Deep Learning-Based Defect Detection System in Steel Sheet Surfaces

Didarul Amin
AISIP Lab, Dept. Of Computer Science and Engineering
International University Of Business Agriculture And Technology
Dhaka, Bangladesh
didarul38@gmail.com

Shamim Akhter
AISIP Lab, Dept. Of Computer Science and Engineering
International University Of Business Agriculture And Technology
Dhaka, Bangladesh
shamimakhter@gmail.com
https://orcid.org/0000-0003-1408-9133

Abstract— ***Steel is one of the most important building materials of modern times and the production process of flat sheet steel is complicated. Before shipping or delivering steel, sheets need to undergo a careful inspection procedure to avoid defects and thus localizing and classifying surface defects on a steel sheet is crucial. In this study, we advance the steel defect inspection methods by designing machine learning models that aim to detect multi-level defects from sample steel sheet images and classify them according to their corresponding classes. We explore two (2) deep learning methods including U-NET and Deep Residual U-NET to solve the steel defect detection problem with a Dice coefficient accuracy of 0.543 and .731 correspondingly.***

Keywords—Deep Learning, U-NET, Deep Residual U-NET, Defect Detection, Steel Surface Images.

I. Introduction

The defect detection system in steel sheet surfaces is playing a critical role in the steel sheets industry by detecting, localizing, recognizing and subsequently correcting causative factors. It is also quite necessary for controlling product quality and generating real-time analysis reports. Detecting process involves determining the existence of the steel surface defects from images taken from the industrial cameras. Localization locates all known content in the scene including the defect regions. Recognition takes the defect regions infers the defect category according to the defect appearance. A defect detection system in steel sheet surfaces is thus a combined process of detection, localization, and classification. Typically, in steel mills, human inspectors manually perform the defect detection process on steel sheets. However, this procedure is very time consuming, costly but lower efficient and does not meet up the requirement of real-time online defect detection. Many recent pieces of research are conducted on a combined approach of computer vision with machine learning methods to solve the requirements for real-time online defect detection on steel sheets. However, they apply some morphological operations on high-frequency images generated by low-cost industrial cameras and simple classifiers to solve the classification problem. Thus, these approaches lower accuracy and unable to handle complex problems including multi-level classifications or localize the defected area within a single image.

In this study, we advance the steel defect inspection methods by applying modern segmentation approach to partition the image into various regions and designing a new machine learning model to feed the region pixels to detect the defect region from a single sample image of steel sheet and classify them according to their corresponding multi-level classes. We apply U-NET and Residual U-NET to solve the given problem. Dice coefficient method is used to trace the accuracy of the selected machine learning models

II. Related Works

Many researchers introduced computer vision steel surface inspection systems. Caleb and Steuer in [1] used artificial neural networks (ANN) to detect defects in hot rolled steel strip. Pakkanen et al. in [2] applied edge histogram, color structure, and homogeneous texture as features extractor, and K-Nearest Neighbor as a classifier on hot-rolled steel strip surface to detect the defected images. Hongbin and Keesug et al. in [3] applied Support Vector Machine (SVM) as a classifier of the inspection system, SVM gives better performance than ANN for their samples on hot rolling steel. Smriti and Bhandari in [4] considered edge detection with a modified scheme based on heuristics used by human inspectors for identifying surface imperfections to compute the features then applied SVM as classifier for classifying surface images into two (2) classes defective and defect-free; the system was applied on surface texture database. Sharifzadeh et al. in [5] used image processing algorithms for detecting four popular classes of steel defects. Liu et al. in [6] used a relevance vector machine as a classifier to detect four kinds of defects on the steel surface. Luiz et al. in [7] adopted Principal Component Analysis as features extractor, and Self-Organizing Maps as a classifier to classify six classes of the hot-rolled steel surface defects. Song and Yan in [8] proposed a new method Adjacent Evaluation Completed Local Binary Patterns as feature extractor and employed SVM as a classifier on Northeastern University (NEU) hot-rolled steel strip surface defect database. Song et al. in [9] adopted a scattering convolution network as a feature extractor and employed SVM as a classifier on the NEU database. Wang et al. [10] proposed fault diagnosis based on a continuous sparse auto-encoder and illustrated the effectiveness of the presented approach by IEC TC 10 dataset of transformers faults. Mao et al. in [11] proposed an intelligent fault-diagnosis by the auto-encoder algorithm the effectiveness of the presented approach is verified by rolling

978-1-7281-7366-5/20/$31.00 ©2020 IEEE

element bearings data set. Lu et al. in [12] used stacked de-noising auto-encoder as fault diagnosis method and rotating machinery datasets were employed to demonstrate the effectiveness of the proposed method.

