Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-01803-8

ORIGINAL RESEARCH

Localization and segmentation of metal cracks using deep learning

Yasir Aslam[1] · N. Santhi[1] · N. Ramasamy[2] · K. Ramar[3]

Received: 14 October 2019 / Accepted: 18 February 2020

Abstract

Detection and quantification of defects from metal and metal-coated surfaces is one of the main challenges in computer vision-based semantic segmentation. Manual inspection is not a practically feasible solution for identifying defects such as surface-level cracks, scratches, and manufacturing mistakes, especially when we have to deal with large number of test subjects. The metal defects can be of different size, shape, and texture and often show close resemblance to the possible artefacts due to normal wear and tear and brush markings of metal coating. Hence it will be quite challenging to come up with an efficient segmentation method for quantifying such defects. In this work, we propose an automatic segmentation and quantification approach for inspecting defects from digital images of titanium-coated metal surfaces with a customized Deep learning architecture: UNet. The scheme uses a supervised learning approach with convolutional neural network (CNN) and can learn the suitable representations and features from the training data without any handcrafted features or human intervention. The proposed image segmentation method also uses appropriate pre-processing and post-processing stages. The input images are filtered using median filter for eliminating possible impulse noises, and the output mask generated from the CNN model is post-processed using suitable morphological operations for eliminating false detections. The detection and segmentation performance is evaluated using standard benchmarks, and the overall Dice score of the proposed model is 91.67% with a precision of 93.46%.

Keywords Segmentation · Convolutional neural networks · UNet · Morphological operations

1 Introduction

In last few decades, the computer-aided solutions are preferred over manual operations to resolve many day-to-day problems, including industrial applications. Technology advancements in both hardware and software directions provide enough room for extensive research in automated systems using artificial intelligence (AI) (Korbicz et al. 2012). Image processing (Schalkoff 1989; Li and Du 2017), computer vision (Chen 2015; Forsyth and Ponce 2002), and machine learning (Duygulu et al. 2002) are the main research domains under AI and scientists are working on building machines which think like human brain using visual information. Usually, this decision making can be of either classification or segmentation. Classification refers to the categorization of an image into a known tag, while segmentation usually refers to the identification and extraction of intended regions from an image or video frames.

Industry uses the advantages of computer vision from long back to automate many applications in product developments, testing's, and error corrections. These computer-aided helps to reduce the human interventions and to increase the performance in lesser cost and time. In industry, a lot of efforts are made for detecting defects and to maintain the product quality. Surface defect detection is one of the critical concerns in such quality control, especially when dealing with precise parts for building complex machines. Surface defects in the production of metal or metal-coated components may happen due to various reasons. Accidental scratches, manufacturing design error, improper metal coating, surface cracks, etc. are some of the common reasons for leaving surface defects in metal plates. These defects

✉ Yasir Aslam
yasiaslam@gmail.com

[1] Department of Electronics and Communication Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, India

[2] Department of Mechanical Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, India

[3] R&D, Muthayammal Engineering College, Rasipuram, India

should be identified and corrected promptly for maintaining a better quality in the whole system or machine where the metal piece need to be used. Manual detection may not be a suitable way to detect such defects since the work environment may not be ideal for people to work with. Computer vision-based surface defect detection approaches are popularly used in industries due to several advantages. Better work efficiency, faster detection, ability to work in extreme conditions, long work schedule, lower maintenance cost, etc. are the main positives of using computer-aided detection over manual procedure (Ghahramani 2015).

Metal coated components and structures are widely used in the manufacture of medical equipments, optical devices, precision cutting tools, moulding cans, aesthetic gears, etc. Most of such fields need very high precision and should be very ideal to the structure design. Surface defects may often characterize vital information such as the quality of the metal coating method, and superiority of the raw materials used for both coating and base. Hence, the timely detections are necessary in such cases for opting suitable solutions and for getting feedback about manufacturing process. In this paper, we are targeted to come up with an accurate computer vision-based surface defect detection on titanium Nitride-coated stainless-steel metal surfaces. Such a problem can be solved in two different approaches. One is an unsupervised method to segment the defected region using unsupervised techniques (Zhang et al. 2008) in image processing such as thresholding (Deng and Manjunath 2001; Niyas et al. 2016) or clustering (Chuang et al. 2006). Such methods do not need a labelled data for training the model and is appropriate for simple decision making. The second one is by using a supervised learning (Khan 2014). Here, a labelled training dataset is required to learn relevant features from them, and such learning will be applied on unseen test data. Such systems are preferred in complex segmentation tasks where the defected regions show much variance, and while showing some intraclass similarity.

