**Steel defect detection with high- frequency** camera images

**Md Abir Hasan & Mihir Roy**

**Bangladesh Army University of Science and Technology, Saidpur, Bangladesh**

**Abstract**

We focus on the problem of steel defect detection, where we are given images of a steel sheet taken by high-frequency cameras and aim to mark pixel wise defected area. We explored four deep learning methods: Random Forest, SVM, KNN and CNN to solve for the task and compared their performance and achieved highest accuracy of 77% with KNN where k==5 and 7.

**1. Introduction**

Steel is one of the most important building materials of modern times. And the production process of flat sheet steel is especially delicate. From heating and rolling to drying and cutting, several machines touch flat steel by the time it’s ready to ship. Before delivering the product, steel sheets need to undergo careful inspection to avoid defects and thus localizing and classifying surface defects on a steel sheet is crucial. Hence, automating the inspection process would accelerate the steel sheet production.

This project is targeted to find efficient and how much it can detect defect. Main goal is to find good accuracy of detection. For this we use those three deep learning method for now. Random forest, SVM, KNN, CNN. KNN achieves highest accuracy of 0.77.

**2. Related work**

For steel defect detection, the input is an image of a steel sheet and the output is a same-size segmented image with dense defect area marks. Since the output is a pixel-wise label for the input image, it’s essentially a segmentation task. Within image segmentation, there are two different sub-task, semantic segmentation and instance segmentation. These problems have been fundamental problems of computer vision for many years and receive great breakthrough since the CNN revival.

**2.1 Semantic Segmentation:**

Semantic segmentation could be considered as a per-pixel classification problem. The most popular CNN-based method on semantic segmentation is the Fully Convolutional Network (FCN) [1], which converts fully connected layers into 1x1 convolutional layers and achieves end-to-end per-pixel prediction. However, traditional FCN method suffers from the problem of resolution loss. There are mainly two main streams of methods proposed to tackle this problem. The first stream of methods [2] use atrous convolution”, which enlarges the feature map via linear interpolation. The second utilized deconvolution [3] to learn the up sampling process.

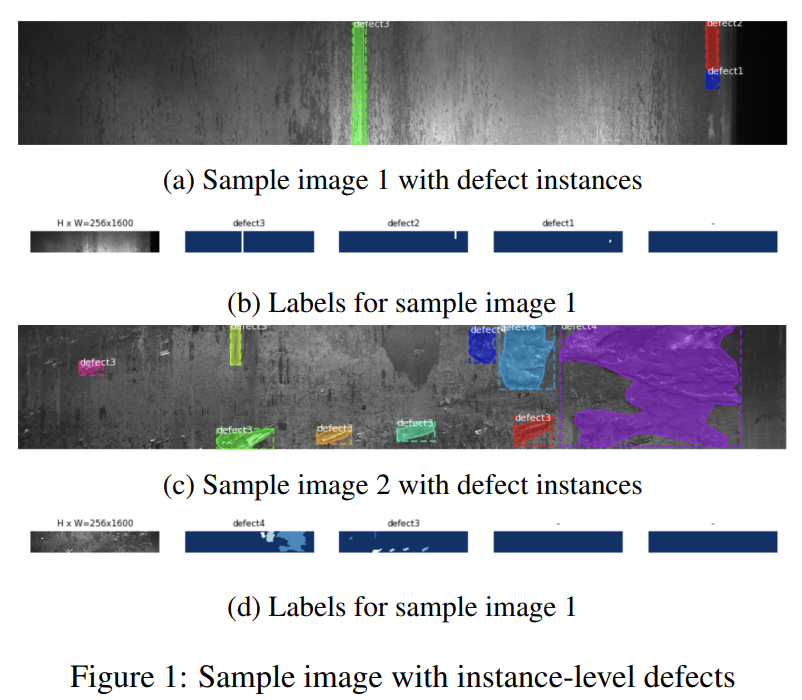
**2.2 Instance Segmentation:**

Compared with semantic segmentation, instance segmentation aims to predict not only class label, but also pixel-wise instance mask to localize varying numbers of instances presented in images. There are mainly two lines of methods to tackle instance segmentation problem: proposal-based methods and segmentation-based methods. Proposal-based methods are closely related to object detection [4, 5]. Mask R-CNN is the most widely-used method in this steam, which proposes to add a fully convolutional network branch based on [5] and achieves great performance on [6, 7]. The other is segmentation-based, which uses the output of semantic segmentation as input and obtain instance-aware segmentation result later. Among them, [8, 9] proposed to use a graphical model to infer the order of instances and [10, 11] utilized RNN to obtain one instance in each time step.