All the related works referred above consider classifier models only to recognize defect classes. Their analyses miss the detection of the defective area or pixel locations. CNN is largely used when the whole image is needed to be classified as a class label. But many tasks require classifying each pixel of the image. Fully connected CNN like U-NET or Residual U-NET can be used as a pixel-level classifier. We also want to approach a combined model for defect classification and defect localization using segmentation and modern fully CNNs. To complete the above goal, we aim to explore image segmentation using U-NET and Residual U-NET. They both are efficient to solve multi-label defects on a single image and localize the defect on images.

III. Dataset and Feature

Our dataset [13] consists of steel sheet images, corresponding defects classes, and defect regions. Steel sheet images are categories into two parts including training and testing. There are 12568 training images and 1801 testing images of $1600\times256\times1$ size each. Out of all the training images, 5902 images are with defects and 6666 images are without defects. The distribution of all training images according to the four (4) defects is presented in fig.1. Few defected sample images from our dataset are presented in fig.2

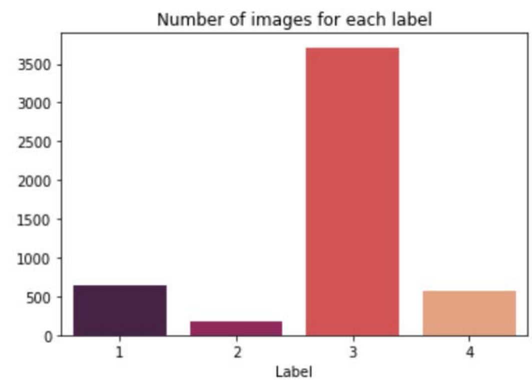
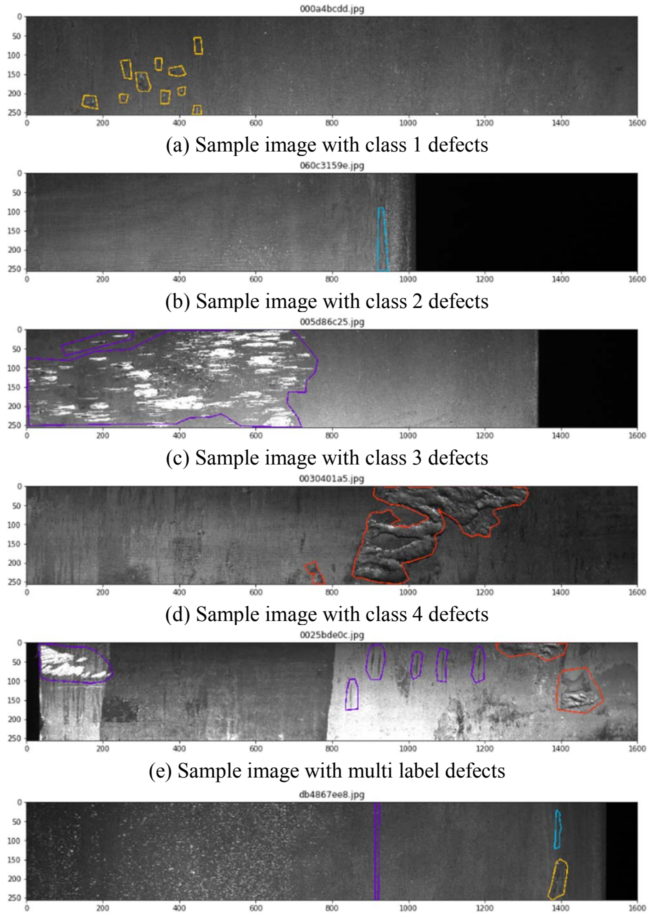


Figure 1: Number of images for each label.



