

TLU-Net: A Deep Learning Approach for Automatic Steel Surface Defect Detection

Praveen Damacharla

Research Scientist
KineticAI Inc.

Crown Point, IN, USA
Praveen@KineticAI.com

Achuth Rao M. V.

Dept. of Electrical Engineering
Indian Institute of Science (IISc)
Bengaluru, KA, India
achuthr@iisc.ac.in

Jordan Ringenberg

Computer Science Dept.
The University of Findlay
Findlay, OH, USA
ringenberg@findlay.edu

Ahmad Y. Javaid

EECS Department
The University of Toledo
Toledo, OH, USA
Ahmad.Javaid@Utoledo.edu

Abstract—Visual steel surface defect detection is an essential step in steel sheet manufacturing. Several machine learning-based automated visual inspection (AVI) methods have been studied in recent years. However, most steel manufacturing industries still use manual visual inspection due to training time and inaccuracies involved with AVI methods. Automatic steel defect detection methods could be useful in less expensive and faster quality control and feedback. But preparing the annotated training data for segmentation and classification could be a costly process. In this work, we propose to use the Transfer Learning-based U-Net (TLU-Net) framework for steel surface defect detection. We use a U-Net architecture as the base and explore two kinds of encoders: ResNet and DenseNet. We compare these nets' performance using random initialization and the pre-trained networks trained using the ImageNet data set. The experiments are performed using Severstal data. The results demonstrate that the transfer learning performs 5% (absolute) better than that of the random initialization in defect classification. We found that the transfer learning performs 26% (relative) better than that of the random initialization in defect segmentation. We also found the gain of transfer learning increases as the training data decreases, and the convergence rate with transfer learning is better than that of the random initialization.

Index Terms—Automated visual inspection (AVI), DenseNet, ResNet, Surface defect detection, Transfer learning, U-Net

I. INTRODUCTION

Steel is one of humanity's most important building materials. Defect inspection is a critical step of quality control in the steel plates. This step mainly involves capturing images of the steel surface using an industrial camera followed by recognizing, localizing, and classifying the defect, which helps rectify the defect's cause. Typically, this process is performed manually, which is not reliable and time-consuming. Unreliable quality control can cause a huge economic problem for manufacturers. Manual detection can be replaced or aided by the automatic classification using computer vision methods. The general flow of automatic visual inspection for quality control is shown in Fig. 1. There are two main steps involved in the defect inspection. The first step is to classify the defect type from the images, and the second step is to identify the defect location in the image. There are various automatic methods in the literature to address one/both of these steps. Some of the early methods use a handcrafted feature to classify the defect type [1, 2, 3], and few methods find the coarse defect locations. The main drawback of these methods is that

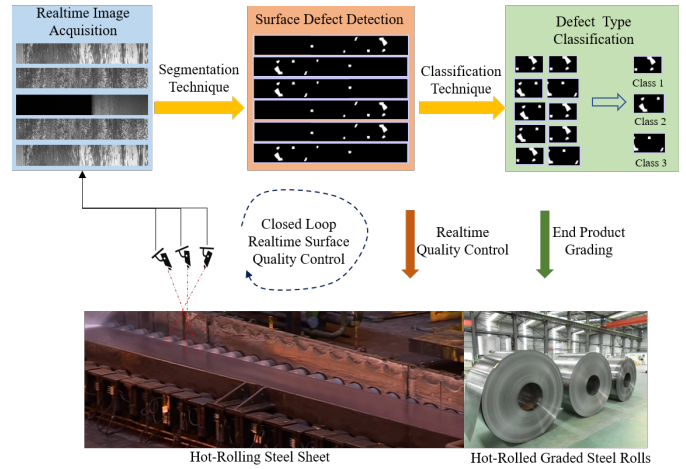


Figure 1. A Generic Automatic Visual Inspection Outline

the features need to be designed by an experts. The designed feature may not generalize to new type of defect. The recent advances in end-to-end deep learning (DL) methods overcame these hand-designed features. It learns to extract the multi-scale features depending on the task using only the data and labels. The DL method has been shown to outperform the hand-designed features in various computer vision tasks [4].