**3. Dataset and Features**

The dataset we are using is one consisting of high-frequency images of steel sheet, corresponding defects classes, and defect regions. This dataset is obtained from [12]. The images provided are of size 1600 × 256 × 1 and totals 12568 counts. Out of all the images, 5902 images are with defects and 6666 images are without defects. There are four labels of defects 1, 2, 3, and 4. In the images with defects, 897 images are of class 1 defect, 247 images are of class 2 defect, 5150 images are of class 3 defects, and 801 images are of class 4 defect. Additionally, 6239 images have only 1 class of defect, 425 images have two classes of defects, and only 2 images have 3 classes of defects. And important thing is we flatten all the images for use of this project.

Because of low end computer we compute only 4000 data to get result. Two samples are shown in Fig. 1



**4. Methods**

**4.1 SVM**

Support Vector Method. In this model image classified well. SVM is supervised learning model. Effective in high dimensional spaces. If the number of features is much greater than the number of samples, avoid over-fitting in choosing kernel function and regularization term is crucial. After importing dataset we create two array. One of them contains class and one image id. Then we put image id on stack. Flattening image array then normalize that data. After normalizing we split dataset into train and test. Now we use those X train, X test and y train, y test for this SVM model. By importing SVM class and set rbf kernel we fit X train and y train. Ready to compile.

|  |
| --- |
| C:\Users\WC\AppData\Local\Microsoft\Windows\INetCache\Content.Word\svm2.png  Fig2: SVM Classifier |

**4.2 KNN**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using this algorithm. K- Nearest Neighbour after importing dataset we create two array. One of them contains class and one image id. Then we put image id on stack. Flattening image array then normalize that data. After normalizing we split dataset into train and test. Now we use those X train, X test and y train, y test for this KNN model. By importing KNeighborClassifier class and set initial neighbour to 2.Then we fit X train and y train. Ready to compile. Main theme is select k and calculate distance from k point to category point. Closest distance of a category point is under that k point class.

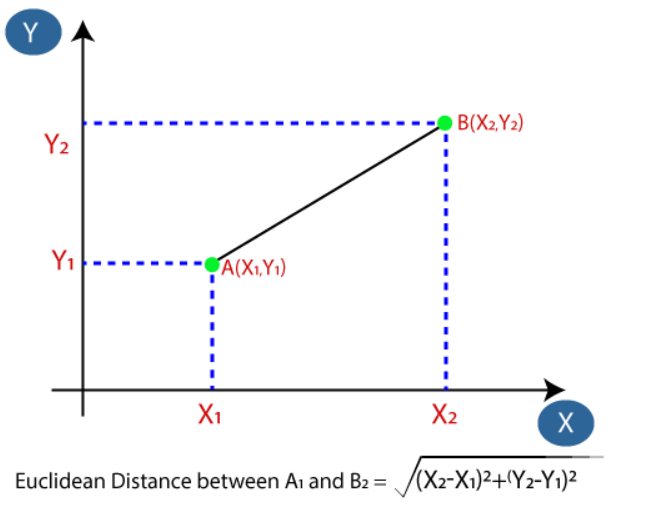


Fig3: Distance Between 2 point

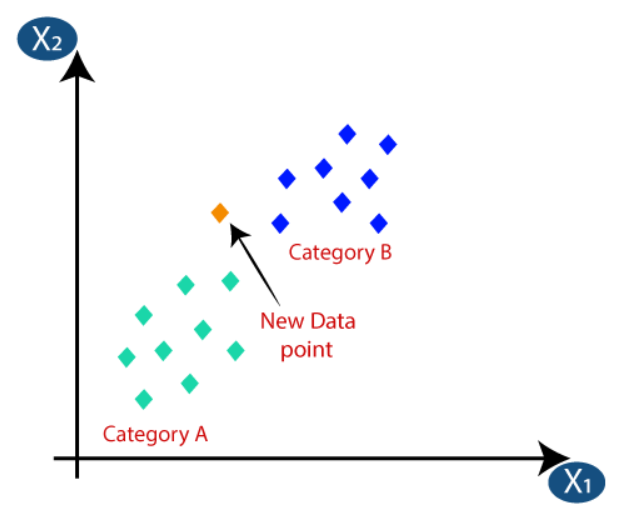


Fig4: KNN classifier

**4.3 CNN**

CNN is a type of neural network model which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image’s raw pixel data, trains the model, then extracts the features automatically for better classification. Convolutional Neural Network. After importing dataset we create two array. One of them contains class and one image id. Then we put image id on stack. Flattening image array then normalize that data. Importing necessary library classes we check dataset shape. We done image argumentation. In here we resize those images into 120x120 and put that with classid in different array. Encode classid array by labelencoder. Then splitting train and test data. Create model and compile. We will get accuracy.

|  |
| --- |
| Fig5: CNN structure |

**4.4 Random Forest**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. After importing dataset we create two array. One of them contains class and one image id. Then we put image id on stack. Flattening image array then normalize that data. Importing necessary library classes for this model and by using those we get our accuracy.

|  |
| --- |
| Fig6: Random Forest structure |

**5. Experiment results:**

|  |  |  |
| --- | --- | --- |
| No. | Model | Accuracy |
| 1 | SVM | 0.7580645161290323 |
| 2 | Random forest | 0.7338709677419355 |
| 3 | KNN | For, k=5&7 0.7661290322580645 |
| 4 | CNN | 0.756 |

**5.1 Random Forest**

We set estimator of 1000. It means it will generate 1000 tree and one single image will check in 1000 tree at a time.

