses existing works related to defect detection, classification, and segmentation on surfaces with non-uniform textures using image processing and deep learning techniques.

A. IMAGE PROCESSING TECHNIQUES
In this subsection, the existing works related to defect detection, classification, and segmentation on surfaces with non-uniform textures using image processing techniques were summarized.

Tural *et al.* [1] propose the image processing procedure that consists of four main stages: preprocessing, segmentation, defect detection, classification. At the preprocessing stage the region of interest (ROI) is extracted, and the Gaussian blur is applied to remove some noise in the image. At the segmentation stage the cartridge case is extracted from background by performing a) Adaptive thresholding, b) Morphological closing, c) Obtaining the cartridge case mask, d) Obtaining the minimum area rectangle, and e) Rotation. At the defect detection stage, the defects are detected by performing a) Bilateral filtering, b) Sobel filtering, c) Thresholding, and d) Morphological closing. Finally, at the classification stage contour-based features such as the hu-moments, contour area, contour perimeter, contour approximation, and convex hull of the defects are extracted, and the support vector machine (SVM) is used for classification. The obtained results show that the proposed image processing procedure guaranteed 96% accuracy. Thus, the proposed procedure is effective for detecting defects on the metallic, cylindrical, non-uniform textured and highly reflective surfaces.

Yun *et al.* [3] propose a defect detection algorithm for steel wire rods produced by the hot rolling process. The authors detect the surface defects by performing the dynamic programming and a discrete wavelet transform. The proposed algorithm has a detection accuracy of 84.7%. Thus, the proposed algorithm is effective for detecting defects in scale-covered surfaces of steel wire rods.

Chondronasios *et al.* [4] investigate the defect detection possibilities in extruded aluminum profiles. The authors used two features and classified defects in three categories: non-defective, blister and scratch. An accuracy of 98.6% was obtained by combining the current literature on the field with a new approach to select and manipulate the variables. The values were obtained from the statistical features of co-occurrence matrices on the gradient magnitude of the image as a result of the Sobel operator.

Liu & Yu [5] present an automated surface defect inspection system for the optical infrared cut-off (IR-CUT) filter. The system includes the illumination and imaging module, moving module, flipping module and machine vision algorithm. Authors used the stationary wavelet transform (SWT) which provides a more accurate estimate of the variances

in the image and further facilitates the identification of the defected regions. The experimental results show an accuracy of 96.44% of the proposed system.

Xue-Wu *et al.* [6] describe the design and testing process of a vision system to detect defects for strongly reflected metal surfaces. A computer vision based system was developed based on the wavelet transform, spectral measure, and support vector machine. The study used the wavelet smoothing method to eliminate noise from the images. Then, the images were segmented by the Otsu threshold. Finally, five characteristics based on the spectral measure of the binary images were collected and entered into a support vector machine (SVM). The classification results demonstrate that the proposed method can effectively identify seven classes of metal surface defects.

Aarthi *et al.* [7] propose a new method to explicitly analyze surface defects. The study developed a discrete wavelet transform technique to diagnose the defect in machineries. In the testing for welding flaws, the reliability and quality of the tests are considerably affected by noise and spurious signals. Thus, signal de-noising and an increase in the signal-noise ratio (SNR) are the key to successful application. Using the discrete wavelet transform, the flaw was isolated using the threshold of the transformed image, and different statistical features were studied.

Zheng *et al.* [8] develop an inspection system to detect structural defects on bumpy metallic surfaces, particularly holes and cracks on the surfaces of aluminum. Their system is based on the morphology and genetic algorithms and was validated with a database collected from industrially produced aluminum samples. The maximum overall accuracy was 91%.

Huang & Ye [9] propose a machine vision system to inspect the micro-spray nozzle. Canny edge detection, a randomized algorithm to detect circles, the circle inspection algorithm and the BPNN classifier were used as the image processing algorithms.

Malarvel and Singh [10] propose an autonomous technique for weld defects detection and classification using multi-class support vector machines in X-radiography images. Porosity, gas pore, tungsten inclusion, longitudinal crack, lack of penetration, slag inclusion weld defects are classified using multi-class support vector machines. Overall accuracy of 97% and 95% was achieved for circular and rectangular groups of weld defects, respectively.

Image segmentation is an important stage in detecting defects in images. The first study in the literature to use image segmentation for error detection was Funck *et al.* [11] in 2003. The authors present the results from the use of various algorithms for wood surface feature detection. Accuracy analysis of different segmentation algorithms has been made on colored wooden images and the results have been presented.

Malarvel *et al.* [12] propose an improved version of Otsu's method for segmentation of weld defects on X-radiography images.

Matic *et al.* [13] propose a novel real-time segmentation method for images from ceramic tile production lines. The improved method for parallel execution on a GPU for real-time requirements was implemented in the study.

Radi *et al.* [14] propose the segmentation techniques for weld defects with horizontal shapes. Proposed method compared to traditional approaches for detection of weld defects and achieved high segmentation accuracy.

Wei *et al.* [15] propose a surface defect segmentation method based on defect training samples using image processing techniques.

The existing studies, other than [1], related to image processing techniques are not efficient for inspection (classification of the products and defect segmentation) of metallic, cylindrical, non-uniform textured, and highly reflective objects. Also, the algorithms and the methods used are not optimized for real-time work.

B. DEEP LEARNING TECHNIQUES
In this subsection, the existing works related to defect detection, classification, and segmentation on surfaces with non-uniform textures using deep learning techniques are summarized.

Samet *et al.* [2] propose the deep learning approach for automated surface inspection systems. The approach has a successful outcome with the use of powerful paradigms in deep learning, including transfer learning and data augmentation. The accuracy of 97% was achieved using the VGG16 model. The results indicate that machine learning models using convolutional neural networks (CNNs) are capable of surface defect detection on military cartridge case samples.

Jiang et. al. [16] propose a method weakly-supervised CNN model to recognize defects based on casting X-ray images. Besides, a novel data-augmentation method guided by these attention maps is proposed to enlarge the dataset. The test accuracy of 95.5% and the recall of 96.0% were achieved using the proposed method.

Würschinger *et al.* [17] use the deep learning model and transfer learning method in a computer vision system and implement it to a manufacturing system to increase the quality.

Wang *et al.* [18] propose a new deep learning model to detect defects of aluminum alloy casting on the X-ray images. Another paper that performs defect detection using deep learning on aluminum alloys is the study of Chen *et al.* in 2021 [19]. The authors propose to use computer vision and deep learning techniques to achieve automatic detection of various defects of aluminum alloys.

Westphal and Seitz [20] propose CNN-based machine learning algorithms for error detection using complex transfer learning. Complex transfer learning methods were presented and an accuracy of 95.8% was achieved.

Wang *et al.* [21] propose a four-stage defect detection model, which uses CNNs. An electronic components production line with two cameras automated optical inspection

system is modeled and the proposed model achieved high precision and velocity.

He *et al.* [22] present a technique using computer vision and deep learning for defect detection and defect classification on hot rolled steel.

Zhang *et al.* [23] detect the foreign objects in coal with computer vision using CNN. They proposed a model with an accuracy of 97% that can be used in real-time.

There are also studies that detect defects on different materials using CNN. Defect-detection studies are on glass panels [24], wood [25], and fabric [26-30].

Mordia and Verma [31] present a review of some techniques based on vision system for detection surface defects. 48 studies on visual defect detection in steel products, including image processing and machine learning methods, were reviewed.

In another review, Tulbure *et al.* [32] offer a structured and analytical advantages and disadvantages of the object detection models. In review, region based CNNs, YOLO, SSD and cascaded architectures were evaluated.

Shi and Chen [33] propose the defect detection algorithm for layer-wise of powder bed. CNN were applied to implement the classification of the defects, and their performances were evaluated and compared. Proposed algorithm extracts the defects in a single image in 15.67 ms with a detection rate of 99.44%.

Xu *et al.* [34] propose a novel tunnel defect inspection method based on the Mask R-CNN. Xiao and Buffiere [35] developed an image segmentation method based on a convolutional neural network for fatigue crack images.

Du *et al.* [36] propose an intelligent defect detection system based on deep learning of aluminum casting parts' X-ray images.

Xu *et al.* [37] propose an automatic welding defect detection system based on semantic segmentation method using Deep CNN. The authors of [38] and [39] present the surface defect detection and segmentation of Deep CNN using different data sets.

Kheradmandi and Mehranfar [40] present a review on image segmentation-based techniques for pavement crack detection. In the review comparison and analysis of various image segmentation algorithms are offered.