There are various deep learning methods are used to perform defect classification. [5] use the features extracted from the OverFeat, a variant of convolutional neural networks (CNN), to do the defect classification. They have also shown that the fixed features from the pre-trained network perform well on some defects and perform poorly on texture kind of defect. These authors have also proposed a structural visual inspection method based on faster region-based CNN (faster R-CNN) to ensure quasi-real-time simultaneous detection of multiple types of defects [6]. [7] proposed to detect weak scratches using deep CNN and skeleton extraction. They have shown that their method is robust to background noise. [8] proposed a variant of you only look once (YOLO) network to detect surface defects of flat steel in real-time. The CNN is used extensively for a different kind of defect classification on different data sets [9, 10]. [11] use a transfer learning ap-

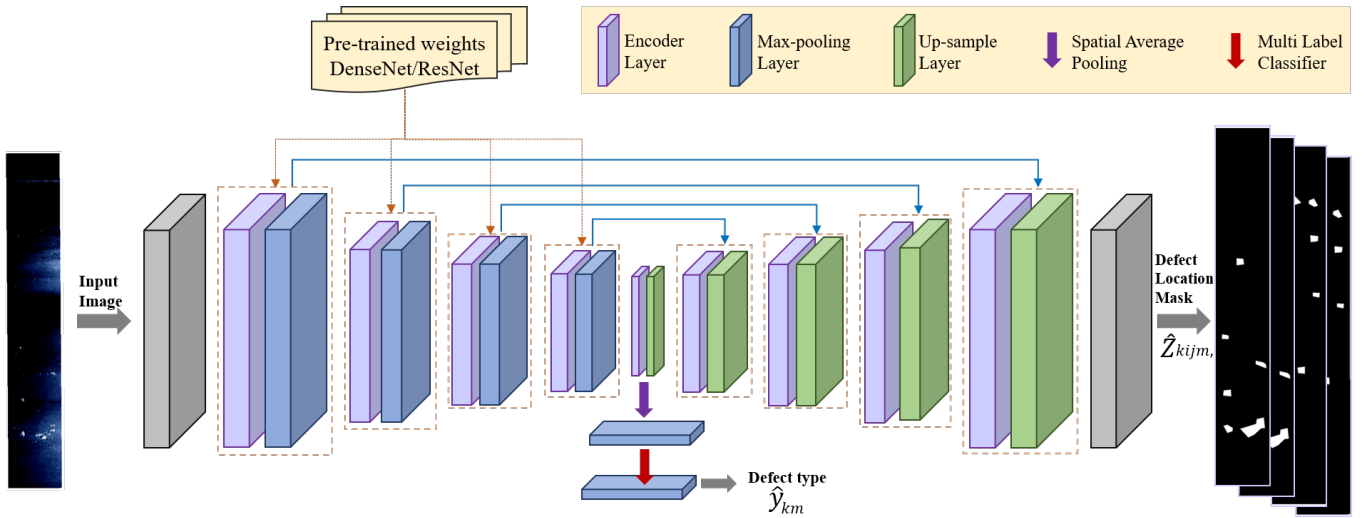


Figure 2. Proposed architecture transfer learning method for the joint steel defect classification and segmentation. The blue line indicate the skip connection and the orange dotted line indicate the initialization.

proach for defect classification. They have shown that transfer learning can help achieve a good classification accuracy with fewer data samples. [12] use a patch-wise classification to do both defect classification and segmentation.

There are various methods in the literature on defect localization. [13] uses a classical CNN to perform steel defect detection. Authors have explored the effect of regularization and unsupervised pre-training. [14] uses a pre-trained ResNet to extract the multi-scale features, and the features from different scales are fused using a multilevel feature fusion network (MFN). The fused features and region proposal network are used to classify the defect type and predict the bounding box. The main drawback of the method is that localization is very coarse. [15] use an U-Net and residual U-Net architecture for the fine segmentation of the steel defect. The method's main drawback is that the networks are trained with random initialization and need a large amount of pixel-level annotation of the defects. The pixel-level annotation process can be very time consuming and expensive. [16] uses a SegNet based semantic segmentation for the steel defect detection. There are various unsupervised and reinforcement learning based methods for steel defect detection. The summary of various methods for defect classification, and segmentation can be found in [17]. The authors discuss the taxonomy of defect detection including the statistical, spectral, model based and machine learning methods.