In computer vision systems with such traditional machine learning concepts, images are analysed to generate handcrafted features (Nixon and Aguado 2012; Mohan and Poobal 2018) required for the required decision making. These features get trained on suitable classifiers to create an appropriate training model, and this model will be used in the test phase for making the required decision. Even though such algorithms can work automatically, designing relevant features for characterizing surface defects may need expertise in image processing domain. For selecting such features, an expert should study the possible characteristics of the defective region and how distinct it is comparing with the typical metal surface. The success in such feature selection is subjective to the expertise of the person, number of data samples that used for the background study, etc. In an article (Yiyang 2014), proposes a thresholding-based surface crack detection algorithm. (Lee 2011) presented a color image processing-based rust detection from steel plates with Eigen value analysis. Different machine learning approaches (Sankarasrinivasan et al. 2015; Adhikari et al. 2014) have also been proposed in this area.

Currently, deep learning approaches (Noh et al. 2015; Garcia-Garcia et al. 2018) are the state-of-the-art segmentation methods primarily when we are provided with sufficient number of data. Deep learning uses convolutional neural network layers to learn features automatically from the given training data (LeCun et al. 2015). Since the multilevel features are adaptively learned from the training data, it might be better than any of the possible handcrafted features. We have high resolution scanned images of target metal plates with a lot of noises such as brush markings. Our objective is to develop an efficient segmentation model without reducing the pixel resolution of the data, and at the same time to test in an acceptable time frame. Basic thresholding and clustering algorithms do not help here as the defects can present in different color and textured patterns. Selection of handcrafted features is not a suitable solution to deal with a data with much diversity inside the segmentation classes. Hence, we used a deep learning-based segmentation model as the backbone of our surface defect detection system (Li et al. 2019).

The rest of the paper is organized as explained below. A thorough overview of the available literatures on surface defect detections are discussed in Sect. 2. A detailed description of the proposed model is presented in Sect. 3. Section 4 dealt with the discussion on experimental setups and results. Subsequently, the final part concludes the paper based on its performance.

2 Literature review

Several automatic surface crack detection systems are proposed in this filed so far. Most of them focuses on machine learning concepts and are preferable for versatile data. In the article (Ahamad and Rao 2016), Rao et al. proposes a segmentation system which is used to detect defects from ceramic tile surfaces. The approach uses many image processing stages such as median filter based denoising (Wang and Lin 1997; Brownrigg 1984), thresholding, morphological operations, etc. The paper also discusses about a fuzzy-based (Naidu et al. 2018) classification after segmentation to categorize the test tiles into different grades based on the extent of defects on it. The method is relatively simple and efficient on relatively uniform and smooth surfaces such as ceramic tiles. However, the approach may not produce the expected result when the test surface is affected by noises and with different textures on regular and defected regions. Moreover, the used algorithms consume much time while using high-resolution test image samples.

Another image analysis-based crack measurement system is proposed in the article (Lins and Givigi 2016) by Givigi et al. The article explains a machine vision approach which detect cracks and abnormal textures with a color model context. Particle filtering is the core algorithm used behind the model and it also aims to quantization the extent of cracks with the segmented results. Results show high segmentation errors (~ 8%) and limits the application to use in environments where accuracy need to be very high.

In the article (Aslam et al. 2019) explains a Cuckoo search algorithm (Gandomi et al. 2013) based adaptive thresholding for segmentation of exterior cracks in metal-coated surfaces. The Cuckoo Optimization Algorithm (COA) is customized here for better thresholding with an adaptive search scheme. The segmentation system uses appropriate stages of data pre-processing for data normalization and post processing of segmentation mask with Level set (Cremers et al. 2007) and morphological operations.

However, the approach has some drawbacks. The COA is too costly in terms of time and it will create unacceptable delays while using relatively high-resolution images. The reported results can also be improved to make use in precise applications. In another article (Aslam et al. 2018), the same author proposes another segmentation tool for surface crack detection using Particle Swarm Optimization (PSO) algorithm. It also fails to make use of the entire resolution of the available data for better segmentation. Another similar algorithm is stated in the article by (Kalaiarasi et al. 2019). The algorithm targets an automated segmentation scheme for detecting holes and cracks from metal pieces using edge detection algorithms. Even though the algorithm provides excellent results in detecting primary cracks and defects, it is not reliable in the cases where the test subjects are susceptible to noises. Many articles were proposed with adaptive thresholding (Otsu 1979; Xue and Titterington 2011).

While analyzing the literatures, we could find many drawbacks of the existing systems such as lack of accuracy while dealing with high-resolution images, need for finding suitable feature space, inability to detect different classes of abnormalities, high time cost, misdetections over boundary regions, etc. In the proposed segmentation approach, we tried to address all of these issues by making use of a customized deep learning architecture (Kumar and Garg 2019). The model uses CNN layers to learn the appropriate features from a train data and classify each pixel of the test images with the help of this trained model. The final detection mask will have two sets of pixels. One set belongs to the background (normal metal surface), and the other class is for the defected region. A more detailed explanation of the methodology and experimental setup is explained in the subsequent sections.