In recent years, it is seen that the studies on defect detection, classification, and segmentation are mainly deep learning based. The existing studies related to the deep learning techniques come to behave like human sense and can detect such kinds of defects. As seen above, there are a few numbers of studies to detect defects, classify the objects and segment the defects using deep learning techniques. The problem with the existing studies is that the accuracy of the existing deep learning techniques is not sufficient for the inspection of critical tasks like military cartridge cases. On the other hand, there are no works, other than [2], in the literature related to the deep learning techniques to classify the military cartridge cases and segment the defects. Tech-

nological development in deep learning is used to solve this problem.

In this study, methodology for classification of the military cartridge cases and segmentation of defects on their surfaces using deep learning techniques is proposed to increase the accuracy.

III. DEEP CONVOLUTIONAL NEURAL NETWORKS

Deep convolutional neural networks (Deep CNNs) have gained great success in recent years, and they have been successfully applied to many computer vision tasks such as image classification, object detection, and image segmentation with the development of CNNs [41].

Existing deep learning techniques can be divided into four categories: 1) structural; 2) statistical; 3) filter-based; and 4) model-based approaches [42]. These techniques provide good results on uniform textured, non-reflective surfaces but most of them are dependent on application and environment variables but not very effective for non-uniform textured and highly reflective samples like the surface of military cartridges [1].

CNNs are widely applied for image-related analysis. CNNs normally consist of three major layers: 1) convolutional layer; 2) pooling layer; and 3) fully connected layer. The convolutional layer applies a number of filters on the local regions of the input, thus obtaining the feature maps of the input image. The pooling layer downsamples inputting spatial dimensions applied after the convolutional layer to reduce the feature dimension and to avoid overfitting problem. The fully connected layers normally constitute the last few layers of CNNs, computing the class scores. A Deep CNN normally consists of alternating convolutional and pooling layers, followed by fully connected layers [43], [44].

In this section, the Deep CNNs that are proposed to classify the military cartridge cases and segment the defects on their surfaces are presented.

A. DEEP CONVOLUTIONAL NEURAL NETWORKS TO CLASSIFY THE MILITARY CARTRIDGE CASES

Following are the models used in this work for classification of the objects like military cartridge cases: (1) VGGNets, (2) ResNets, (3) Inception, (4) Xception, (5) DenseNet and (6) EfficientNet.

1) VGGNETS

VGG network is a deep convolutional neural network whose main contribution was the study of the effect of depth of networks in the classification of large-scale images while using an architecture with a very small convolution filter (3 × 3) [45]. VGG-16 and VGG-19 are two known VGG network examples where 16 and 19 determine the number of weighted layers. VGG networks are made up of convolution, max pooling, activation, and fully connected layers. Both trained on more than 14 million images belonging to 1000 classes VGG16 and VGG19 models respectively

achieve almost 90.1% and 90.0% top-5 test accuracy on ImageNet dataset [46].

2) RESNETS

ResNets (Residual Networks) [47] use a technique called skip connections. Along the network these connections skip training flow from a few layers and connect directly to the output. This approach allows the layers across the network to fit the residual mapping instead of learning the underlying mapping. So, instead of initial mapping, H(x), the network fits F(x):=H(x)–x and gives H(x):=F(x)+x as output. ResNets authors evaluated this architecture on the ImageNet dataset with a depth of 152 layers, 8x deeper than VGGNets [45] but still being less complex. Some of the most known ResNets based models are ResNet50, ResNet101, ResNet152 where 50, 101, 152 determine the number of layers in the network. In 2016, the authors of ResNets brought a new improvement to ResNets which they called ResNetV2 [48]. The prominent improvement is the insertion of batch normalization and ReLU activation before 2D convolutions. The models like ResNet50V2, ResNet101V2, and ResNet152V2 are the second versions of the related model.

3) INCEPTION

One of the biggest problems encountered in the development of deep neural networks is overfitting, the level of which can rise as the network gets deeper and deeper. To solve this problem C. Szegedy *et al.* proposed DCNN architecture [49]. The main idea behind this architecture is to use multiple filters of different sizes at the same level instead of using multiple layers. This results in a wider model with parallel layers rather than a deeper model. This architecture is designed from the building blocks symmetric and asymmetric which contain convolutions, average/max pooling, concatenations, dropouts, and fully connected layers. Having contributed a lot to the Computer Vision literature, this architecture has been reviewed many times and different versions have been developed. One of the best known of these versions is InceptionV3 [50]. Major improvements done on this version are factorization of large filters into smaller ones, spatial factorization into asymmetric convolutions, use of auxiliary classifiers and efficient reduction of grid size. In 2016, the developers of Inception model published a new article [51] in which they presented a new model resulting from the combination of Inception and ResNet architectures (so named InceptionResNet). Leveraging the breadth of the Inception architecture and the depth of the ResNet architecture, with its performance this has made a great contribution to image classification tasks.

4) XCEPTION

Standing for Extreme version of Inception, Xception [52] is a deep convolutional neural network proposed by F. Chollet from Google. Inspired by Inception, the author replaced the inception module in Inception architecture with depthwise separable convolutions. Having the same number of parameters as Inception, this architecture has been proven to perform better than InceptionV3 on datasets such as ImageNet (for which InceptionV3 was developed). The performance of this architecture is due to its efficient use of the model parameters.

5) DENSENET

Like the ResNet architecture, the DenseNet architecture [53] also uses dense blocks and skip connections. But unlike ResNet, in DenseNet each layer gets additional inputs from all previous layers and passes its own feature maps to all subsequent layers. Another difference is that DenseNet uses concatenations to combine features instead of summation. These multiple inputs allow DenseNet to limit the number of filters. In DenseNet layers are made up of pre-activation batch normalization, ReLU activation then 3×3 convolution with output feature maps of k channels and between contiguous dense blocks, 1×1 convolution and 2×2 average pooling are respectively used. Some advantages of DenseNet are: its strong gradient flow, maintenance of low complexity features and computational efficiency. Some of the most known DenseNet models are DenseNet101, DenseNet121 and DenseNet169, where 101, 121, and 169 determine the number of layers in the model.

6) EFFICIENTNET

EfficientNet [54] is a convolutional neural network architecture and model scaling method. Unlike conventional practices in which the depth, width and resolution dimensions of the network are arbitrarily scaled, named compound scaling method, this method uses a compound coefficient ø to uniformly scale these factors. For example, α, β, γ being fixed coefficients that can be determined by a small grid search and if we want to use a resource costing $2^{\wedge}ø$ we increase the depth of the network to $\alpha^{\wedge}ø$, width to $\beta^{\wedge}ø$ and resolution (input shape) to $\gamma^{\wedge}ø$. The need for such a method is justified by the intuition that the higher the image resolution, the deeper the network must be to have a large receptive field and more feature maps to capture finer patterns. The authors first demonstrated the efficiency of their method in scaling the ResNets and MobileNets architectures, and then, as the second part of this work, they developed a new base network that they scaled to obtain a family of networks called EfficientNets. All these models are made up of 7 inverted residual blocks that use all squeeze and excitation blocks with swish activation. Their only difference is at the level of the parameterization which allows them to have different dimensions of width/depth/resolution. Significant results have been obtained with EfficientNetB7 (84.3% top-1 accuracy on ImageNet) allowing it to have its place among SOTA models.

B. DEEP CONVOLUTIONAL NEURAL NETWORKS TO SEGMENT DEFECTS ON SURFACE OF MILITARY CARTRIDGE CASES

Following Deep CNNs are proposed to segment the defects on the military cartridge case images: (1) U-Net, (2) ResUnet and (3) DeepLabv3+.

1) U-NET

U-Net is architecture for semantic segmentation. U-Net [55] was developed for biomedical image segmentation that is based on the fully convolutional network. U-Net is an improvement and development of FCN [56]. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization so U-Net has a U shaped symmetric architecture.

2) RESUNET

Deep Residual U-Net [57] is referred to as ResUnet. It's a semantic segmentation encoder-decoder architecture created by Zhengxin Zhang *et al.* It was first employed in the field of remote sensing image analysis to extract roads from high-resolution aerial photos. Researchers later used it for a variety of other applications, including polyp segmentation, brain tumor segmentation, human image segmentation, and many others. ResUnet is a fully convolutional neural network that is designed to get high performance with fewer parameters. It is an improvement over the existing U-Net architecture. It combines the strengths of deep residual learning and U-Net architecture. This architecture brings two advantages: 1) the residual unit will ease training of the network; 2) The skip connections inside a residual unit and between low and high levels of the network will makes it possible to design a neural network with much fewer parameters however could achieve comparable ever better performance on semantic segmentation.