In this work, we systematically study the transfer learning effectiveness for steel defect classification and localization (SDCL). The transfer learning or domain adaption aims to reuse the feature learned in one domain to improve the learning in the other domain. This is a popular approach in cases where the annotated data is limited. The transfer learning has shown really good application in various tasks such a object detection [18], semantic segmentation [19, 20] etc. It is already shown that the transfer-learning from an arbitrary domain to another

domain may not be useful. Transfer learning is most effective when two domains are similar [20, 21]. Hence, it is important to study the effectiveness of transfer learning in the case of SDCL. We consider a baseline architecture of U-Net for steel defect segmentation. U-Net has demonstrated state of the art performance in various image segmentation tasks [22]. It uses an encoder-decoder architecture with skip connections. The encoder learns the images' features at different scales, and the decoder uses these features to predict the segmentation masks. In this work, we explore two kinds of pre-trained encoder networks– ResNet and DenseNet networks. Both of these networks have been shown to perform well on various computer vision tasks. The networks are pre-trained on the ImageNet data set [23]. We use a linear classifier using the bottleneck representation of U-Net to classify the defect. We fine-tune both the encoder and decoder of the network using the Severstal dataset [24]. The experiments on Severstal data shows that the performance of both segmentation and classification is better in case of the pre-trained network compared to the random initialization. It is found that performance gain by using the pre-trained networks is even higher if 50% of data is used for training. We also show that the convergence of transfer learning is faster compared to random initialization.

II. PROPOSED TRANSFER LEARNING BASED U-NET

The proposed architecture for joint steel defect segmentation and classification is shown in Fig. 2. The architecture takes an input image of dimension $H \times W$ and classifies each pixel to be one or more type of the defect. It involves mainly four parts (1) The U-Net architecture, (2) the type of initialization (3) classification and (4) objective function

A. U-net architecture

The U-Net is an encoder-decoder architecture with a skip connection. The encoder encodes the image using an encoder block and reduces the resolution using pooling. This helps in

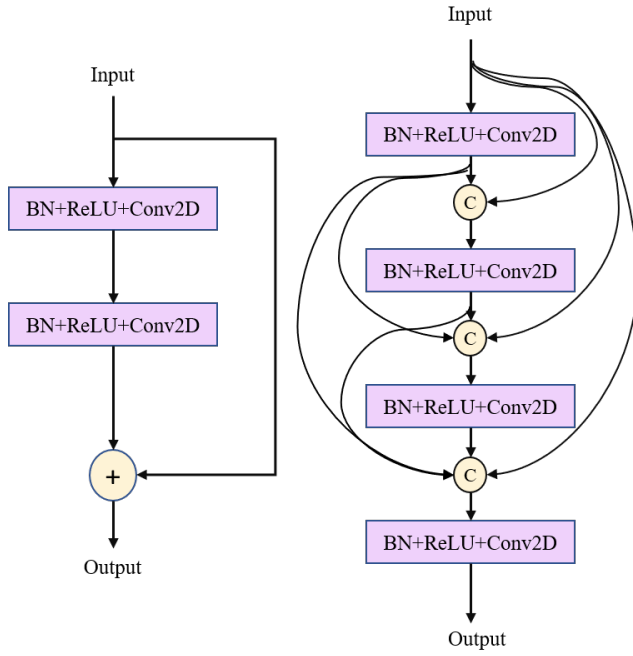


Figure 3. The structure of encoder layer Resnet (left) and the Densenet (right). The concatenation of inputs is indicated by (c) and + indicate the add operation. BN+ReLU+Conv2D indicate the batch normalization, Relu activation and convolution of kernel size 3x3

extracting a multi-scale feature of images. The decoder up-samples the representation in every step. The skip connection can enable the decoder to select the feature at a different scale to make a more accurate prediction of the object boundaries. The output of the U-Net is $256 \times 1600 \times N$ with sigmoid activation, where N is the types of steel defects.

