AUTOMATIC STEEL DEFECTS DETECTION BASED ON AI

SHENGCHANG ZHANG { victorzh } JIANCHEN XU { jx2318 } ZHENG NIE { zhengnie}

CS221: Artificial Intelligence
Stanford / Autumn 2019-2020

OBJECTIVES

Automatic steel defect detection based on surface images is a challenging task in industry. In this project, we converted this problem to image segmentation problems and applied advanced AI technology to achieve this goal. The final product of this project will be an end-to-end deep learning model. Given steel images as input, the model will output the masks of the images with highlight defection regions.

RELATED WORKS

In this project, we mainly focus on models with spatial pyramid pooling and encoder-decoder structure. Spatial pyramid pooling works by keeping partitioning image into smaller sub-regions and then taking weighted sum of the number of matches at each sub-region level. Encoder-decoder structure apply encoder part to gradually decrease feature maps and captures higher semantic information, decoder part to gradually recovers the spatial information.

DATA

The training set includes 12568 images. The overall training data has 6666 images with 7095 human labeled defects regions. Defection classes are quite unbalanced. In terms of sample and specially in terms of area,Our network may have a hard time finding class 1 and 2 two because of their small size.

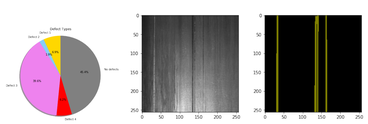

Figure 4: Distribution of defection classes and image samples

BASELINE MODEL

Baseline model is a bare bone DeeplabV3+ by removing encoder module. Input will first feed in to a pre-trained Reset101 model then feed into DeeplabV3+ decoder module directly which basically 2 convolution layers followed by a up sample layer, batch norm and ReLu activation functions are added followed by echo of these 2 convolution layers. The first convolution layer in decoder module has 2048 input channels, 256 output channels, with kernel size 3. The second convolution layer has 256 input channels, 256 output channels, with kernel size 3. we kept the upsample layer to be consistent with DeepLabV3+ decoder structure

Figure 5: Bare Bone DeepLabV3+

PRE-PROCESSING & METRIC

Data Preprocessing
we performed input image resizing, reduced image size from 256 x 1700 to 256 x 256. Then encode segmentation label and classification label together into a 2D mask matrix with same size as input data. Each element in the mask has value from 0 to 4, which corresponding to 4 types of defects, none defect or background area will be marked as 0.

Evaluation Matrices
we calculated mean IoU [2] for data set under test. So if we use X_i denotes each input image, Y_i denotes mask matrix. N denotes total test data set size, mean IOU could be calculate as:

$$mIoU = \frac{1}{N}\sum_{i}^{N}\frac{X_i \cap Y_i}{X_i \cup Y_i}$$

MODEL

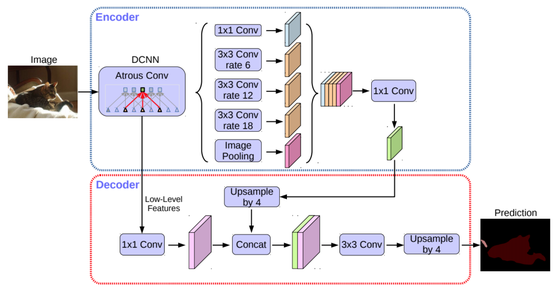

Figure 1: Model Architecture

Model Description
Model utilized encoder-decoder structure, based on the DeepLabv3+ [1].The encoder start with 5 normal convolution modules, the output, on one hand, will be copied and saved as an input to the decoder. On the other hand, be passed to 4 Atrous convolution modules and 1 average max pooling module in parallel. The decoder part has two inputs, the first input from normal convolution modules will be passed to another convolution module first, then concatenate with the final output from encoder part. The total trainable parameters in our model implementation is 59,340,197

Model Training
We trained all models on Google Colab with Nvida Telsa P100-GPU with learning rate=0.01 on scheduler with 10 times decrease, batch size = 16, weight decay = 1e-4, and data split is train/evaluation/test=80/10/10.

Model Results
On test set, we got 0.29 mIoU scores with baseline model. We got 0.56 mIoU with DeepLabV3+ model. Later model improved by 0.27. The max IoU we got is 0.96. At same time, we got a defect classification accuracy of 0.9

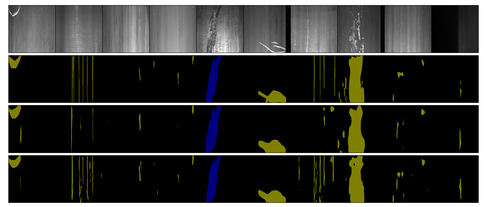

Figure 2: Comparison between ground truth vs baseline model vs trained model

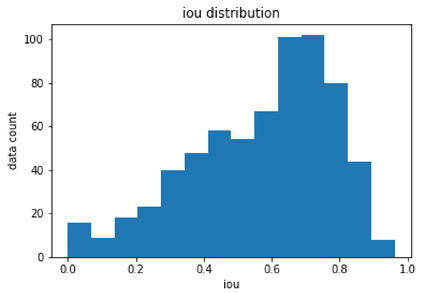

Figure 3: IoU Accuracy of distribution

CONCLUSION,ANALYSIS & FUTURE WORKS

Analysis
1. The encoder module helped on learning more details on the segmentation edges and could detect more detailed defect areas.
2. Unbalanced data could lead to low mIoU score. Data augmentation could improve this.
3. Model could detect the defect that did not labeled in the test set, but it will impact IoU result.

Future Works: We will continue improving model mIoU performance and implement model inference serving infrastructure to be capable for a production deployment.

REFERENCES

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[2] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

CONTACT INFORMATION

github https://tinyurl.com/tzet39g
email {victorzh,jx2318,zhengnie}stanford.edu