3) DEEPLABV3+

DeepLabv3+ [58] extends DeepLabv3 [59] by adding a simple yet effective decoder module to refine the segmentation results along object boundaries. It modified the main network again on the original basis. They are able to encode multi-scale contextual information by using atrous spatial pyramid pooling (ASPP). The encoder uses dilated convolution at multiple scales to handle multiscale contextual information, while the decoder module refines the segmentation results along object boundaries. The separable convolution is also explored, which makes the suggested model faster and stronger while reducing the computational complexity significantly.

C. TRANSFER LEARNING

The transfer of knowledge from a basic task to a more specific task remains an effective solution in the field where data acquisition is difficult, especially in machine/computer vision tasks. Besides being an efficient optimization procedure, the use of already pre-trained models on millions of data points with a large number of classes in the classification task also allows the improvement of the classification. The first layers of CNN-based models are used to learn low-level features such as edges and blobs while the top layers are used to learn high-level features that are more abstract and task-specific aspects of the image, such as military cartridge case defects.
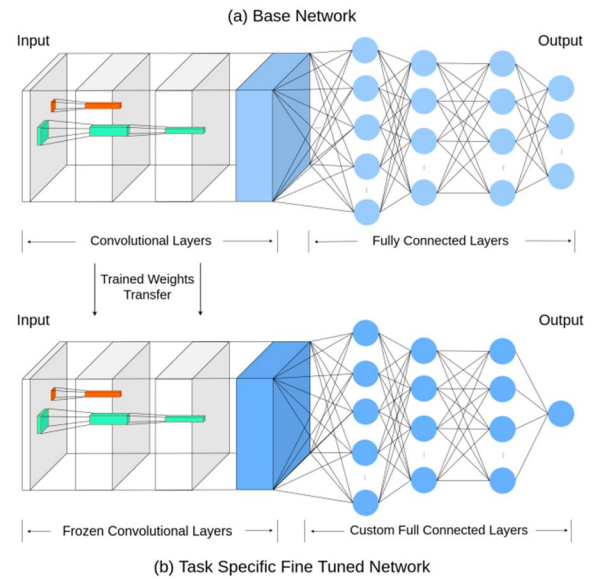

FIGURE 1. Transfer learning process adopted in this study.

However, as illustrated in Fig. 1, the most adopted way of using already pre-trained models in computer vision tasks is to train the top layers according to the task concerned while keeping the initial parameters of the first layers. This practice allows having fewer parameters to train, which reduces the risk of overfitting, which remains a big problem in training neural networks.

As seen, the trained weights of a base network (Fig. 1(a)) pre-trained on a large-scale dataset are transferred to a task specific fine-tuned network (Fig. 1(b)) to be trained for a new task. In the new task network, the first layer group is set untrainable by freezing the corresponding layers (see "Frozen Convolutional Layers" caption in Fig. 1(b)) and top layers are reconstructed according to the task (see "Custom Fully Connected Layers" caption in Fig. 1(b)).

Table 1 shows the details and performance on the ImageNets dataset [46], [60] of the different CNN models used in the image classification task which can also be used for the feature extraction and fine-tuning tasks.

IV. PROPOSED METHODOLOGY

The proposed methodology for classification of the military cartridge cases and segmentation of the defects on the surfaces of the military cartridge cases consists of four stages:
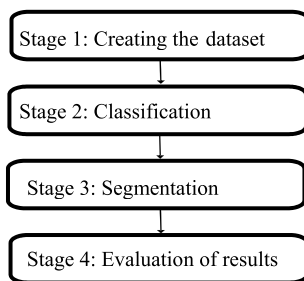
1. Creating the datasets;
2. Classification of the military cartridge cases;
3. Segmentation of the defects on the surfaces of the military cartridge cases;
4. Evaluation of the results.

The flow chart of the proposed methodology is shown as follows (Fig.2).

The stages of the proposed methodology are explained in detail in the following subsections.

TABLE 1. The details and performance of pre-trained Deep CNN models on ImageNets dataset.

Models	Sub-models	Size (MB)	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Parameters (M)	References
Xception		88	79.0	94.5	22.9	Chollet [52]
VGG	16	528	71.3	90.1	138.4	Simonyan and Zisserman [45]
	19	549	71.3	90.0	143.7	
ResNet	50	98	74.9	92.1	25.6	He et al. [47]
	101	171	76.4	92.8	44.7	
	152	232	76.6	93.1	60.4	
Inception V3		92	77.9	93.7	23.9	Szegedy et al.[49]
DenseNet	121	33	75.0	92.3	8.1	Huang et al. [53]
	169	57	76.2	93.2	14.3	
	201	80	77.3	93.6	20.2	
EfficientNet	B3	48	81.6	95.7	12.3	Tan and Le [54]
	B7	256	84.3	97.0	66.7	
	V2B3	59	82.0	95.8	14.5	
	V2S	88	83.9	96.7	21.6	
	V2M	220	85.3	97.4	54.4	
	V2L	479	85.7	97.5	119.0	

Stage 1: Creating the dataset

↓

Stage 2: Classification

↓

Stage 3: Segmentation

↓

Stage 4: Evaluation of results

FIGURE 2. The flow chart of the proposed methodology.

A. STAGE 1: CREATING THE DATASETS

The military cartridge case is the metallic, cylindrical and highly reflective object with non-uniform textured surface. There are no commonly used datasets for deep learning models to classify the military cartridge cases and to segment the defects on their surfaces in existing literature. So, special datasets should be created. Following are the main steps adopted in this work to create such dataset.

1) Taking the images from the cylindrical surface with sectors of 60 degrees using a machine vision system [1]. So, 6 images taken from the surface of each cartridge case should cover the whole 360-degrees surface without overlapping;

2) Preprocessing all images [1];

3) Dividing the images into smaller patches;

4) Creating the dataset for classification of the military cartridge cases by grouping all images into non-defective and defective classes;

5) Creating the dataset for segmentation of the defects on the surfaces of the military cartridge cases by labeling/masking the defects on the images in defective class created in previous step;

So, at Stage 1 of the methodology, 2 datasets are created: 1) Dataset for classification and 2) Dataset for segmentation. The first dataset will be used at Stage 2 to classify the military cartridge cases as defective or non-defective. On the other hand, the second dataset will be used at Stage 3 to segment the defects on the surfaces of the military cartridge cases.

B. STAGE 2: CLASSIFICATION OF THE MILITARY CARTRIDGE CASES

Following are the main steps to classify the military cartridge cases.

1) Using the classification dataset with non-defective and defective classes created at Step 4 of Stage 1;

2) Selecting the deep learning model(s) presented in Section III-A;

3) Training and validating the selected model(s) on the training and validation sets generated from the classification dataset, respectively;

4) Evaluating the trained models on the test set using the metrics described at Stage 4;

5) If the obtained values of metrics are appropriate, finish the classification stage. Otherwise trying to increase the values of metrics using one or some of the following techniques:

(a) Fine-tuning the model architecture and/or hyperparameters;

(b) Optimizing the network using different neural network optimizers;

(c) Applying preprocessing and/or post processing to the datasets.

C. STAGE 3: SEGMENTATION OF THE DEFECTS ON THE SURFACES OF THE MILITARY CARTRIDGE CASES

Following are the main steps to segment the defects.

1) Using the segmentation dataset with labeled/masked defects created at Step 5 of Stage 1;

2) Selecting the deep learning model(s) presented in Section III-B;

3) Training and validating the selected model(s) on the training and validation sets generated from the segmentation dataset, respectively;

4) Evaluating the trained models on the test set using the metrics described at Stage 4;

5) If the obtained values of metrics are appropriate, finish the defect segmentation stage. Otherwise trying to increase the values of metrics using one or some of following techniques;

(a) Fine-tuning the model architecture and/or hyperparameters;

TABLE 2. Confusion matrix.

Confusion Matrix	Estimated Positive	Estimated Negative
True Positive	True Positive (TP)	False Negative (FN)
True Negative	False Positive (FP)	True Negative (TN)

(b) Optimizing the network using different neural network optimizers;

(c) Applying preprocessing and/or post processing to the datasets.

D. STAGE 4: EVALUATION OF THE RESULTS

Various evaluation metrics are used to determine how successful the classification of the military cartridge cases and the segmentation of the defects on the surfaces of the military cartridge cases are. The values used in these metrics are obtained from the confusion matrix (Table 2).

In order to evaluate the obtained results of classification and segmentation the following metrics can be used: (1) Accuracy; (2) Precision; (3) Recall; (4) F1-Score; (5) Jaccard Index; (6) Mean Intersection over Union.