B. Transfer learning

We explore two kinds of encoder blocks for transfer learning. Both of these nets are trained using ImageNet data set [23]. We briefly review the features of the two networks in the following subsections.

1) *ResNet*: The residual networks (ResNet) are very deep convolutions neural networks with skip connection [25]. The vanishing gradient problem is addressed by having a skip connection after each block. Each block contains a two 3x3 convolution with batch normalization and ReLU activation. Fig 3 (left) shows one layer of ResNet. The total parameters of the encoder are 11 million.

2) *Densenet-121*: Densely connected convolutions neural nets (DenseNets) are stacked convolution networks where the feature map of the L -th layer is concatenated with the feature maps of the previous layer [26]. This has been shown to alleviate the vanishing gradient problem. The network's representation power is also increased because the deep layer has access to the previous layer feature maps. Fig 3 (right) shows one layer of DenseNet. The total parameters of this encoder are 6 million.

C. Classification

The encoder output encodes the rich abstract representation of the input image. Hence we propose to spatial average pooling the encoder output to extract the image representation. The image representation is passed through a linear classifier with sigmoid activation to enable the multi-label classification.

D. Objective function

The joint segmentation and classification problem is formulated as a weighted combination of the two losses, as shown below.

$$\mathcal{J} = \sum_{k=1}^L \sum_{m=1}^N \left[BCE(\hat{y}_{km}, y_{km}) + \sum_{j=1}^{256} \sum_{i=1}^{1600} BCE(z_{kijm}, \hat{z}_{kijm}) \right] \quad (1)$$

where BCE indicate the binary cross-entropy loss, k indicate the data point index, m is the defect class index, i, j are the spatial index, \hat{y} indicate the predicted probability and y is the ground truth defect labels. The predicted and ground truth pixel labels are indicated by z and \hat{z} . During the test stage, the labels from the probability are obtained using the threshold of 0.5.

III. EXPERIMENTS AND RESULTS

A. Data-set and Pre-processing

The Kaggle Competition - "Severstal: Steel Defect Detection" data is used for all the experiments. In each experiment, the input image could contain one or more kinds of defects. The training set includes 12568 images, and 6666 of them include at least one defective region. Ground truth classification was performed by an expert to provide the defect type classification and the annotation of the defective region by visual inspection. The resolution of images is 256x1600 px. We normalize the image using the global mean and standard deviation. We apply a random vertical/horizontal flip as data augmentation. The same augmentation applies to both the original image and the corresponding ground truth masks to pair with the augmented images.

B. Experimental setup

We use a total of 75% data for the training, 12.5 % data for validation, and 12.5% data for the testing. The U-Net of five encoders and decoder is used for all the experiments. The network is trained using the objective function in eq. 1 with the batch size 16 and Adam optimizer with learning rate of 5×10^{-4} , $\beta_1=0.99$ and $\beta_2=0.99$ [27]. We train the network for 10 epochs with early stopping. We use the U-Net with random initialization as the baseline. We have implemented the network in PyTorch [28] with the PyTorch segmentation library [29]. The ResNet/DenseNet with random initialization is indicated by ResNet(Random)/DenseNet(Random). The Imagenet pre-trained counterparts are indicated by ResNet(Imagenet)/DenseNet(Imagenet) respectively. To understand the model's sample complexity, we also train these networks using 50% of the training data. We refer to the pre-trained initialization as TLU-Net and the random initialization as just U-net.

