INTP22-ML-4

STEEL DEFECT DETECTION USING COMPUTER VISION

PHASE-2 REPORT

Submitted by:

PAVAN KUMAR M

Under the mentorship of Devesh Tarasia

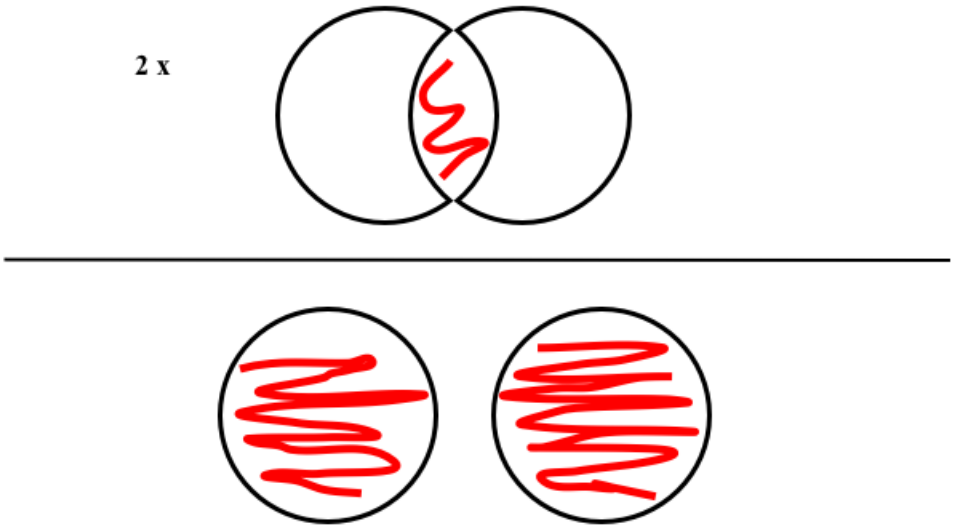
PHASE 2 OBJECTIVE

1. Deep learning model selection
2. Training the model
3. Testing the model

Dice Coefficient

The mean Dice Coefficient is applied to assess the model of semantic segmentation and for the comparison of the pixel consistency between the predicted segmentation and its relative ground truth value. X is the value of the predicted set of pixels and Y is the relative ground truth value. We can see the formula as following:

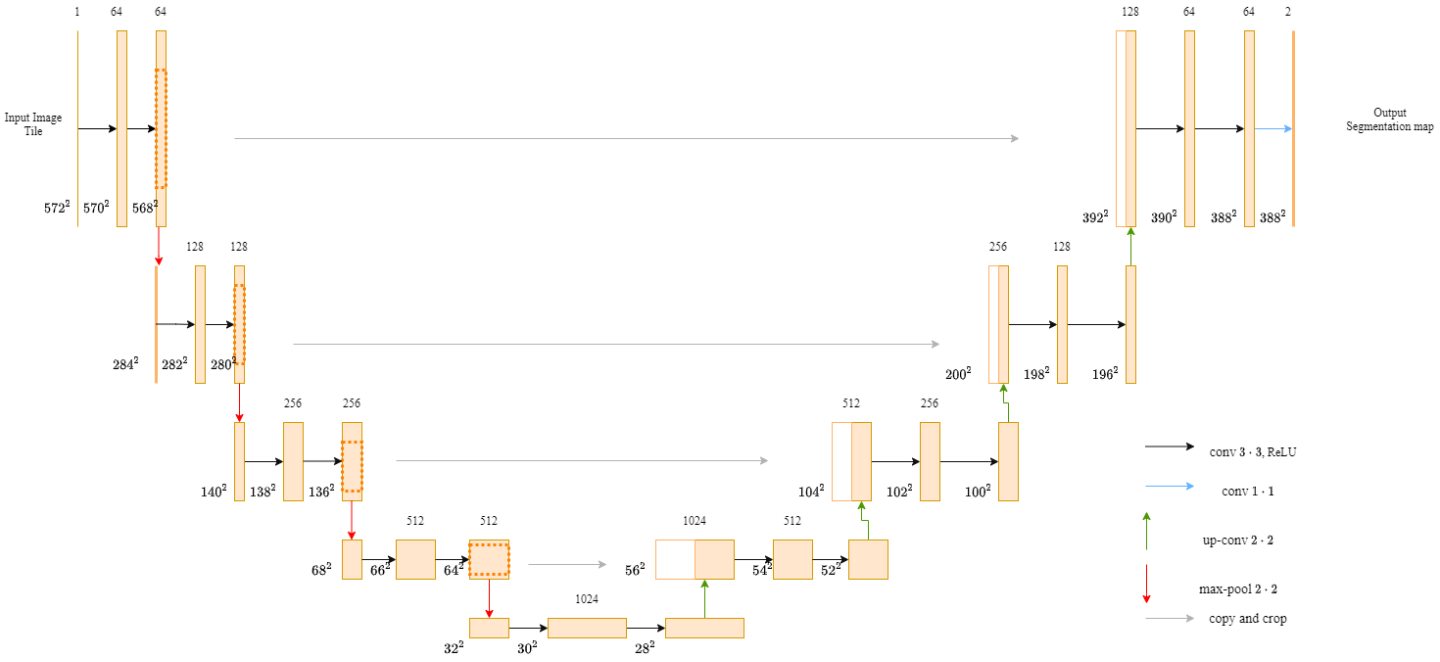
$$Dice(X,Y) = \frac{2 * |X \cap Y|}{|X| + |Y|}$$