1) ACCURACY

Accuracy gives information about the overall performance of the model. The accuracy score is between 0 and 1, and the closer to 1, the higher the success of the model. It should be used in cases where the class variable is evenly distributed in the dataset. In an unevenly distributed dataset, the model tends to predict the value of the majority class, resulting in a biased high accuracy result. For this reason, other metrics should be considered besides accuracy to evaluate the model established with unevenly distributed data sets.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

2) PRECISION

Precision measures how many of the cartridge cases predicted as non-defective are actually non-defective. Precision is critical, especially when the cost of FP estimation is high.

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

3) RECALL

Recall measures how many of the cartridge cases predicted as defective are actually defective. Recall can be thought of as a measure of the integrity of the classifiers. Recall is a critical metric where the cost of FN estimation is high. It should be as high as possible.

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

4) F1-SCORE

F1-Score is the harmonic mean of precision and recall values. F1-Score provides the balance between precision and recall. F1-Score is important as it fulfills the need for an evaluation

FIGURE 3. Examples for the cartridge cases.

metric that will include all error costs. The reason for using harmonic mean instead of arithmetic mean is to prevent ignoring data samples with extreme values. It is a better guide than the accuracy metric, especially in unevenly distributed data sets.

$$F1\ score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

$$F1\ score = \frac{2 \times TP}{(2 \times TP + FP + FN)} \quad (5)$$

5) JACCARD INDEX

JI, also called the IoU score (Intersection over Union) calculates the percentage of pixel overlap between the segmented output and the ground truth.

$$JI (or\ IoU) = TP/(TP + FP + FN) \quad (6)$$

6) MEAN INTERSECTION OVER UNION

$$mean\ IoU = \frac{1}{K} \sum_{k=1}^{K} IoU_k \quad (7)$$

V. IMPLEMENTATION OF PROPOSED METHODOLOGY

In this section, the implementation details and results of the proposed methodology on military cartridge cases are presented.

A. CREATING THE DATASETS

This paper aims to classify the military cartridge cases and segment the defects on their surfaces. As seen in Fig. 3. the military cartridge cases have a metallic, cylindrical, non-uniform textured and highly reflective surface.

In order to detect, classify and segment the defects such as corrosion, scratch, crack, split, dent, fold, bulge, buckle, and wrinkle on the surface of the military cartridge cases [61] using deep learning techniques the dataset is required.

A custom prototype machine was built, and images are collected from the prototype machine. Below are the main steps adopted to create the dataset.

1) Taking the images from the cylindrical surface of the cartridge case with sectors of 60 degrees using a machine
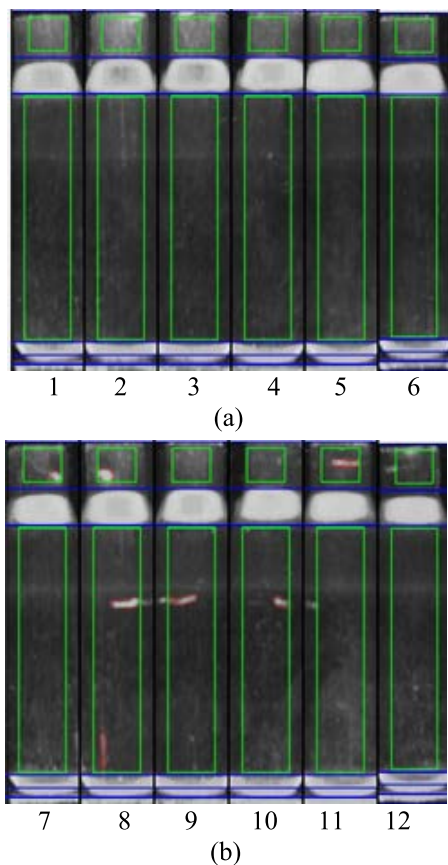
FIGURE 4. Six images taken from the cylindrical surface of (a) the non-defective and (b) the defective cartridge cases with sectors of 60 degrees.

FIGURE 5. (a) Six images obtained from the non-defective cartridge case shown in Fig.4(a) and (b) Six images obtained from the defective cartridge case shown in Fig.4(b).
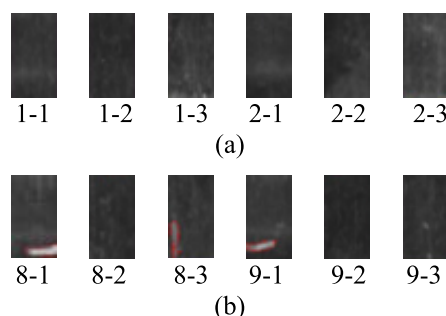
FIGURE 6. Examples for divided images of: (a) the non-defective (Fig. 5 (a)) and (b) the defective (Fig. 5 (b)) cartridge cases.
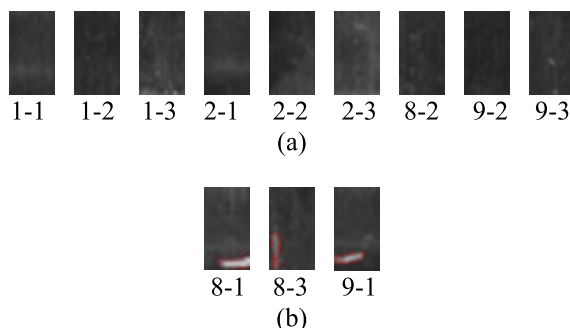
FIGURE 7. Example for (a) non-defective and (b) defective classes of dataset.

vision system [1]. So, 6 images taken from the surface of each cartridge case should cover the 360-degree surface without overlapping (Fig. 4).

2) Extracting the surface parts. In this study, only the surface part (see the big green rectangles in Fig. 4) of the cartridge cases was under consideration. So, 6 images are obtained from each cartridge case (Fig. 5).

3) Dividing each image in Fig.5 into 3 equal parches (Fig.6).

4) Creating the non-defective and defective classes of dataset that will be used for classification by grouping images with defective and non-defective ones (Fig. 7).

By repeating the steps 1-4 for many cartridge cases, the non-defective image class of dataset with 750 images and the defective image class of dataset with 600 images were obtained. Defective class of dataset was increased to 750 images by using data augmentation and creating synthetic defects. After this process, a balanced dataset that has 2 classes with 750 non-defective and 750 defective images was created.

5) Creating the dataset for segmentation of the defects on the surfaces of the military cartridge cases by labeling/masking the defects on the images in the defective class

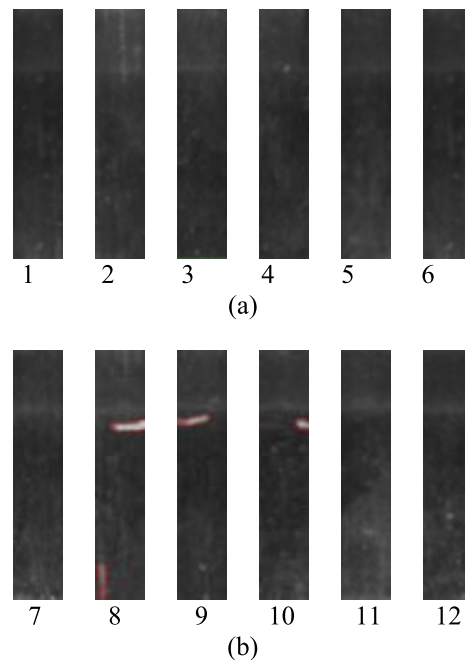created in the previous step. Examples for images (masks) in the labeled dataset are shown in Fig. 8.

So, the non-defective and the defective classes of dataset created for classification in the Step 4 will be used below to classify the military cartridge cases. On the other hand,

8-1   8-3   9-1
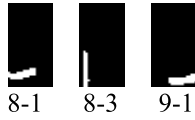
FIGURE 8. Examples for images (masks) in labeled dataset for defect segmentation.

dataset obtained by labeling/masking the images in the defective class of dataset created in the Step 4 will be used below to segment the defects.

B. CLASSIFICATION OF THE MILITARY CARTRIDGE CASES
In this section, the implementation setup and results of the classification of military cartridge cases are presented.

1) IMPLEMENTATION SETUP
In this study, deep learning models listed in Table 1 were used to classify the military cartridge cases as defective or non-defective. These models were implemented using Keras neural network library running on top of Tensorflow framework [60]. The input size of the models was set to $128 \times 128 \times 3$ and the images in the dataset were resized accordingly. Training/validation/test splitting ratio was set to 60:20:20 which allowed the training, validation and test sets to preserve the general trends of the original dataset. Due to the use of models already pre-trained on large datasets the number of epochs (the number of rounds the model completes through the entire training dataset) was chosen to set to 20, SGD was used as default optimizer with a mini-batch size of 32. The learning rate was set to 0.01. Due to its effectiveness in binary classification, the cross-entropy loss function was used. Accuracy metric was chosen to evaluate the trained models.