3 Methodology

The overall surface defect segmentation system can be subdivided into three main modules: a pre-processing, CNN based segmentation and post-processing and are showed in Fig. 1. Each of these modules are explained here.

3.1 Pre-processing

The images that we used here are color images with a very high resolution (~ 9000×9000). Some data samples and corresponding ground truths are shown in Fig. 2 Since the high-resolution images are not good choices on deep CNN models, we split the images into favourable image sizes.

This image splitting will help to fasten the learning process as well as avoids the data loss due to image resizing.

The image resizing has to be done in all image samples in train and test data. Some images may affect by impulse noise due to scanning artefacts. In such case we can also use a median filter for denoising.

3.2 Defect detection and segmentation using deep CNN

The segmentation is achieved using a fully convolutional neural network model (Long et al. 2015) based on the U-Net architecture (Ronneberger et al. 2015).

The main advantages of the architecture are: (1) it won't have any fully-connected neural network layers in this architecture, and (2) create a same size output mask as that of the input image. As a result, each pixel in the output segmentation mask represents whether the pixel in the corresponding position of the input image belongs to the normal or defected one.

The U-Net architecture uses an encoder-decoder approach to make the output segmentation mask with the required resolution. The CNN architecture used in the proposed model is shown in Fig. 3. The detailed description of the network architecture is given in the next subsection.

3.2.1 Network architecture

In the proposed model, the CNN input layer is designed to accept the image size of $128 \times 128 \times 3$. In the encoding path, each level consists of two convolution layers (with size 3×3) followed by a ReLU activation layer (Nair and Hinton 2010). Batch normalization is applied once at each level, and the down sampling of feature space is made through a Maxpooling layer (Scherer et al. 2010) with size 2×2.

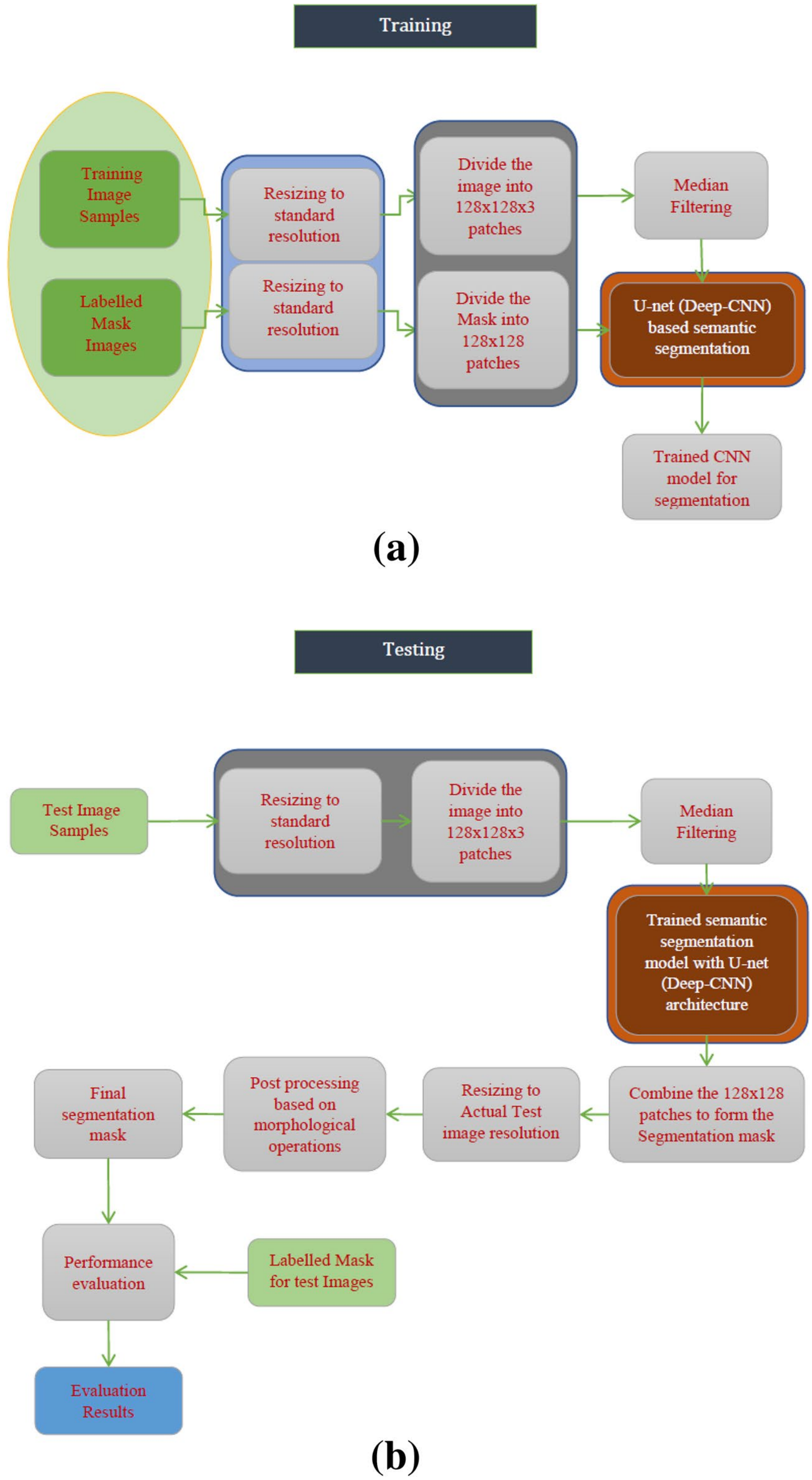
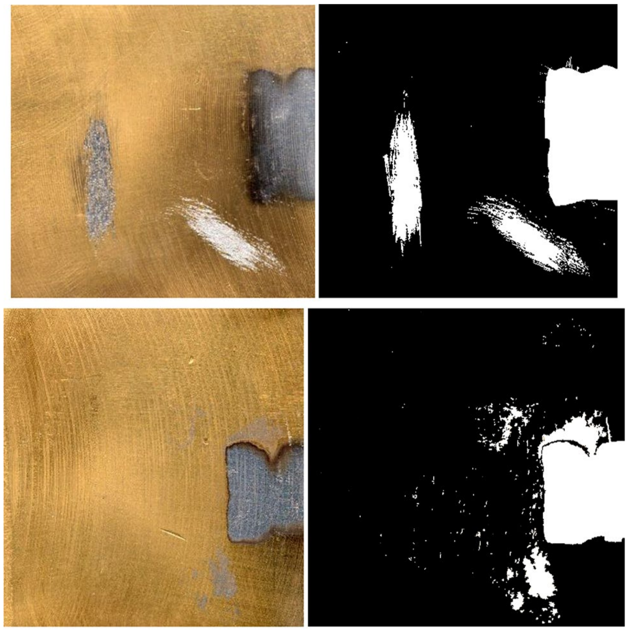