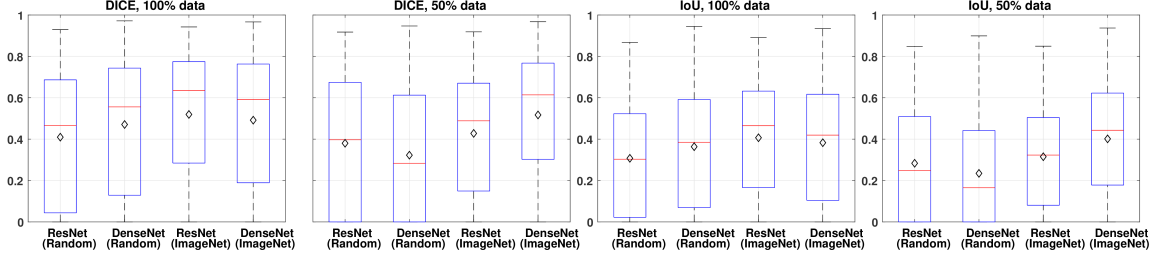


Figure 4. Boxplot comparison of DICE and IoU for different networks and the initialization. The red line indicate the median, the black dot indicate the mean, the blue box indicate the 75% confidence interval and the black whiskers indicate the 95% confidence intervals.

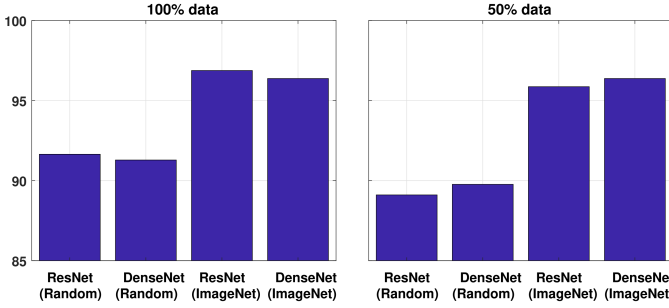


Figure 5. Comparison of average MLA for different networks and initialization using different amount of training data.

C. Evaluation metrics

We evaluate the steel defect classification performance using **multi-label classification accuracy (MLA)** and the **average area under receiver operating curve (AUC)** across 4 classes. The MLA is defined as the proportion of the predicted correct labels to a total number of labels for instance. We treat the multi-label classification as 4 separate binary classifiers and compute the average AUC across four classifiers.

We have used DICE and the intersection of unions (IoU) to evaluate the performance of steel defect segmentation. The DICE metric for each class is defined as follows:

$$DICE = \frac{2|X \cap Y|}{|X| + |Y|} \quad (2)$$

The IoU is defined as follows:

$$IoU = \frac{|X \cap Y|}{|X \cup Y|} \quad (3)$$

where X and Y are the ground-truth and the predicted segmentation masks, \cup indicates the union operation, \cap indicated the intersection, and $|\cdot|$ indicates the cardinality.

D. Results and discussion

Fig. 5 shows the MLA(%) comparison for different networks and the initialization. It is clear from the figure that the TLU-Net achieves 5% (absolute) improvement in the MLA compared to the random initialization with 100% training data. This indicates that the features learned using ImageNet can help for steel defect classification as well. The MLA gap

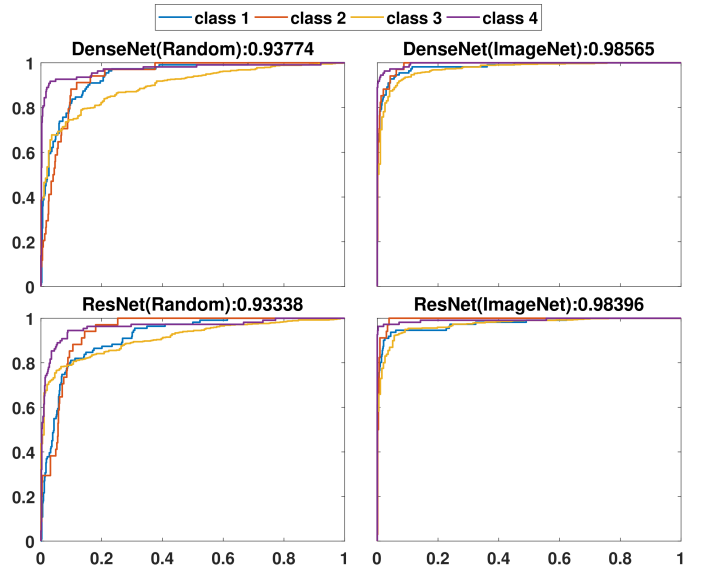


Figure 6. comparison of AUC plot of four classes for different networks and initialization. The title shows the average AUC across four classes.