2) IMPLEMENTATION RESULTS
The Deep CNNs models were trained and compared based on their accuracy on the test dataset (Table 3).

Columns 3 and 4 in Table 3 show the implementation results using initial/default models/sub-models listed in columns 1 and 2.

To obtain more efficient and task-specific models, a 2-stage performance upgrade study was adopted.

The first stage was to fine-tune the initial models. Columns 5, 6 and 7 in Table 3 show the fine-tuning implementation results. The fine-tuning was carried out as follows:

1. The number of dense layer neurons in VGGNets was reduced by half;

2. The dropout ratio decreased from 0.5 to 0.2;

3. The top of the models like ResNets, EfficientNets; which uses average pooling and dense layers at the head of the model, has been replaced by a sequence of layers which is as follows: flatten - dense (2048 neurons) – dropout (0.2 ratio) - dense (2048 neurons) – dropout (0.2 ratio) - dense (1 neuron). The output of the last layer corresponds to the probability that the input is a non-defective cartridge case.

TABLE 3. Implementation results of classification of the military cartridge cases.

Models	Sub-Models	Accuracy of Initial Models (%)	Number of Parameters of Initial Models (M)	Accuracy of Fine-Tuned Models (%)	Number of Parameters of Fine-Tuned Models (M)	Fine-Tuning effect (%)	Maximum Accuracy
1	2	3	4	5	6	7	8
VGG	16	0.980	65.06	**0.987**	33.60	+00.7	**0.987**
	19	0.987	70.37	0.987	38.90	±00.0	0.987
ResNet	50	0.987	23.59	0.990	29.88	+00.3	0.990
	101	0.980	42.66	**0.990**	48.95	+01.0	**0.990**
	152	0.983	58.37	0.990	64.67	+00.7	0.990
DenseNet	121	0.980	07.04	0.983	11.24	+00.3	0.983
	169	0.980	12.64	**0.987**	18.15	+00.7	**0.987**
	201	0.980	18.32	0.983	24.36	+00.3	0.983
Xception		**0.970**	20.86	0.963	27.16	- 00.7	**0.970**
InceptionV3		**0.980**	21.80	0.970	28.10	- 01.0	**0.980**
EfficientNet	B3	0.970	10.79	0.983	63.22	+01.3	0.983
	B7	0.907	64.10	0.940	150.09	+03.3	0.940
	V2B3	0.953	12.93	0.993	65.36	+04.0	0.993
	V2S	0.883	20.33	**0.993**	64.38	+11.0	**0.993**
	V2M	0.870	53.15	0.963	97.19	+09.3	0.963
	V2L	0.697	117.75	0.953	161.79	+25.6	0.953

The majority of pre-trained models are complex models specially designed for large datasets like ImageNet. Therefore, fine-tuning is a critical process in transferring knowledge from these models to a specific task network to avoid problems like overfitting and non-generalization during model training. As shown in Table 3, except for the Xception and InceptionV3 models, the above fine-tuning processes allowed us to obtain more efficient and task-specific models than the default models which did not generalize as well as expected in our experiments, especially the default EfficientNet models. The fine-tuning has also made it possible to reduce the number of parameters almost by half of the VGGNets models while keeping the same good results or even better than the default models. During model training, the weights of the convolutional neural network filters are optimally adjusted to minimize the loss function. The direction and magnitude of this adjustment is very dependent on the optimizer used to train the model.

The effect of this fine-tuning was measured as a percentage according to the accuracy obtained (increase / neutral / decrease) (see column 7 in Table 3). Then, for each model, the one with the highest accuracy was selected (see results

TABLE 4. Evaluation of optimizers' performance on Deep CNNs pre-trained models.

	VGG 16	ResNet 101	Dense Net169	Xception	Inception V3	Efficient NetV2S
Optimizer	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy
SGD	0.987	0.990	0.987	0.970	0.980	0.993
RMSprop	0.987	0.987	**1.000**	0.900	0.970	0.967
Adam	0.980	0.987	**1.000**	**0.980**	0.980	0.987
Nadam	0.980	0.980	**1.000**	**0.980**	0.963	0.987
Adagrad	0.983	0.990	0.987	0.887	0.963	0.983

TABLE 5. Implementation results of defect segmentation of the military cartridge cases.

Models/Sub-models		mIoU	JI	F1-Score
Standard U-Net		0.843	0.699	0.823
Improved U-Net		**0.924**	**0.854**	**0.921**
ResUnet		**0.924**	**0.854**	**0.921**
Deeplabv3+	R50 (ImageNet 1k)	0.909	0.825	0.904
	R101 (ImageNet 1k)	0.907	0.822	0.902
	R101 (ours)	0.909	0.825	0.904
	VGG16 (ImageNet 1k)	0.901	0.809	0.894
	VGG16 (ours)	0.906	0.818	0.900
	DenseNet169 (ImageNet 1k	0.909	0.826	0.904
	DenseNet169 (ours)	0.912	0.831	0.908
	Xception (ImageNet 1k)	0.908	0.823	0.903
	Xception (ours)	0.908	0.822	0.903
	Inceptionv3 (ImageNet 1k)	0.913	0.832	0.908
	Inceptionv3 (ours)	0.913	0.832	0.908
	EfficientNetV2S (ImageNet 1k)	0.914	0.835	0.910
	EfficientNetV2S (ours)	0.915	0.837	0.911

marked in bold in columns 3 and 5). The best result was obtained with the fine-tuned EfficientNetV2S with 99.3% accuracy with an improvement of 11.0% on the initial model and the lowest result with the initial EfficientNetV2L (69.7% accuracy) whose performance was also improved by 25.6% by fine-tuning. According to their accuracies, for each architecture shown in column 1 in Table 3, the model with the highest performance was selected as Maximum Accuracy as shown in column 8 in Table 3.

At the second stage of performance upgrade, the selected models of fine-tuned VGG16 for VGGNets, fine-tuned ResNet101 for ResNets, fine-tuned DenseNet169 for DenseNet, initial Xception, initial InceptionV3 and fine-tuned EfficientNetV2S for EfficientNet were trained using different optimizers such as SGD, RMSprop, Adam, Nadam, Adagrad and obtained results were compared in Table 4. As seen in Table 4, the improved results are marked in bold. While this optimizer setting did not have a positive effect on some models such as fine-tuned VGG16, fine-tuned ResNet101, and initial InceptionV3 and fine-tuned EfficientNetV2S, it allowed the performance improvement of models like fine-tuned DenseNet169 which achieved 100.0% accuracy with RMSProp, Adam, and Nadam optimizers.

C. SEGMENTATION OF THE DEFECTS ON THE SURFACES OF THE MILITARY CARTRIDGE CASES

In this section, the implementation setup and results of the segmentation of the defects on the surfaces of the military cartridge cases are presented. Besides, the details of implementation of the improved U-Net, ResUnet and DeepLabv3+ models are also presented.

1) IMPLEMENTATION SETUP

In this section the images (750 in total) from the defective cartridge case image class of the dataset for classification created in Step 4 of Section V-A were labeled/masked. As a result, the dataset with labeled/masked images was created for defect segmentation. This dataset was split into train, test and validation classes in 70:15:15 ratios. The dataset images have been resized to 128×128 and the CNN models' input has been adjusted accordingly. U-Net [54], ResUnet [57, 62] and

DeepLab3+ [58] are the three main CNN models used for the defect segmentation. All models are implemented using the Keras API with Tensorflow 2.8 as backend [60]. Training, validation and testing of models were done on Google Colab pro using an NVIDIA Tesla P100 GPU with 32 GB (27.3 usable) of RAM. The Adam optimizer [63] was used as the default optimizer for calculating optimal weights during backpropagation along the model. Binary cross entropy loss function was used. The mini-batch size and the learning rate were set to 32 and 0.01, respectively. The learning rate is decreased by a factor of 0.1 when the loss function does not register a decrease for 5 successive epochs. To avoid the vanishing gradients problem the minimum learning was set to 1e-6. We trained all models for 60 epochs. Inspired by the [64] experiment the threshold was set to 0.50.

2) IMPLEMENTATION RESULTS

Mean Intersection over Union, Jaccard Index and F1-Score metrics were used for all models to evaluate the segmentation results. Implementation results are shown in Table 5. It can be seen that the best result was obtained with the improved U-Net and ResUnet models with mIoU=0.924, JI=0.854, F1-Score=0.921. Below are the improvement details of these models.