I have selected UNET model for this project

UNET

Unet is currently the most popular semantic segmentation model. It adopts the structure of the encoder and decoder. Encoder is simply a conventional stack of maximum pool layer and convolution layer, whereas the decoder utilizes the transposed convolution to achieve accurate positioning. As a result, it is the end-to-end convolutional network.



Unet Architecture

Model Implementation

In this part, according to the dataset and model used, some part need to be changed to fit the dataset.

ResUnet

Since ResUnet is composed of ResNet and Unet, the model can be divided into these two parts. The difference between ResNet 18 and ResNet 50 are the identity blocks. The former one is called basic block and the later one is called bottleneck. The implementation and parameters of the ResUnet model used are presented in the below listings:

Algorithm 3 ResNet architecture

```
procedure RESNETENCODER(resnet)
    layers ⟵ intended layers
    if resnet == 'resnet18' then                          ▷ block: BasicBlock
        layers ⟵ zero padding 3 × 3
        layers ⟵ convolution 7 × 7
        layers ⟵ batch normalization
        layers ⟵ ReLu
        layers ⟵ maximum pooling
        layers ⟵ conv block 64 × 64
        layers ⟵ 1 identity block 64 × 64
        layers ⟵ conv block 128 × 128
        layers ⟵ 1 identity block 128 × 128)
        layers ⟵ conv block 256 × 256
        layers ⟵ 1 identity block 256 × 256
        layers ⟵ conv block 512 × 512
        layers ⟵ 1 identity block 512 × 512
    else if resnet == 'resnet50' then                     ▷ block: Bottleneck
        layers ⟵ zero padding 3 × 3
        layers ⟵ convolution 7 × 7
        layers ⟵ batch normalization
        layers ⟵ ReLu
        layers ⟵ maximum pooling
        layers ⟵ conv block 64 × 64 × 256
        layers ⟵ 2 identity blocks 64 × 64 × 256
        layers ⟵ conv block 128 × 128 × 512
        layers ⟵ 3 identity blocks 128 × 128 × 512
        layers ⟵ conv block 256 × 256 × 1024
        layers ⟵ 5 identity blocks 256 × 256 × 1024
        layers ⟵ conv block 512 × 512 × 2048
        layers ⟵ 2 identity blocks 512 × 512 × 2048
    output layers
```

Algorithm Implementation

Dice Coefficient

The dice coefficient algorithm was also implemented in Python by using the mathematical formula as described, resulting in the below algorithm.

```
procedure DICE(pred, mask)
    batch_size ⟵ length(pred)
    change pred dimension
    change mask dimension
    pred ⟵ float elements in pred if elements>0.5
    mask ⟵ float elements in mask if elements>0.5
    let smooth ⟵ 0.0001
    intersection ⟵ sum(pred × mask)
    dice_pos ⟵ (2. × intersection + smooth) / (sum(pred) + sum(mask) + smooth)
    intersection ⟵ sum(element if element in (pred + mask) == 0)
    dice_neg ⟵ (2. × intersection + smooth) / ( sum(element if element in pred ==
0) + sum(element if element in mask == 0) + smooth)
    dice ⟵ (dice_pos + dice_neg) / 2.0
output dice
```

Controlled Experiment with training data

After applying data preprocessing to increase the data availability for the training process, independent variables are set in this controlled experiment, the training data is transformed to horizontally flipped images. The following parameters are considered for this project.