(a) Sample image with class 1 defects

(b) Sample image with class 2 defects

(c) Sample image with class 3 defects

(d) Sample image with class 4 defects

(e) Sample image with multi label defects

(f) Sample image contain three type defects

Figure 2: Sample images with instance-level defects

IV. Methodology

A. U-NET

U-NET [14] architecture is presented in fig.3. It consists of a contracting path (the left side) and an expanding path (the right side). There are 4 down-sampling layers in the contracting path. Each down-sampling layer contains two 3x3 convolutional units, each followed by batch normalization and ReLU, and then a 2x2 max pooling. The contextual information from the contracting path is then transferred to the expanding path by skip connection. There are four (4) up-sampling layers in the expanding path. Each up-sampling layer contains a 2x2 transposed convolution, a concatenation with the corresponding feature maps from the encoding path, and two (2) 3x3 convolutional units, each followed by batch normalization and ReLU. Finally, there is a 1x1 convolutional layer and a SoftMax layer to map the feature vector at each pixel into five (5) different classes (0 for no defect, 1-4 for defects of different classes).

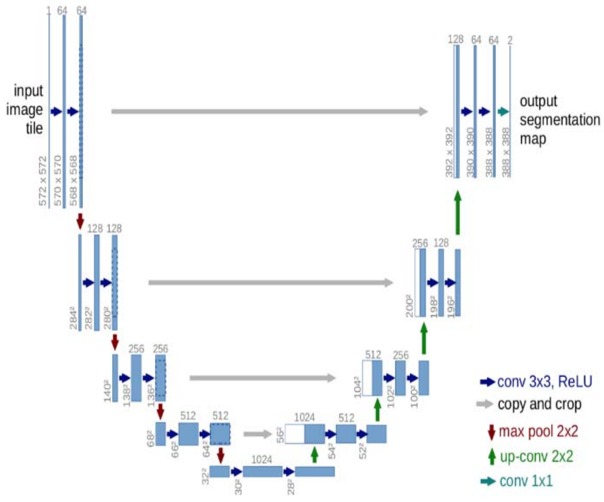


Figure 3: Architecture of the U-NET

B. *Deep Residual U-NET*

Deep Residual U-NET [15] is presented in fig. 4. It is a semantic segmentation NN which combines strengths of both U-NET and Res-Net networks. This combination brings us two benefits: 1) the residual unit makes ease training of the network; 2) skip connections within a residual unit and between low levels and high levels of the network will facilitate information propagation without degradation, making it possible to design a neural network with much fewer parameters however could achieve comparable ever better performance on semantic segmentation. In our proposed work we utilize a 7-level deep architecture and comprises of three parts: encoding, bridge and decoding. The first part encodes the input image into compact representations. The last part recovers the representations to a pixel-wise categorization, i.e. semantic segmentation. The middle part serves as a bridge connecting the encoding and decoding paths. All of the three parts are built with residual units which consist of two 3 × 3 convolution blocks and an identity mapping. Each convolution block includes a BN layer, a ReLU activation layer and a convolutional layer. The identity mapping connects the input and output of the unit. Encoding path has three (3) residual units. In each unit, instead of using pooling operation to downsample, the feature map size, a stride of 2 is applied to the first convolution block to reduce the feature map by half. Correspondingly, decoding path composes of three residual units. Before each unit, there is an up-sampling of feature maps from the lower level and concatenation with the feature maps from the corresponding encoding path. After the last level of decoding path, a 1 × 1 convolution and a sigmoid activation layer are used to project the multi-channel feature maps into the desired segmentation. In total, we have 15 convolutional layers comparing with 23 layers of U-NET. It is worth noting that the indispensable cropping in U-NET is unnecessary in our network. The parameters and output size of each step are presented in Table-I.