**5.2 SVM:**

We use Radial basis function(rbf) karnel to perform SVM.

**5.3 KNN:**

We set neighbour list 2 to 9. Initially it will start from 2. For this we get high accuracy at k=5 and 7. They got same accuracy.

**5.4 CNN:**

For this first we find the shape of dataset. Then we did reshaping on images into 120 X 120. After label encoding we split dataset.

Then construct the model with input shape 120,120 and channel 3. Pool size 3 by 3. Inside we flatten images for get those intensity value for good.

We got accuracy of

Accuracy: 0.756

**Loss Curve:**

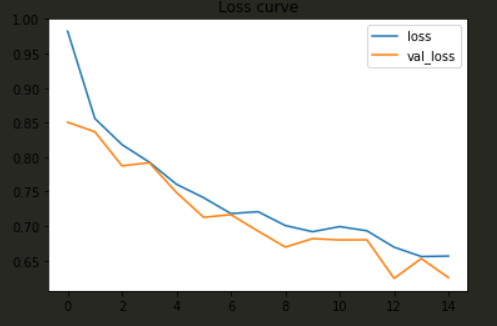
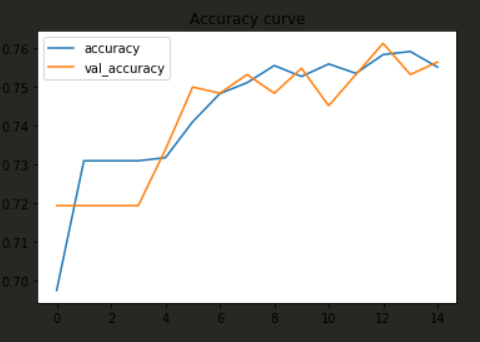


Fig7: loss curve

**Accuracy curve:**

Fig8: Accuracy curve



**6. Discussion:**

**6.1 SVM:**

In this model image classified well but I think we can get more accuracy on other model. SVM is supervised learning model. Effective in high dimensional spaces.

If the number of features is much greater than the number of samples, avoid over-fitting in choosing kernel function and regularization term is crucial. That’s why it will not give proper accuracy on this project.

**6.2 Random forest:**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the *concept of ensemble learning****,****which is a process of*combining multiple classifiers to solve a complex problem and to improve the performance of the model. It use various model, a combination of models. Finally it gives a voted output with best outcome. More the datasets are more accurate the accuracy it gives. I think in this project random forest will give most accuracy but we didn’t use all the datasets that’s why we find less accuracy then other.

**6.3 KNN:**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using this algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. In this project by using 4000 data and we get highest accuracy of 77%.

**6.4CNN:**  
CNN is a type of neural network model which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image’s raw pixel data, trains the model, then extracts the features automatically for better classification. CNN is good for this project too. We got 76% accuracy on it and much efficient than others.

**7. Future Work:**

We need to measure accuracy without reshaping size. We resize image in this project. For this project future work is measure everything without reshaping size. Use fully original size of image. Due to the imbalance between different kinds of labels, more data augmentation processes such as rotation and scaling on the images should be performed on the rare defect classes

**8. Conclusion:**

The models outputs the pixel wise defected area with corresponding labels. Among all the models, the task and compared their performance and achieved highest accuracy of 0.77 with KNN where k=5 and 7 but it is not efficient. And I think random forest will give more accuracy if we give more data to train. We used 4000 data in these models. More we increases data, random forest gives higher and higher accuracy. But for now because of low end device we use only 4000 data and for this KNN gives efficient result.

**9. Contributions:**

Abir is responsible for training and use models for this project and Mihir is responsible for dataset organization and related work.

**References:**

1. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.
2. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017
3. H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE international conference on computer vision, pages 1520– 1528, 2015
4. R. Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015
5. T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and ´ S. Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017
6. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- ´ mon objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.
7. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016.
8. Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In Proceedings of the IEEE International Conference on Computer Vision, pages 2614–2622, 2015
9. Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 669–677, 2016.
10. B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In European conference on computer vision, pages 312–329. Springer, 2016.
11. L. Shen, Z. Lin, and Q. Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In European conference on computer vision, pages 467–482. Springer, 2016.
12. Severstal: Steel defect detection
13. F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on  
    computer vision and pattern recognition*, pages 1251–1258,  
    2017.