- Batch size = 16
- Epochs = 30
- Learning rate = 10^{-3}

- After completing through the 30 Epochs it is found that the best values for dice_coef and val_loss is obtained as 0.6461 and 0.0145 respectively at 29th Epoch.
- So, the best weighted values are considered for testing the model.
- And training time took nearly 4 hours.

Testing

- In order to test the model the dataset is given with the sample_submission.csv file which consists of ImageId, the model will consider a bunch of images as a batch size and starts predicting the EncodedPixel which determines the defective region and also the model classifies the defect type of that particular image.
- And it is observed that model has identified the defective region along with the defect class type pretty accurately.

Comparison of Different Algorithm Performance

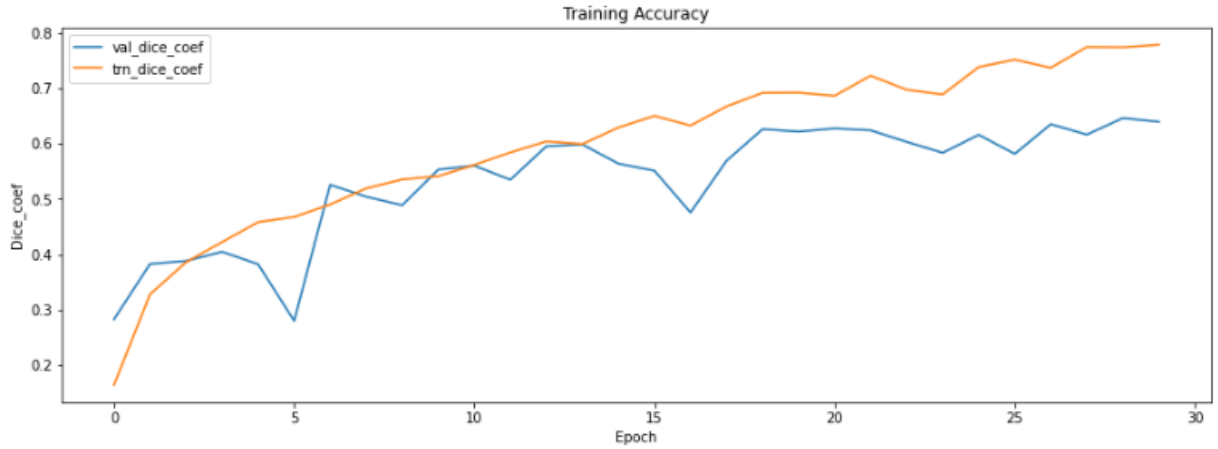
Table below shows the brief results of dice coefficient (best value achieved) of different models. It is obviously demonstrated that there is the highest dice coefficient score of improved mask R-CNN (0.81), but it takes the longest training time (12 hours), while the worst performance is Deeplab v3 plus.

	Basic Mask R-CNN	ResUnet	Deeplab v3 plus	Improved Mask R-CNN
Dice Coefficient	0.80891874	0.79928034	0.77426002	0.81071570
Training Time	around 9 hours	around 7 hours	around 7 hours	12 hours

Results

- **Unet**

The below figure reveals the training accuracy over Epochs that can be visualized using a histogram.



Conclusion

The main purpose of this project is to investigate the application of the deep learning approach to surface anomaly. The dataset is used for the research, including 5902 images with defects and 6666 distinct images. Amongst them, there exist four classes of defects. The models output the pixel-wise defected area with corresponding labels. One of the objective is aimed to explore the deep learning for image processing. After gaining the relevant knowledge including semantic segmentation and instance segmentation, we are capable to conduct the Unet model. Unet has performed well by considering the value of dice coefficient and training time compared to other models. Hence for semantic segmentation Unet is most popular. The research uses only a large dataset and it is imbalanced, which causes some problems

when training the models. More extensive work could be done by training more datasets, where other samples may improve the model.

Future Work

First of all, since it is an imbalanced dataset between various labels, some other kinds of approaches of data augmentation can be applied for the scarce anomaly classes on the input data, including but not limited to scaling, cropping, and increasing Gaussian noise. Furthermore, the new model also can be improved to get a much higher dice coefficient at least 90 by doing more deep experiments on the design part. On the other hand, efforts can be made to help steel industries to evaluate the severity of damage to each type of anomaly on the steel plate in order to determine whether to dispose of this steel plate via the information of the severity.