3) IMPROVED U-NET

We have significantly improved the original U-Net model to suit our problem while keeping the general structure of the architecture. The architecture input was set to 128 × 128 × 3
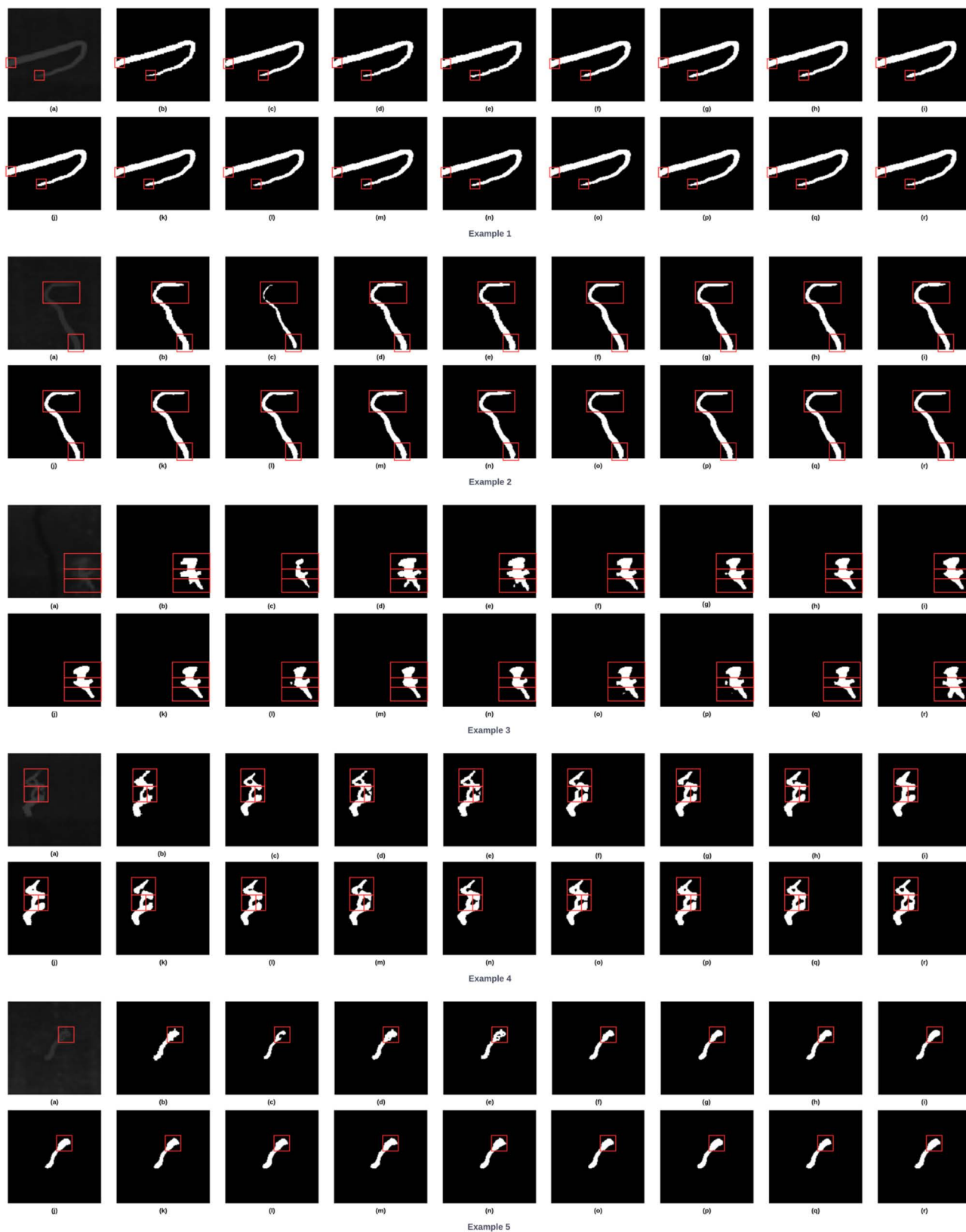
FIGURE 9. Improved U-Net.

FIGURE 11. Encoder/decoder blocks of the improved ResUnet.

FIGURE 10. Encoder/Decoder Blocks of Standard ((a), (b)) and Improved U-Net ((c) and (d)).

FIGURE 12. Architecture of DeepLabv3+.

and for feature map extraction, 4 successive coding blocks were used along the contraction path (Fig. 9).

For the downsampling through the contraction path each block is followed by a 2 × 2 max pooling operation with stride 2. The number of feature maps (number of channels) is doubled at each downsampling step [54] to iteratively extract new features. Dropouts at different ratios (0.25, 0.50 and 0.75) were applied to regularize the network [65]. The last encoder block was followed by a bottleneck with a feature map size of 256 and a drop rate of 0.50 which ensured feature map compression and the transition of these compressed feature maps to the decoder (expansive path) [66]. In the expansive path max pooling operations were replaced with 2 × 2 convolution transpose operations to up-sample the feature maps then the skip connections were used to concatenate the feature maps of the shallow and deep layers of the contraction path with the high-level features (semantic and spatial information) constructed through the expansive path [67]. At the final layer we performed a 1 × 1 convolution

to map the features to 1 channel where the value of each pixel represents that pixel to be a part of the defect regions in the corresponding input image. To solve the interval covariate shift [67] we used a batch normalization layer after each convolution layer (before activation function) in the network and parameters were set as epsilon=1e-3, beta_initializer=0.01, gamma_initializer=1.0, momentum=0.5. The weights of the convolutional filters in the network were initialized using Xavier's uniform initializer ("glorot_uniform initializer") [68]. Fig. 10. illustrates the encoder and decoder blocks of the standard ((a) and (b)) and improved ((c) and (d)) U-Net architectures.

4) RESUNET

We implemented the ResUnet model according to [62] where Zhang *et al.* combined the U-Net architecture [54] and the strengths of residual learning [47] for the segmentation of road areas. The main difference between the implementation of ResUnet and U-Net is the use of the residual unit in the ResUnet model as encoder and decoder blocks. The residual

FIGURE 13. Examples from obtained results on the test set for defect segmentation. (a) Input image; (b) Mask (Ground Truth); (c) Standard U-Net; (d) Improved U-Net; (e) ResUnet; (f) DeepLabv3+ (ResNet-50-I1k); (g) DeepLabv3+ (ResNet-101-I1k); (h) DeepLabv3+ (ResNet-101-Ours); (i) DeepLabv3+ (VGG-16-I1k); (j) DeepLabv3+ (VGG-16-Ours); (k) DeepLabv3+ (DenseNet-169-I1k); (l) DeepLabv3+ (DenseNet-169-Ours); (m) DeepLabv3+ (Xception-I1k); (n) DeepLabv3+ (Xception-Ours); (o) DeepLabv3+ (InceptionV3-I1k); (p) DeepLabv3+ (InceptionV3-Ours); (q) DeepLabv3+ (EfficientNetV2S-I1k); (r) DeepLabv3+ (EfficientNetV2S-Ours).

TABLE 6. Comparison between improved U-Net and ResUnet models.

Model	F1-score	Precision	Recall
Improved U-Net	0.921	0.916	0.927
ResUnet	0.921	0.921	0.922

unit is composed of convolution, activation and batch normalization layers and an addition layer that combines the input feature maps of this unit with its output. This unit can be illustrated as a general form:

$$y_l = h(x_l) + F(x_l, W_l), \quad x_{l+1} = f(y_l), \qquad (8)$$

where x_l, x_{l+1} and W_l are respectively the input, output and weights of the filters of the first residual unit. $F(\cdot)$ is the residual function, $h(x_l)$ is the identity mapping function $h(x_l) = x_l$ and $f(y_l)$ is the activation function. Just like in the U-Net network implementation, the weights of the convolutional filters in the ResUnet network were also initialized with Xavier's uniform initializer [68]. Fig. 11 shows the residual units used in ResUnet architecture as encoder and decoder blocks.

It can be observed in Table 5 that in general the models based on DeepLabv3+ architecture in which we used the weights pre-trained on our dataset during classification task perform better than those in which the ImageNet pre-trained weights were used. Among these models Deeplabv3+ EfficientNetV2S (ours) performed best with mIoU=0.915, JI=0.837, F1-Score=0.911 metrics results. Below is the improvement detail of this model.

5) DEEPLABV3+

DeepLabv3+ [58] was implemented using the same ASPP (Atrous Spatial Pyramid Pooling) architecture as in the original paper [58] with dilation rates of 1, 6, 12, 18 respectively and a feature channel size of 256. In order to benefit from the trained weights of the most efficient models developed in the classification section, we used the models in Table 3 (plus the ResNet-50 model) as backbone networks (feature extractor networks) in the DeepLabv3+ implementation, as illustrated in Fig. 12.