Fig. 1 Block diagram of proposed segmentation system. **a** Training, and **b** Testing

The model makes use of such four levels in the encoding path and hence is a 4-depth U-Net model. The number of filter kernels used in each depth is mentioned in Fig. 3. In the decoding path, transposed convolution is used for up sampling the feature space. After up sampling the feature space from the encoding part is concatenated with the feature planes in the decoding part and is referred as skip connections. These skip connections are provided to get finer details from the encoder while increasing the feature plane resolutions in the expanding (or decoding) path. Finally, a



(a) **(b)**

Fig. 2 **a** Scanned images of metal plates, and **b** the corresponding ground truth

1×1convolution with a sigmoid activation function at the end layer of the decoding network followed by a thresholding is performed to make a single plane segmentation mask. In the proposed model we used a depth of 4. The training is designed to use a batch size of 32 to get a fair trade between learning rate and time.

We used a combination of binary cross-entropy (BCE) and the dice loss (DL) (Janocha and Czarnecki 2017) as the loss function to progress training. The loss function is computed as per the given formula.

$$\text{Loss\_function} = \text{BCE} + \text{DC} \tag{1}$$

The model also uses Adam optimizer with a learning rate of 0.001 which we found empirically over trying different combinations. The convolution operations are processed with required zero padding to keep the output same as that of the input to the convolution layer. Since there was enough training data (more than 25,000 training images), the model did not use any augmentations while training the model.

3.2.2 Post-processing

The training is designed to receive input images with a size of $128 \times 128 \times 3$. Hence testing can also do using the exactly same resolution. In order to satisfy this, the test images are split into 128×128 color image tiles and each tile will get tested one after one. After getting the segmentation mask of every tiles in a single test image, they will stitch together appropriately using concatenation. As a result, the final segmentation mask of the test image holds same resolution as that of the test image and will be considered as the segmentation mask from the proposed CNN model. However, there may be small noisy subjects in the segmented mask and we used morphological post-processing to eliminate such false objects from the segmentation mask. We used a morphological opening (with structural element size of 10×10) followed by a area opening by eliminating all connected components whose area is less than 50 pixels. The very high resolution of the images guides to select such high structural elements.