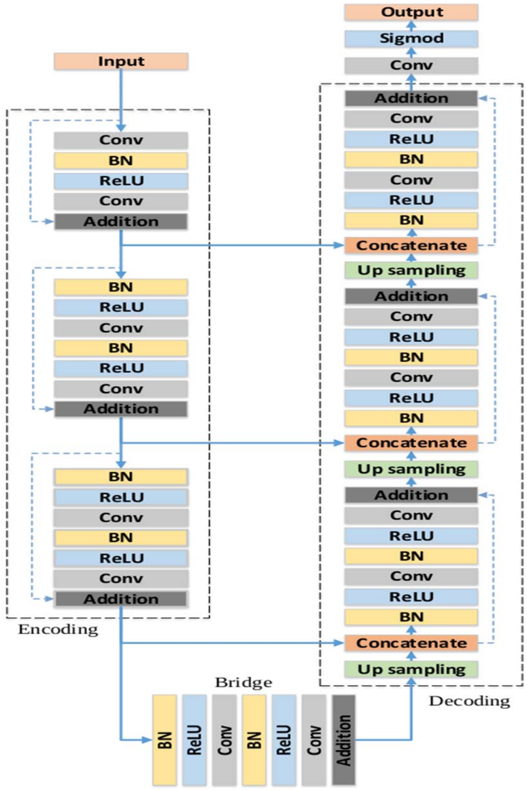


Figure 4: Architecture of the Deep Residual U-NET

TABLE I: THE NETWORK STRUCTURE OF RESUNET

	Unit level	Conv layer	Filter	Stride	Output size
Input					224 × 224 × 3
Encoding	Level 1	Conv 1	3 × 3/64	1	224 × 224 × 64
		Conv 2	3 × 3/64	1	224 × 224 × 64
	Level 2	Conv 3	3 × 3/128	2	112 × 112 × 128
		Conv 4	3 × 3/128	1	112 × 112 × 128
	Level 3	Conv 5	3 × 3/256	2	56 × 56 × 256
		Conv 6	3 × 3/256	1	56 × 56 × 256
Bridge	Level 4	Conv 7	3 × 3/512	2	28 × 28 × 512
		Conv 8	3 × 3/512	1	28 × 28 × 512
Decoding	Level 5	Conv 9	3 × 3/256	1	56 × 56 × 256
		Conv 10	3 × 3/256	1	56 × 56 × 256
	Level 6	Conv 11	3 × 3/128	1	112 × 112 × 128
		Conv 12	3 × 3/128	1	112 × 112 × 128
	Level 7	Conv 13	3 × 3/64	1	224 × 224 × 64
		Conv 14	3 × 3/64	1	224 × 224 × 64
Output		Conv 15	1 × 1	1	224 × 224 × 1

V. RESULT AND ANALYSIS

A. *Evaluation Metrics*

The Dice coefficient is used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The formula is given by:

$$\mathrm{Dice}(X, Y) = \frac{2*|X \cap Y|}{|x|+|Y|} \quad \dots (1)$$

Where X is the predicted set of pixels and Y is the ground truth. The Dice coefficient is defined to be 1 when both X and Y are empty. We used Dice loss function during training, which is defined as

$$L_{Dice} = 1 - \frac{1}{B}\sum_{i=1}^{B}\frac{2\sum_{j=1}^{n} y_j^{(i)}\hat{y}_j^{(i)}}{\sum_{j=1}^{n} y^{(i)} + \sum_{j=1}^{n}\hat{y}^{(i)}} \quad \dots (2)$$

Where B is the batch size, $n = 4 \times 256 \times 1600$ is the total number of pixels of the 4 defect classes, $y_{j(i)}$ is the true label of each pixel (1 for defect and 0 for normal), and $\hat{y}_{j(i)}$ is the predicted probability.

B. *Image Segmentation and Multi Labels Classification*

We apply instance segmentation method which uses the output of semantic segmentation as input and obtains instance-aware segmentation result. U-NET is configured with a batch size of 4 due to the limitation of GPU memory. Adam is used as the optimizer and Dice loss function during training. The loss values and the positive Dice score during training are shown in Fig. 5a and 5b, respectively. After 20 epochs, Dice scores results in 0.543. The U-NET model presents good performance for predicting defects of class 3 (Fig. 5c). However, it fails to predict many defects of class 1, 2 and 4 (Fig. 5d).

Deep Residual U-NET is configured with a batch size of 4 due to the limitation of GPU memory. We also use Adam as the optimizer. We use Dice loss function during training. The loss values and the positive Dice score during training are shown in Fig. 6a and 6b, respectively. After 20 epochs, Dice scores are 0.731. The deep residual U-NET model shows good performance for predicting defects of all classes (Fig. 6c).

