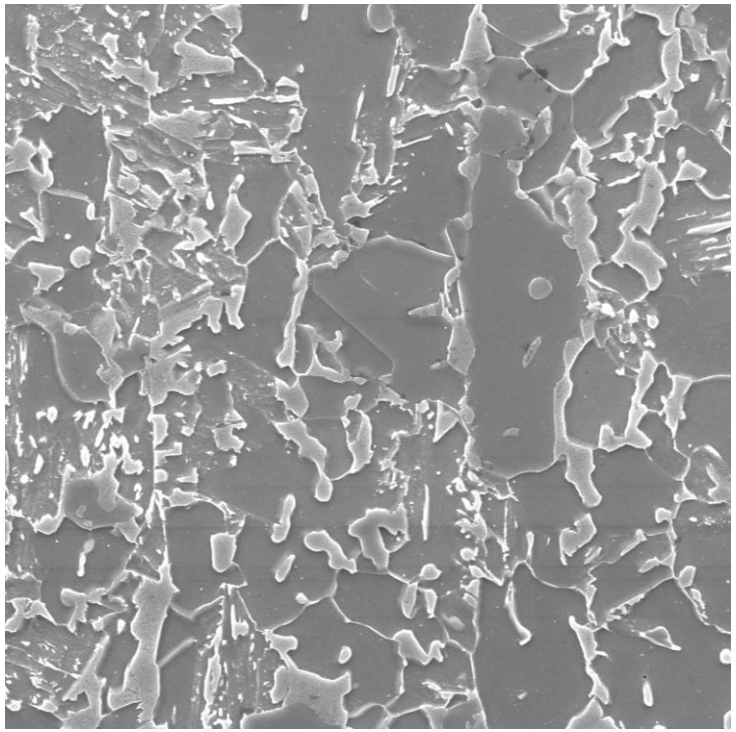
CVPR LAB

Steel Image Segmentation

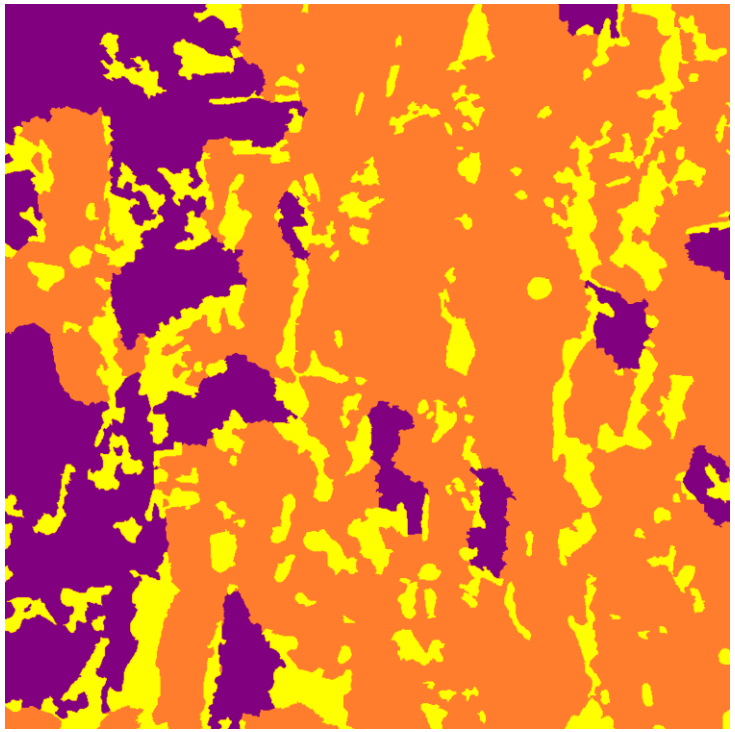
Swain Bishal Ranjan, Urokov Salohiddin Orif Ugli