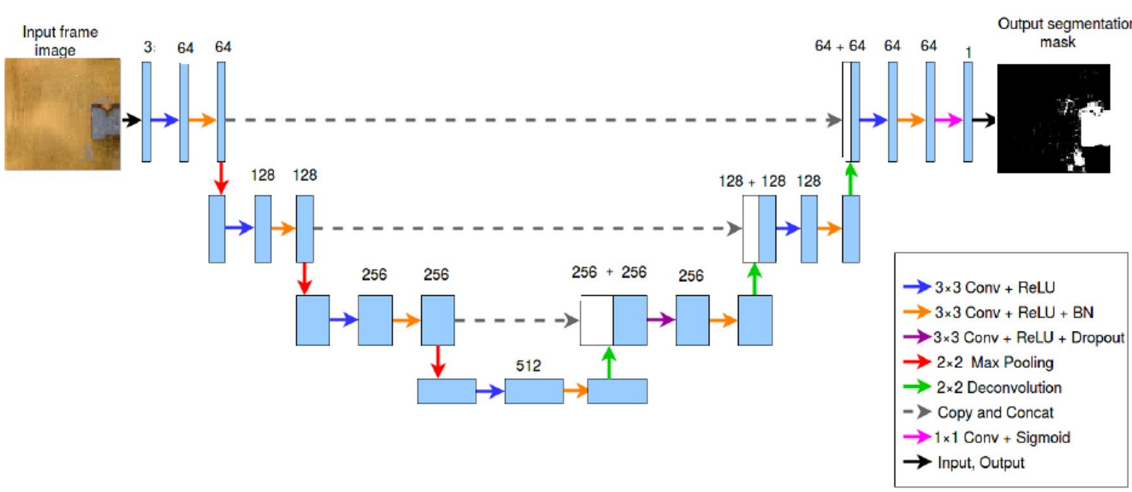


Fig. 3 Proposed CNN model for surface defect detection

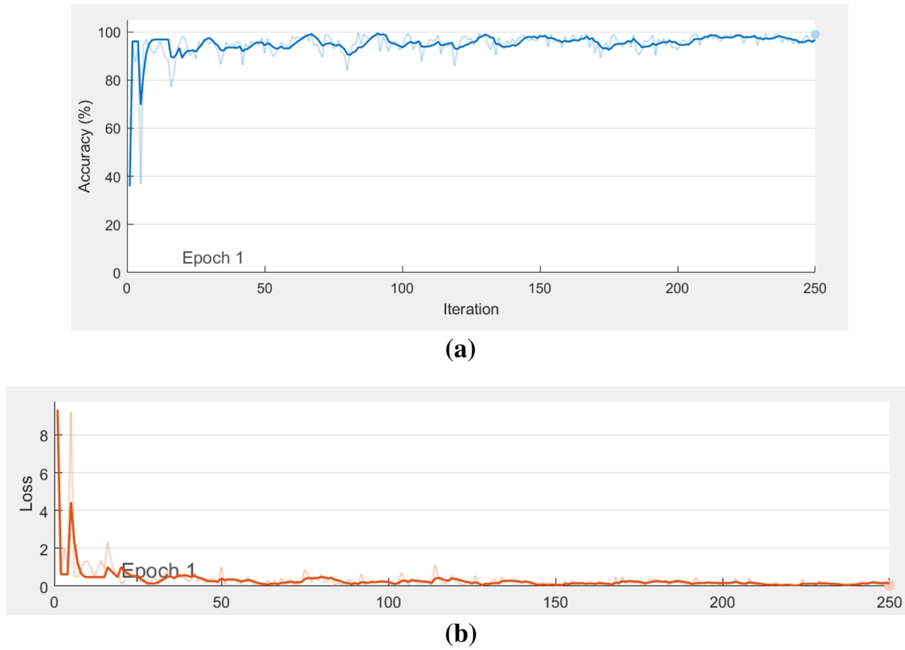


Fig. 4 Training progress curves. **a** Accuracy vs. iterations, and **b** loss value vs. iterations

4 Experimental setup

The whole experiments are carried out by Titanium Nitride (TiN) coated stainless steel plates (with a grade of 303). Defected regions are made using random scratching with different size, shape, and textures. The scratches and markings are made randomly without any biasing or adjustments. There were digital images of 20 such plates with a resolution of ~ 9000×9000. The implementations were carried out in MATLAB 2019a on a workstation with windows 10 OS, Intel Core i7 -7700 HQ CPU @ 2.80 GHz, solid-state hard drive, 8 GB RAM and with a 4 GB GPU card NVIDIA GTX 1050. We did both quantitative and qualitative analysis on the observed results to validate and compare with state-of-the-art approaches.