increases to 8% (absolute) as the training data is reduced to 50%. The performance of TLU-Net does not drop significantly as the training data is reduced. This indicates that the TLU-Net is helpful when there is a limited number of annotated data points. The ResNet(ImageNet) and DenseNet(ImageNet) performs best in case of 100% and 50% of training data respectively. This could be because the DenseNet has fewer parameters than the ResNet and hence needs less data but has less representational power.

Fig. 6 shows the AUC for different networks and initialization. It is clear from the figure that the best AUC archived for class 1 in all cases. Class 3 and class 4 defects demonstrate the poorest performance. This is mainly because of the number of samples of the training data for each class. The AUC of all classes improved by using the TLU-Net compared to U-Net. The DenseNet(ImageNet) has the highest AUC.

Fig. 4 shows the box plot of the DICE and IoU for different network with the 100% and 50% of the training data. In all cases, the median and mean performance of the TLU-Net perform better than that of the U-Net. In case of 100% training

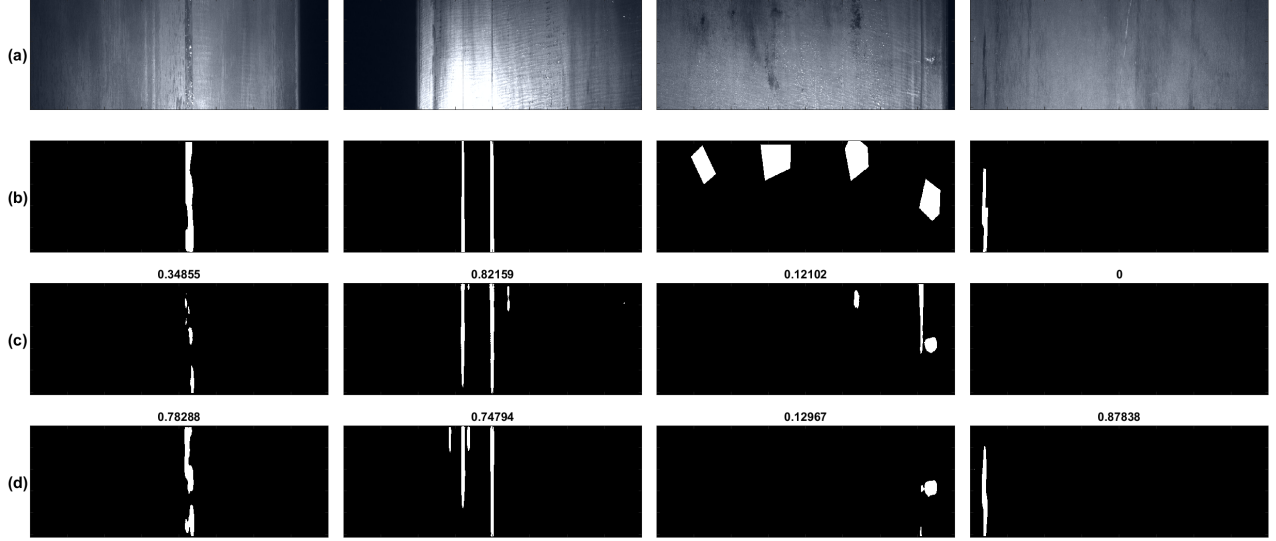


Figure 7. Illustration of segmentation mask prediction. (row a) The input images (row b) The ground truth masks (row c) The mask predicted by ResNet(Random) (row d) the mask predicted by ResNet(ImageNet). The corresponding DICE for the prediction is shown in the title of the image.