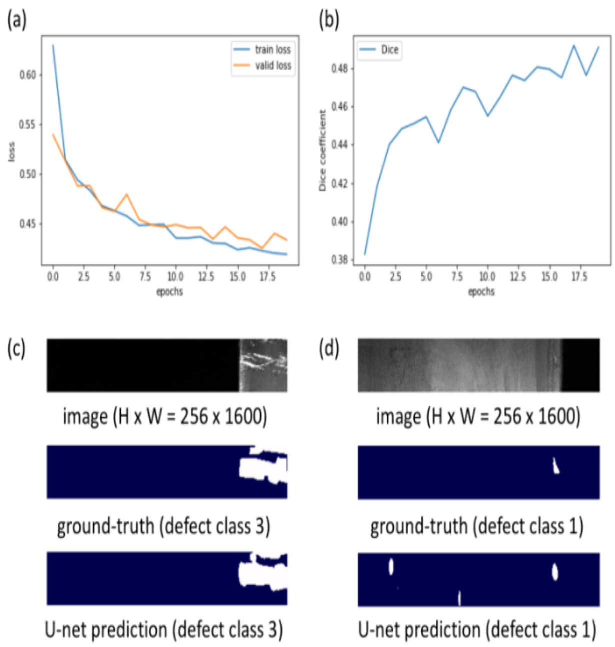


Figure 5: Results of the U-NET. (a) Loss of the training and the validation sets. (b) Average positive Dice score of the validation set during training. (c) A sample image containing defect of class 3. (d) A sample image containing defect of class 1

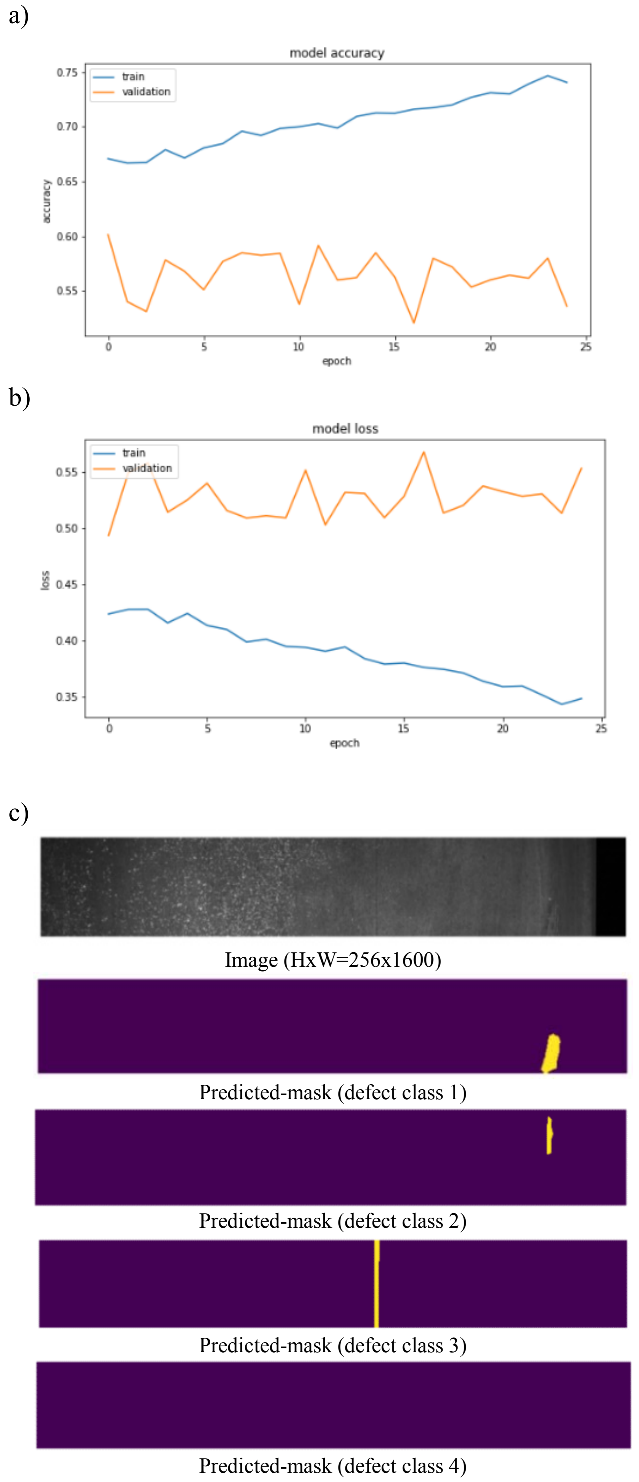


Figure 6: Results of the Deep Residual U-NET. (a) Model accuracy of the training and the validation sets. (b) Model loss of the training and the validation sets. (c) A predicted image sample with multiple defects.