4.1 Training

For training, 80% of the images from the whole dataset are randomly chosen as train data and the remaining for testing and evaluation. During training, each single train image (of resolution ~ 9000×9000) is divided into 1600 patches and resized into 128×128 resolution. Since we used a global dataset with 20 images, train data holds 16 images with high resolution and there will be a total of 25,600 (16*1600) color patches will be available for training with a resolution of 128×128. The training converged to less than 99% training accuracy and near to 0.1 training loss in 3 epochs and in 2400 iterations. The training accuracy and loss curves are shown in Fig. 4. Training took nearly four hours to complete 2400 iterations (800 iterations per epoch) of the training process. The trained model will be saved and is used in the further testing process.

The observations made on the number of training features with respect to the change in depth and filter shapes during different training experiments are reported in Tables 1, 2.

Table 1 Number of trainable parameters with different depths at a convolution size of 3×3

Depth	Number of convolution layers	Number of filters in the first layer	Total number of parameters
1	6	64	4,04,869
2	10	64	1.8 Million
3	14	64	7.6 Million
4	18	64	31 Million

Table 2 Number of trainable parameters with different number of filters

Depth	Number of filters in the first layer	Total number of parameters
4	8	4,85,957
4	16	3.2 Million
4	32	8.9 Million
4	64	31 Million

5 Results and discussions

The segmentation results observed for detecting surface-level defects are evaluated qualitatively and quantitatively. Some of the sample observations are shown in Fig. 5. Quantitative evaluation is performed with state-of-the-art segmentation benchmarks such as Precision, Recall, Dice coefficient, Accuracy, Specificity, and Miss Rate (Zhang 1996). Since the images available for experiments are less in number, we used a five-fold cross validation to avoid any kind of biasing in the results.

These metrics can be explained based on the True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). TP is the number of correctly predicted pixels from the positive (here it refers the regions with defects) areas. TN is the correctly identified background or negative regions. FP counts the number of pixels which are wrongly classified as Positives and FN counts the number of pixels which erroneously classified as Negatives. These annotations are made by the pixel level comparison of Predicted Mask and the actual Ground Truth.

(i). Recall or sensitivity

Recall refers to the overall detection rate of the target area. So, it computes the ratio of detected positive pixels with total positives present. Mathematically it can be represented as:

$$\mathrm{Recall} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{2}$$

(ii). Precision

Precision refers to the number of correct objects among the detected ones. It can be computed using