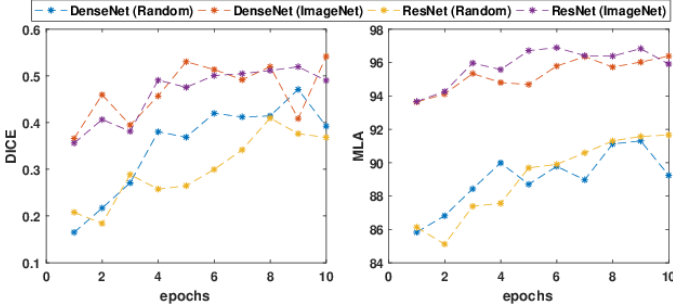


Figure 8. Comparison of validation loss/MLA evolution over epoch for different networks and initialization.

data, the ResNet(ImageNet) DICE/IoU is better than all the other models and it has the small 75% confidence interval. The TLU-net with ResNet shows an improvement of $\sim 26\%$ (relative) compared to U-Net with ResNet. The TLU-net with DenseNet shows an improvement of $\sim 5\%$ (relative) compared to U-Net with ResNet. But in the case of 50% training data, the DenseNet with transfer learning shows an improvement of 60%(relative), and the ResNet shows an improvement of 12%(relative). This clearly indicate the gain of using transfer learning is higher as the number of annotated samples are lower.

Fig. 8 shows the DICE/MLA metric using validation data during the course of training. This helps us understand the convergence rate of different networks. It is clear from the figure that the TLU-Net has higher DICE at the beginning of the epochs than the U-Net. The converged DICE value for the TLU-Net is higher compared to the U-Net. This clearly indicates that transfer learning is helping in faster

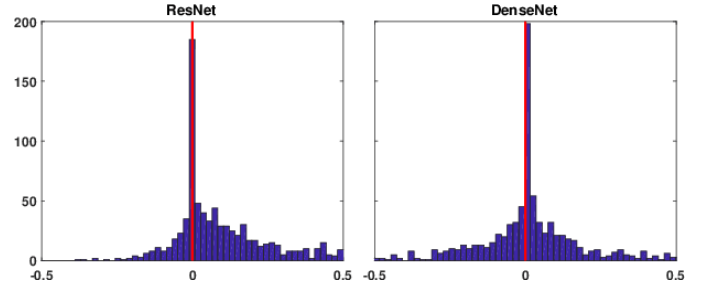


Figure 9. Histogram of difference between DICE obtained using transfer learning and the random initialization. The red line indicate the zero line.

convergence of the model. Similar observations are applicable for the MLA plot also. It is interesting to note that the pre-trained model's starting accuracy is significantly higher than the random initialization. This is mainly because the classifier directly uses the encoder output for classification, and the encoder is initialized with the pre-trained weights. It implies that the pre-trained features are useful in discriminating against the different steel defects.

Fig. 7 shows the four illustrative examples of mask predictions for TLU-Net using ResNet and the U-Net using ResNet. In the first column, the TLU-Net can detect the defect more accurately, and the U-Net fails to detect some parts of the defects. In the second column, the TLU-net is showing some false positive detection because of some illumination differences. In the third column, both networks failed to detect some of the defects. This mainly because the training data has few defects belonging to this class. We hypothesize that the TLU-net also fails because the defect share is more complex, and pre-learned features may not help this scenario. In the

fourth column, the TLU-Net can detect the defect in the form of a fine line, and the U-Net is failing to detect these lines.

Fig. 9 shows the histogram of the DICE difference between the TLU-Net and the U-Net using both ResNet/DenseNet. It is clear from the figure that the histogram is skewed toward positive values. We observe improvement in the case of 81% images in case ResNet and 63% of images in DenseNet.

IV. CONCLUSION

In this work, we propose to use the transfer learning framework for steel defect classification and segmentation. We use a U-Net architecture as a base architecture and explore two kinds of encoders: ResNet and Dense Net. We compare these nets' performance using random initialization and the pre-trained networks trained using ImageNet data set. We found that the performance of the transfer learning is superior both in terms of defect segmentation and classification. We also found the performance gap increases as the training data decreases. We also found that the convergence rate with transfer learning is better than that of the random initialization. We have found that transfer learning performance is poor in rare defect types and complex shape defects. As a part of future work, we would like to work on transfer learning to handle more complex shapes using synthetic data and the rare defect type generalization using generative models. We want to explore the semi/weakly supervised learning approaches to reduce the annotated training data requirement.