VI. CONCLUSION

The proposed work anchorages the segmentation-based instance segmentation computer vision techniques and modern deep NNs including U-NET and Residual U-NET to localize steel surface defects and do region-based multiple labels classification. Among U-NET and Residual U-NET, Residual U-NET performs better with the Dice Coefficient score of 0.731. In the near future, we will explore more deep NNs on this problem, analyze their performance and improve classification accuracy.

ACKNOWLEDGMENT

This research is supported and funded by the Information and Communication Technology (ICT) Division, Government of the People's Republic of Bangladesh. The authors would like to express their sincere gratitude to the International University of Business Agriculture and Technology (IUBAT), Sector-10, Uttara Model Town, Dhaka, Bangladesh for their necessary supports.

REFERENCES

1] Caleb, P., and Steuer, M., Classification of surface defects on hot rolled steel using adaptive learning methods, Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Bright, UK, August 30 – September 1, 2000.

[2] Pakkanen, J., Iivarinen, J., Rautkorpi, R., and Rauhamaa, J., Content-based retrieval of surface defect images with PicSOM, International Journal of Fuzzy Systems, vol. 6, no. 3, pp.160-167, 2004.

[3] Keesug, C., Kyungmo, K., Lee, J., Development of defect classification algorithm for POSCO rolling strip surface inspection system, SICE-ICASE International Joint Conference, Busan, Korea, Oct. 18-21, 2006.

[4] Smriti, H. and Bhandari, S., A Simple Approach to Surface Defect Detection, the Third International Conference on Industrial and Information Systems, Kharagpur, India, December 8-10, 2008.

[5] Sharifzadeh, M., Alirezaee, S., Amirfattahi, R., and Sadri, S., Detection of steel defect using the image processing algorithms, 12th IEEE International Multitopic Conference, Karachi, Pakistan, December 23-24, 2008.

[6] Liu, Y., Hsu, Y., and Sun, Y., A Computer Vision System for Automatic Steel Surface Inspection, 5th IEEE Conference on Industrial Electronics and Applications, USA, June 15-17, 2010.

[7] Luiz, A., Flavio, L., and Paulo, E., Automatic detection of surface defects on rolled steel using Computer Vision and Artificial Neural Networks, 36th Annual Conference on IEEE Industrial Electronics Society, Glendale, USA, November 7-10, 2010.

[8] Song, K., and Yan, Y., A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects, Applied Surface Science, vol. 285, pp. 858-864, 2013.

[9] Song, K., Hu, S, Yan, Y., Automatic Recognition of Surface Defects on Hot-rolled Steel Strip using Scattering Convolution Network, Journal of Computational Information Systems, vol. 7, pp. 3049-3055, 2014.

[10] Wang, L., Xiaoying, Z., Jiangnan, P., and Gongyou, T., Transformer fault diagnosis using continuous sparse autoencoder, Springer Plus ,vol. 5, pp. 1-13, 2016.

[11] Mao, W., He, J., Li, Y., and Yan, Y., Bearing fault diagnosis with auto-encoder extreme learning machine: A comparative study, Journal of Mechanical Engineering Science, vol. 1, no. 1, pp. 1–19, 2016.

[12] Lu, C., Zhen-Ya, W., Wei-Li, Q., and Jian, M., Fault diagnosis of rotary machinery components using a stacked denoising auto-encoder based health state identification, Signal Processing, vol. 130, pp. 377-388, 2016.

[13] Severstal: Steel defect detection. [Online] Available at: https://www.kaggle.com/c/severstal-steel-defect-detection

[14] O. Ronneberger, P. Fischer, and T. Brox. U-NET: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015 J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[15] Z. Zhang, Q. Liu, and Y. Wang, “Road Extraction by Deep Residual U-NET,” IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 5, pp. 749–753, 2018