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{3}$$

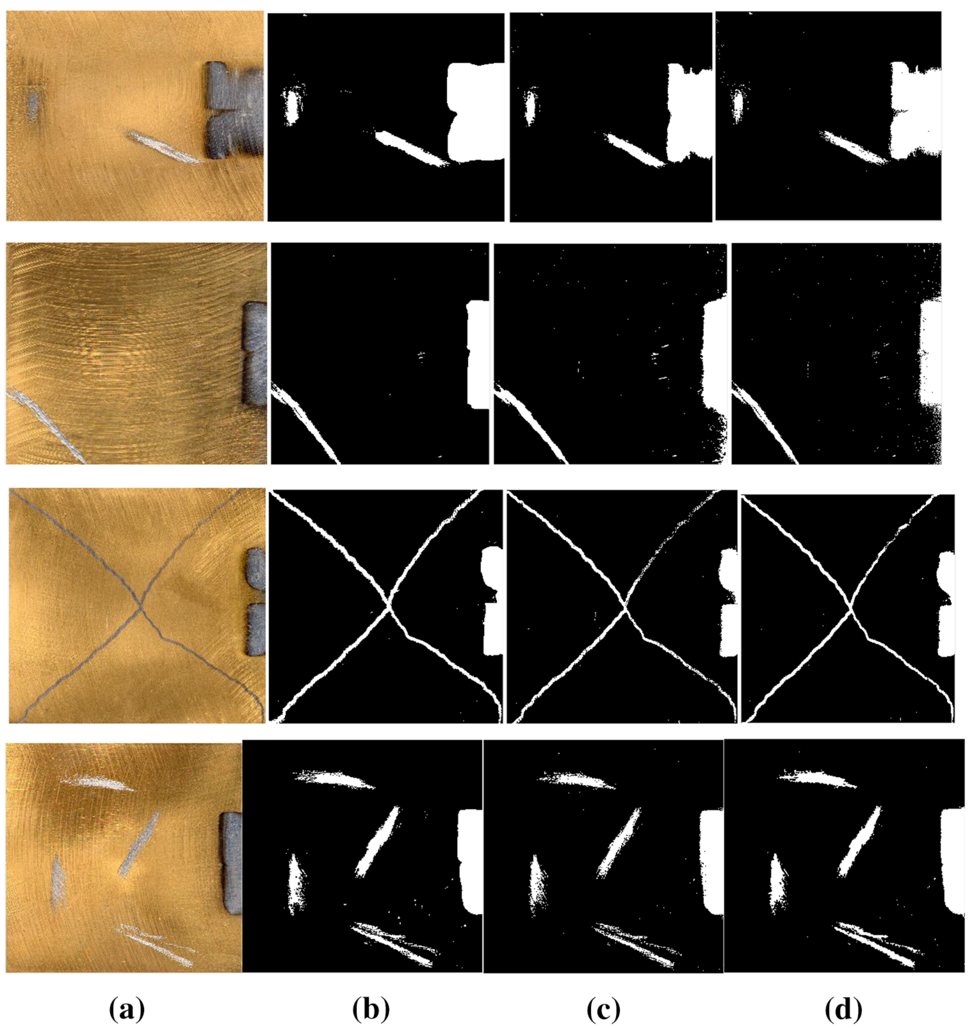


Fig. 5 Results **a** input image, **b** ground truth, **c** prediction using CNN, and **d** predicted result after post-processing

Table 3 Segmentation results

Image index	Precision	Recall	Dice score	Accuracy	Specificity	Miss rate
1	0.9944	0.8448	0.9135	0.9789	0.9994	0.1552
2	0.9167	0.9770	0.9459	0.9812	0.9907	0.0230
3	0.8346	0.9857	0.9040	0.9871	0.9927	0.0141
4	0.9926	0.8290	0.9035	0.9858	0.9997	0.1710
Average results	0.9346	0.9092	0.9167	0.9833	0.9920	0.091

Table 4 Comparison of average segmentation results with state-of-the-art approaches ()

Paper reference	Algorithms used	Precision	Recall	Dice score	Accuracy	Specificity	Miss rate	Computation time (in seconds)
Otsu (1979)	Otsu	0.7588	0.7145	0.7226	0.7258	0.7004	0.2855	3.44
Xue and Titterington (2011)	Median based Otsu	0.7523	0.7387	0.7448	0.7335	0.7118	0.2552	15.8
Aslam et al (2018)	Adaptive thresholding using PSO	0.8331	0.8451	0.8335	0.8906	0.816	0.155	14.7
Aslam et al (2019)	Adaptive thresholding using COA	0.8977	0.8013	0.8471	0.8336	0.8223	0.1987	13.24
Proposed method	Deep CNN	0.9346	0.9092	0.9167	0.9833	0.9920	0.091	47.35

(iii). Dice score

In a segmentation problem (as well as in classification problem), an ideal system will have both precision and recall values very close to 1. Dice score computes the harmonic mean of precision and recall and holds high values only when both the precision and recall values are high.

$$\text{Dice Score} = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall}) \tag{4}$$

(iv). Accuracy

This term denotes the overall correct detection by considering both positive and negative regions. Accuracy is calculated by

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{5}$$

(v). Specificity

It denotes the effectivity in identifying the background (Negative) regions. So, it is calculated as

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP}) \tag{6}$$

(vi). Miss rate

It counts the missed positive elements during the detection process. So, it took the ratio of false negatives to the total positives.

$$\text{Miss Rate} = \text{FN}/(\text{TP} + \text{FN}) \tag{7}$$

The test results in different test images are tabulated in Table 3. The evaluated results are compared against state-of-the-art metal crack segmentation algorithms and are given in Table 4.

From the results it is clear that the proposed segmentation approach could be able to beat the existing methods with significant lead in performance. For comparison we have used the results reported in their papers.

6 Conclusion

The article proposes decent computer vision-based surface defect detection from metal or metal-coated surfaces. The approach uses a Deep CNN based segmentation architecture customized to learn the required features automatically. The overall segmentation model uses different pre and post-processing techniques to optimize the algorithm performance to make it useful in a practical scenario. The system precisely segments out the defected regions and the false detections are very less even the images contain lot of distortions due to brush markings. The system possesses excellent results over state-of-the-art segmentation systems in the relevant domain. Both the quantitative and qualitative analysis states the algorithm performance in different types of data. Another advantage of the proposed model is that the model can be easily converted into similar segmentation problems by slight modifications in the algorithm. Future research is planned in the localization and segmentation of surface defects in all metal surfaces and classification of defects using deep learning based instance segmentation.