Two pre-trained weights were used in backbone networks: (1) ImageNet pre-trained weights (I1k) and (2) Cartridge cases dataset pre-trained weights (ours).

For a more specific comparison between U-Net and ResUnet models Precision and Recall were used. Table 6 shows the comparative results of the improved U-Net and ResUnet models used in this study for defective cartridge case segmentation.

If the cost of FP estimation is high ResUnet model, else if the cost of FN estimation is high the U-Net model should be used.

Fig. 13 gives the results on the test set of datasets for segmentation. Fig. 13 illustrates five examples of defective military cartridge case images segmented using the Deep

CNN models listed in Table 5. To provide a good visual understanding of the results, the defect regions have been outlined with red boxes. As can be observed in the Fig. 13, except for the standard U-Net model, all other models are able to segment simple defect surfaces well (see Examples 1 and 2 in Fig. 13), especially the improved U-Net and ResUnet models, which segment defect regions with high precision. In Example 3, we can see that the improved U-Net, ResUnet and EfficientNetV2S (ours) models are the most efficient models (in accordance with the results of Table 5) in the segmentation of less visible defects. Because the labeling of segmentation data is on a per-pixel basis, it is difficult to perform segmentation perfectly. Therefore, it can be seen that some pixels are missed as a result of labeling. As a result of the studies, it can be observed that our improved U-Net and ResUnet models are able to segment these defects (missed at the labeling stage) with very high precision (as shown in Example 4). As reported in Table 6, ResUnet performs better than the improved U-Net model in terms of precision while the latter has the highest recall. This qualitative comparison is illustrated in Example 5 where ResUnet predicts the defective surface of the military cartridge with higher precision than all other models as well as the mask in which some defective pixels are missed. It can also be noticed in some complicated examples the better performance of the models initialized with our pre-trained weights above those initialized with the weights pre-trained on ImageNet (g-h, q-r pairs in Example 3 and i-j, k-l pairs in Example 4).

VI. CONCLUSION
Low accuracy, low performance and high cost are the disadvantages of manual quality control of industrial products, especially military products like cartridge cases. To find solutions to these disadvantages the automatic quality control using machine vision systems is widely used. These systems take the images from the surfaces of objects by cameras during the production process in real-time. So, the quality of images is very important to get suitable accuracy of defects detection. Another factor that affects the accuracy is the non-uniform texture of the product surface. Distinguishing the product non-uniform texture from defect texture is a challenging problem. In order to bring the solutions to this problem, the methodology for classification of the military cartridge cases and segmentation of defects on their surfaces with non-uniform texture was proposed to increase the accuracy. In the scope of proposed methodology, the dataset was created and the deep learning models for classification and segmentation were developed and implemented. In the existing literature, the accuracy of classification of 97% and 96% of the military cartridge cases have been obtained with image processing [1] and deep learning [2] techniques, appropriately. In this study, the accuracy of the existing literature has been improved. In other words, obtained results have shown that the proposed methodology has increased the accuracy of classification of the military cartridge cases to 100% with the

DenseNet169 model and the F1-Score of defect segmentation to 92.1% with improved U-Net and ResUnet.

The main finding of this study is that the deep learning techniques have potential to reach the required NATO standards for classification and segmentation of the defects on the surface of the military cartridge cases. Another finding is that the obtained results have proved that the deep learning techniques can distinguish the defect textures from non-uniforms textures on the surfaces of the cartridge cases better than image processing techniques.

As feature work, the effect of data argumentation applied to our dataset on the defect detection and segmentation will be examined. Also, improved U-Net and ResUnet models which have given the best results will be optimized.

REFERENCES

[1] S. Tural and R. Samet, "Automated defect detection on surface of military cartridges," *J. Modern Technol. Eng.*, vol. 4, no. 3, pp. 178–189, 2017.

[2] R. Samet and S. Tural, "Deep learning based defect detection of metallic surfaces," in *Proc. 7th Int. Conf. Control Optim. Ind. Appl.*, Aug. 2020, pp. 362–364.

[3] J. P. Yun, D.-C. Choi, Y.-J. Jeon, C. Park, and S. W. Kim, "Defect inspection system for steel wire rods produced by hot rolling process," *Int. J. Adv. Manuf. Technol.*, vol. 70, nos. 9–12, pp. 1625–1634, Feb. 2014.

[4] A. Chondronasios, I. Popov, and I. Jordanov, "Feature selection for surface defect classification of extruded aluminum profiles," *Int. J. Adv. Manuf. Technol.*, vol. 83, nos. 1–4, pp. 33–41, Mar. 2016.

[5] Y. Liu and F. Yu, "Automatic inspection system of surface defects on optical IR-CUT filter based on machine vision," *Opt Lasers Eng.*, vol. 55, pp. 243–257, Apr. 2014.

[6] Z. Xue-Wu, D. Yan-Qiong, L. Yan-Yun, S. Ai-Ye, and L. Rui-Yu, "A vision inspection system for the surface defects of strongly reflected metal based on multi-class SVM," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5930–5939, May 2011.

[7] T. Aarthi, M, Karthi, and M. Abinesh, "Detection and analysis of surface defects in metals using wavelet transform," *Int. J. Sci. Res. Publications*, vol. 3, no. 6, pp. 1–6, 2013.

[8] H. Zheng, L. X. Kong, and S. Nahavandi, "Automatic inspection of metallic surface defects using genetic algorithms," *J. Mater. Process. Technol.*, vols. 125–126, pp. 427–433, Sep. 2002.

[9] K.-Y. Huang and Y.-T. Ye, "A novel machine vision system for the inspection of micro-spray nozzle," *Sensors*, vol. 15, no. 7, pp. 15326–15338, Jun. 2015.

[10] M. Malarvel and H. Singh, "An autonomous technique for weld defects detection and classification using multi-class support vector machine in X-radiography image," *Optik*, vol. 231, Apr. 2021, Art. no. 166342.

[11] J. W. Funck, Y. Zhong, D. A. Butler, C. C. Brunner, and J. B. Forrer, "Image segmentation algorithms applied to wood defect detection," *Comput. Electron. Agricult.*, vol. 41, nos. 1–3, pp. 157–179, Dec. 2003.

[12] M. Malarvel, G. Sethumadhavan, P. C. R. Bhagi, S. Kar, and S. Thangavel, "An improved version of Otsu's method for segmentation of weld defects on X-radiography images," *Optik*, vol. 142, pp. 109–118, Aug. 2017.

[13] T. Matić, I. Aleksi, Ž. Hocenski, and D. Kraus, "Real-time biscuit tile image segmentation method based on edge detection," *ISA Trans.*, vol. 76, pp. 246–254, May 2018.

[14] D. Radi, M. E. A. Abo-Elsoud, and F. Khalifa, "Accurate segmentation of weld defects with horizontal shapes," *NDT & E Int.*, vol. 126, Mar. 2022, Art. no. 102599, doi: 10.1016/j.ndteint.2021.102599.

[15] T. Wei, D. Cao, C. Zheng, and Q. Yang, "A simulation-based few samples learning method for surface defect segmentation," *Neurocomputing*, vol. 412, pp. 461–476, Oct. 2020.

[16] L. Jiang, Y. Wang, Z. Tang, Y. Miao, and S. Chen, "Casting defect detection in X-ray images using convolutional neural networks and attention-guided data augmentation," *Measurement*, vol. 170, Jan. 2021, Art. no. 108736, doi: 10.1016/j.measurement.2020.108736.

[17] H. Würschinger, M. Mühlbauer, M. Winter, M. Engelbrecht, and N. Hanenkamp, "Implementation and potentials of a machine vision system in a series production using deep learning and low-cost hardware," *Proc. CIRP*, vol. 90, pp. 611–616, Jan. 2020.

[18] Y. Wang, C. Hu, K. Chen, and Z. Yin, "Self-attention guided model for defect detection of aluminium alloy casting on X-ray image," *Comput. Electr. Eng.*, vol. 88, Dec. 2020, Art. no. 106821.

[19] K. Chen, Z. Zeng, and J. Yang, "A deep region-based pyramid neural network for automatic detection and multi-classification of various surface defects of aluminum alloys," *J. Building Eng.*, vol. 43, Nov. 2021, Art. no. 102523.

[20] E. Westphal and H. Seitz, "A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks," *Additive Manuf.*, vol. 41, May 2021, Art. no. 101965.

[21] K.-J. Wang, H. Fan-Jiang, and Y.-X. Lee, "A multiple-stage defect detection model by convolutional neural network," *Comput. Ind. Eng.*, vol. 168, Jun. 2022, Art. no. 108096.