REFERENCES

- [1] Praminda Caleb-Solly and Jim E Smith. "Adaptive surface inspection via interactive evolution". In: *Image and Vision Computing* 25.7 (2007), pp. 1058–1072.
- [2] Kechen Song and Yunhui Yan. "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects". In: *Applied Surface Science* 285 (2013), pp. 858–864.
- [3] Yongsheng Dong et al. "Texture classification and retrieval using shearlets and linear regression". In: *IEEE transactions on cybernetics* 45.3 (2014), pp. 358–369.
- [4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.
- [5] Pei-Hung Chen and Shen-Shyang Ho. "Is overfeat useful for image-based surface defect classification tasks?" In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 749–753.
- [6] Young-Jin Cha et al. "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types". In: *Computer-Aided Civil and Infrastructure Engineering* 33.9 (2018), pp. 731–747.
- [7] Limei Song et al. "Weak micro-scratch detection based on deep convolutional neural network". In: *IEEE Access* 7 (2019), pp. 27547–27554.
- [8] Jiangyun Li et al. "Real-time detection of steel strip surface defects based on improved yolo detection network". In: *IFAC-PapersOnLine* 51.21 (2018), pp. 76–81.
- [9] Shiyang Zhou et al. "Classification of surface defects on steel sheet using convolutional neural networks". In: *Mater. Technol* 51.1 (2017), pp. 123–131.
- [10] Li Yi, Guangyao Li, and Mingming Jiang. "An End-to-End Steel Strip Surface Defects Recognition System Based on Convolutional Neural Networks". In: *steel research international* 88.2 (2017), p. 1600068.
- [11] Vidhya Natarajan et al. "Convolutional networks for voting-based anomaly classification in metal surface inspection". In: *2017 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2017, pp. 986–991.
- [12] Ruoxu Ren, Terence Hung, and Kay Chen Tan. "A generic deep-learning-based approach for automated surface inspection". In: *IEEE transactions on cybernetics* 48.3 (2017), pp. 929–940.
- [13] Daniel Soukup and Reinhold Huber-Mörk. "Convolutional neural networks for steel surface defect detection from photometric stereo images". In: *International Symposium on Visual Computing*. Springer. 2014, pp. 668–677.
- [14] Yu He et al. "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features". In: *IEEE Transactions on Instrumentation and Measurement* 69.4 (2019), pp. 1493–1504.
- [15] Didarul Amin and Shamim Akhter. "Deep Learning-Based Defect Detection System in Steel Sheet Surfaces". In: *2020 IEEE Region 10 Symposium (TENSYP)*. IEEE. 2020, pp. 444–448.
- [16] Domen Tabernik et al. "Segmentation-based deep-learning approach for surface-defect detection". In: *Journal of Intelligent Manufacturing* 31.3 (2020), pp. 759–776.
- [17] Qiwu Luo et al. "Automated Visual Defect Detection for Flat Steel Surface: A Survey". In: *IEEE Transactions on Instrumentation and Measurement* 69.3 (2020), pp. 626–644.
- [18] Jonti Talukdar et al. "Transfer learning for object detection using state-of-the-art deep neural networks". In: *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2018, pp. 78–83.
- [19] Varun Belagali et al. "Two step convolutional neural network for automatic glottis localization and segmentation in stroboscopic videos". In: *Biomedical Optics Express* 11.8 (2020), pp. 4695–4713.
- [20] Ruoqi Sun et al. "Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4360–4369.
- [21] Shai Ben David et al. "Impossibility theorems for domain adaptation". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 129–136.
- [22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [23] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [24] "Severstal: Steel Defect Detection on Kaggle Challenge". In: ().
- [25] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [26] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.
- [27] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).
- [28] Adam Paszke et al. "Automatic differentiation in pytorch". In: (2017).
- [29] Pavel Yakubovskiy. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2020.