References

Adhikari RS, Moselhi O, Bagchi A (2014) Image-based retrieval of concrete crack properties for bridge inspection. Autom Constr 39:180–194

Ahamad NS, Rao JB (2016) Analysis and detection of surface defects in ceramic tile using image processing techniques. Microelectronics Electromagnetics and Telecommunications. Springer, New Delhi, pp 575–582

Aslam Y, Santhi N, Ramasamy N, Ramar K (2018) An effective surface defect detection method using adaptive thresholding fused with PSO algorithm. Int J Simul Syst Sci Technol 19(6):1–7

Aslam Y, Santhi N, Ramasamy N, Ramar K (2019) A modified adaptive thresholding method using cuckoo search algorithm for detecting surface defects. Int J Adv Comput Sci Appl 10(5):214–220

Brownrigg DR (1984) The weighted median filter. Commun ACM 27(8):807–818

Chen CH (2015) Handbook of pattern recognition and computer vision. World Scientific, Singapore, pp 1–483

Chuang KS, Tzeng HL, Chen S, Wu J, Chen TJ (2006) Fuzzy c-means clustering with spatial information for image segmentation. Comput Med Imaging Graph 30(1):9–15

Cremers D, Rousson M, Deriche R (2007) A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. Int J Comput Vis 72(2):195–215

Deng Y, Manjunath BS (2001) Unsupervised segmentation of color-texture regions in images and video. IEEE Trans Pattern Anal Mach Intell 23(8):800–810

Duygulu P, Barnard K, de Freitas JF, Forsyth DA (2002) Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In: European conference on computer vision, pp. 97–112

Forsyth DA, Ponce J (2002) Computer vision: a modern approach. Prentice Hall, Upper Saddle River, pp 1–720

Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35

Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, Garcia-Rodriguez J (2018) A survey on deep learning techniques for image and video semantic segmentation. Appl Soft Comput 70:41–65

Ghahramani Z (2015) Probabilistic machine learning and artificial intelligence. Nature 521(7553):452–459

Janocha K, Czarnecki WM (2017) On loss functions for deep neural networks in classification, pp. 1–10

Kalaiarasi M, Indolia Y, Seema N, Kumar B (2019) Metal crack detection using image processing. Int Res J Eng Technol (IRJET) 6(6):930–934

Khan MW (2014) A survey: image segmentation techniques. Int J Futur Comput Commun 3(2):89–93

Korbicz J, Koscielny JM, Kowalczuk Z, Cholewa W (2012) Fault diagnosis: models, artificial intelligence, applications. Springer, Berlin, pp 1–919

Kumar A, Garg G (2019) Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. J Ambient Intell Hum Comput, pp.1–16

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–521

Lee S (2011) Color image-based defect detection method and steel bridge coating. Texas 47th ASC Annual International Conference Proceedings, Texas, pp. 1–7

Li D, Du Y (2017) Artificial intelligence with uncertainty. CRC Press, Boca Raton, pp 1–55

Li X, Hu G, Pan Z (2019) An eigendecomposition method based on deep learning and probabilistic graph model. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-019-01555-0

Lins RG, Givigi SN (2016) Automatic crack detection and measurement based on image analysis. IEEE Trans Instrum Meas 65(3):583–590

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440

Mohan A, Poobal S (2018) Crack detection using image processing: a critical review and analysis. Alexandria Eng J 57(2):787–798

Naidu MSR, Kumar PR, Chiranjeevi K (2018) Shannon and fuzzy entropy based evolutionary image thresholding for image segmentation. Alexandria Eng J 57(3):1643–1655

Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th international conference on machine learning, pp. 807–814

Nixon M, Aguado AS (2012) Feature extraction and image processing for computer vision. Academic Press, Cambridge, pp 1–601

Niyas S, Reshma P, Thampi SM (2016) A color image segmentation scheme for extracting foreground from images with unconstrained lighting conditions. The International Symposium on Intelligent Systems Technologies and Applications, pp. 3–19

Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. Proceedings of the IEEE international conference on computer vision, pp. 1520–1528

Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, pp. 234–241

Sankarasrinivasan S, Balasubramanian E, Karthik K, Chandrasekar U, Gupta R (2015) Health monitoring of civil structures with integrated UAV and image processing system. Proced Comput Sci 54:508–515

Schalkoff RJ (1989) Digital image processing and computer vision. Wiley, New York, pp 1–489

Scherer D, Müller A, Behnke S (2010) Evaluation of pooling operations in convolutional architectures for object recognition. In: International conference on artificial neural networks, pp. 92–101

Wang JH, Lin LD (1997) Improved median filter using minmax algorithm for image processing. Electron Lett 33(16):1362–1363

Xue JH, Titterington DM (2011) Median-based image thresholding. Image Vis Comput 29(9):631–637

Yiyang Z (2014) The design of glass crack detection system based on image preprocessing technology. IEEE Joint International Information Technology and Artificial Intelligence Conference, pp. 39–42

Zhang YJ (1996) A survey on evaluation methods for image segmentation. Pattern Recogn 29(8):1335–1346

Zhang H, Fritts JE, Goldman SA (2008) Image segmentation evaluation: a survey of unsupervised methods. Comput Vis Image Underst 110(2):260–280