[22] D. He, K. Xu, and P. Zhou, "Defect detection of hot rolled steels with a new object detection framework called classification priority network," *Comput. Ind. Eng.*, vol. 128, pp. 290–297, Feb. 2019.

[23] K. Zhang, W. Wang, Z. Lv, Y. Fan, and Y. Song, "Computer vision detection of foreign objects in coal processing using attention CNN," *Eng. Appl. Artif. Intell.*, vol. 102, 2021, Art. no. 104242.

[24] Z. Pan, J. Yang, X.-E. Wang, F. Wang, I. Azim, and C. Wang, "Image-based surface scratch detection on architectural glass panels using deep learning approach," *Construct. Building Mater.*, vol. 282, May 2021, Art. no. 122717.

[25] L. Pan, R. Rogulin, and S. Kondrashev, "Artificial neural network for defect detection in CT images of wood," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106312.

[26] A. Ç. Seçkin and M. Seçkin, "Detection of fabric defects with intertwined frame vector feature extraction," *Alexandria Eng. J.*, vol. 61, no. 4, pp. 2887–2898, Apr. 2022.

[27] F. G. Y. Çiklaçandir, S. Utku, and H. Özdemir, "Fabric defect classification using combination of deep learning and machine learning," *J. Artif. Intell. Data Sci.*, vol. 1, no. 1, pp. 22–27, 2021.

[28] H. Uzen, M. Turkoglu, and D. Hanbay, "Texture defect classification with multiple pooling and filter ensemble based on deep neural network," *Expert Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114838.

[29] Z. Zhan, J. Zhou, and B. Xu, "Fabric defect classification using prototypical network of few-shot learning algorithm," *Comput. Ind.*, vol. 138, Jun. 2022, Art. no. 103628.

[30] Y. M. Fouda, "Integral images-based approach for fabric defect detection," *Opt. Laser Technol.*, vol. 147, Mar. 2022, Art. no. 107608.

[31] R. Mordia and A. K. Verma, "Visual techniques for defects detection in steel products: A comparative study," *Eng. Failure Anal.*, vol. 134, Apr. 2022, Art. no. 106047.

[32] A.-A. Tulbure, A.-A. Tulbure, and E.-H. Dulf, "A review on modern defect detection models using DCNNs—Deep convolutional neural networks," *J. Adv. Res.*, vol. 35, pp. 33–48, Jan. 2022.

[33] B. Shi and Z. Chen, "A layer-wise multi-defect detection system for powder bed monitoring: Lighting strategy for imaging, adaptive segmentation and classification," *Mater. Des.*, vol. 210, Nov. 2021, Art. no. 110035.

[34] Y. Xu, D. Li, Q. Xie, Q. Wu, and J. Wang, "Automatic defect detection and segmentation of tunnel surface using modified mask R-CNN," *Measurement*, vol. 178, Jun. 2021, Art. no. 109316.

[35] C. Xiao and J.-Y. Buffiere, "Neural network segmentation methods for fatigue crack images obtained with X-ray tomography," *Eng. Fract. Mech.*, vol. 252, Jul. 2021, Art. no. 107823, doi: 10.1016/j.engfracmech.2021.107823.

[36] W. Du, H. Shen, G. Zhang, X. Yao, and J. Fu, "Interactive defect segmentation in X-ray images based on deep learning," *Expert Syst. Appl.*, vol. 198, Jul. 2022, Art. no. 116692, doi: 10.1016/j.eswa.2022.116692.

[37] H. Xu, Z. H. Yan, B. W. Ji, P. F. Huang, J. P. Cheng, and X. D. Wu, "Defect detection in welding radiographic images based on semantic segmentation methods," *Measurement*, vol. 188, Jan. 2022, Art. no. 110569, doi: 10.1016/j.measurement.2021.110569.

[38] L. Yang, J. Fan, B. Huo, E. Li, and Y. Liu, "A nondestructive automatic defect detection method with pixelwise segmentation," *Knowl.-Based Syst.*, vol. 242, Apr. 2022, Art. no. 108338, doi: 10.1016/j.knosys.2022.108338.

[39] R. Xu, R. Hao, and B. Huang, "Efficient surface defect detection using self-supervised learning strategy and segmentation network," *Adv. Eng. Informat.*, vol. 52, Apr. 2022, Art. no. 101566, doi: 10.1016/j.aei.2022.101566.

[40] N. Kheradmandi and V. Mehranfar, "A critical review and comparative study on image segmentation-based techniques for pavement crack detection," *Construct. Building Mater.*, vol. 321, Feb. 2022, Art. no. 126162, doi: 10.1016/j.conbuildmat.2021.126162.

[41] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[42] S. Mittal, C. Chopra, A. Trivedi, and P. Chanak, "Defect segmentation in surfaces using deep learning," in *Proc. Int. Conf. Issues Challenges Intell. Comput. Techn. (ICICT)*, Sep. 2019, pp. 1–6.

[43] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2017.

[44] S. Kim, W. Kim, Y.-K. Noh, and F. C. Park, "Transfer learning for automated optical inspection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2517–2524.

[45] A. Zisserman and K. Simonyan, "Very deep convolutional networks for large-scale image recognition," in *Proc. CVPR*, 2014, pp. 1–14.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "ImageNet large scale visual recognition challenge," in *Proc. CVPR*, 2015, pp. 1–14.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. CVPR*, 2016, pp. 630–645.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[50] 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. (CVPR)*, Jun. 2016, pp. 2818–2826.

[51] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. CVPR*, 2016, pp. 1–7.

[52] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[53] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[54] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. CVPR*, 2019, pp. 6105–6114.

[55] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[56] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[57] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-Net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, May 2018.

[58] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 833–851.

[59] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[60] *Keras Applications*. Accessed: Jan. 8, 2022. [Online]. Available: https://keras.io/applications/

[61] P. Gainsford, "Mil-STD-636, military standard-visual inspection standards for small arms ammunition through caliber.50," Comput. Softw. Manual, U.S. Government Printing Office, Washington, DC, USA, Tech. Rep., 1958.

[62] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-Net," in *Proc. CVPR*, 2017, pp. 1–5.

[63] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[64] T. Wan, L. Zhao, H. Feng, D. Li, C. Tong, and Z. Qin, "Robust nuclei segmentation in histopathology using ASPPU-net and boundary refinement," *Neurocomputing*, vol. 408, pp. 144–156, Sep. 2020.

[65] M. Alom, M. Hasan, C. Yakopcic, T. Taha, and V. Asari, "Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation," in *Proc. CVPR*, 2018, pp. 1–12.

[66] S. Lal, D. Das, K. Alabhya, A. Kanfade, A. Kumar, and J. Kini, "NucleiSegNet: Robust deep learning architecture for the nuclei segmentation of liver cancer histopathology images," *Comput. Biol. Med.*, vol. 128, Jan. 2021, Art. no. 104075.

[67] X. Zhou and G. Yang, "Normalization in training U-Net for 2D biomedical semantic segmentation," *Proc. CVPR*, 2019, pp. 1–8.

[68] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist*, 2010, pp. 249–256.

SERHAT TURAL received the B.S. degree in computer engineering from Dokuz Eylul University, Izmir, Turkey, in 2007, and the M.S. degree in computer engineering from Ankara University, Ankara, Turkey, in 2011, where he is currently pursuing the Ph.D. degree in computer engineering. His research interests include 3D visualization, signal processing, machine vision, surface inspection, segmentation, deep learning, and geographic information systems.

REFIK SAMET (Member, IEEE) received the Ph.D. degree from the Department of Fault-Tolerant Multicomputer and Multiprocessor Systems, Russian Academy of Sciences, Institute of Control Sciences. He is currently a Professor with the Computer Engineering Department, Ankara University. He has published several papers in international journals and conferences. His research interests include parallel systems, fault-tolerant systems, reliability, cyber security, computer forensics, image processing, machine learning networks, and mobile applications.

SEMRA AYDIN received the B.S. degree in computer systems education from Gazi University, Turkey, the M.S. degree from the Graduate School of Natural and Applied Sciences, and the Ph.D. degree in computer science from Gazi University. She is currently an Assistant Professor with the Department of Computer Engineering, National Defense University, Ankara, Turkey. Her research interests include parallel computing, image processing, multicore processing, and GPU programming.

MOHAMED TRAORE received the B.S. degree in computer engineering from Istanbul Sabahattin Zaim University, Istanbul, Turkey, in 2021. He is currently pursuing the M.S. degree in computer engineering at Ankara University, Ankara, Turkey. His research interests include computer vision, deep learning, and big data.

• • •