Aim

To predict the contains of steel by looking at microscopic images



Steel Image

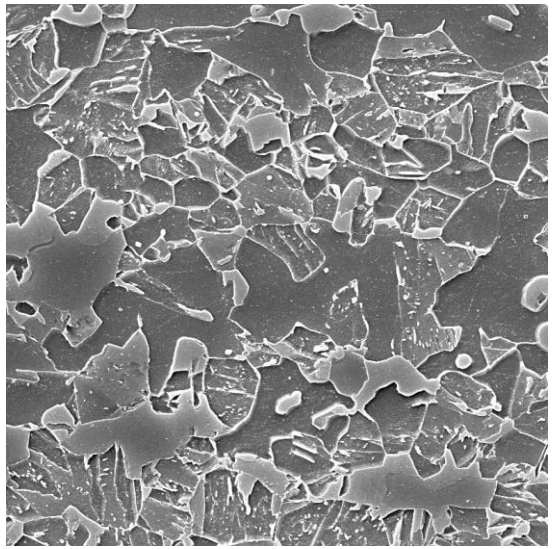


Labeled Image

Martensite Ferrite Bainite

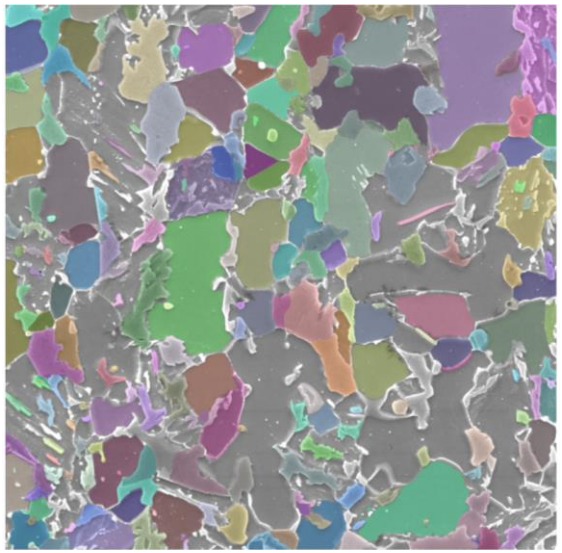
Problems

- Hand-labelling such data is not only **time consuming** but also **expensive.**
- Using off-the-self state of the art models is not viable as they are very generalized and finding minute structures needs exclusive training.
- Number of training samples is only – **6 images**
- The microstructures vary from steel to steel and vary with different magnification.



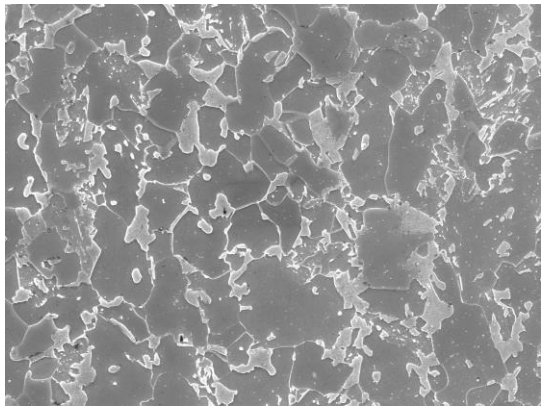
Test Image

NNUNET

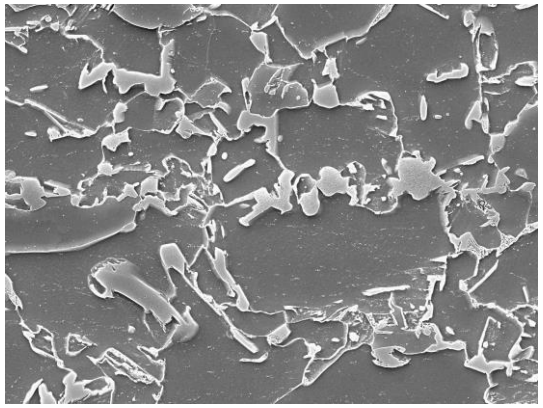


SEM

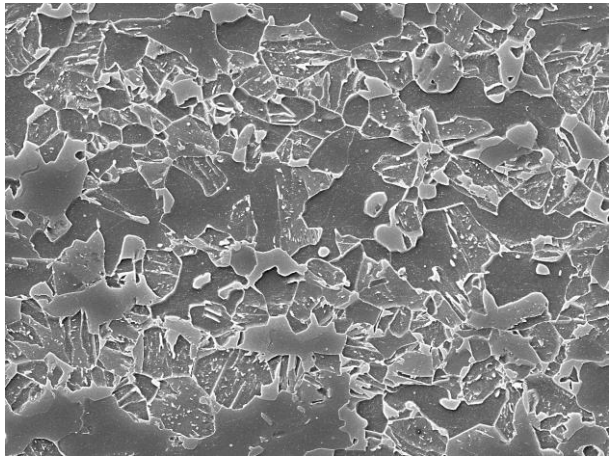
Different Types of Steel



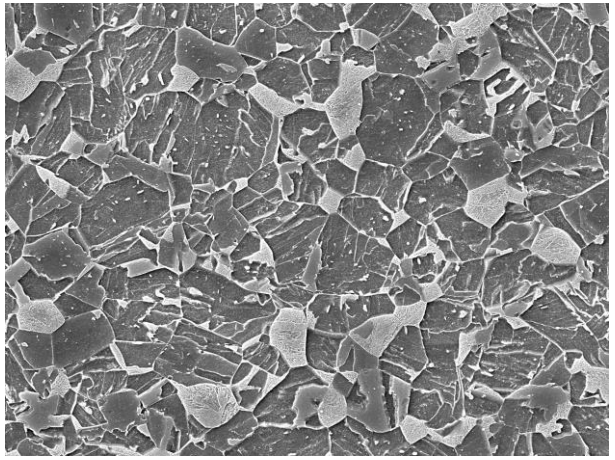
X3000



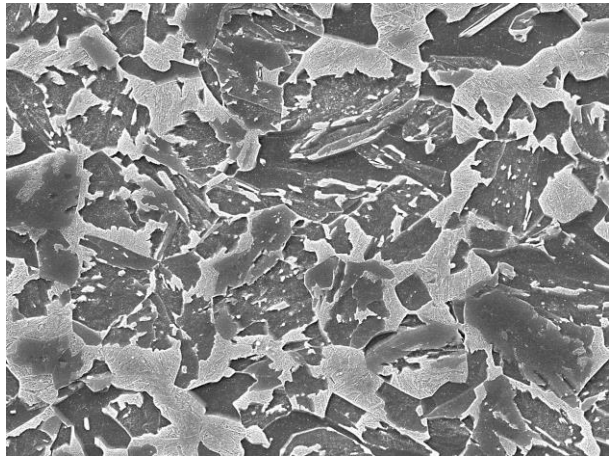
X5000



A Type



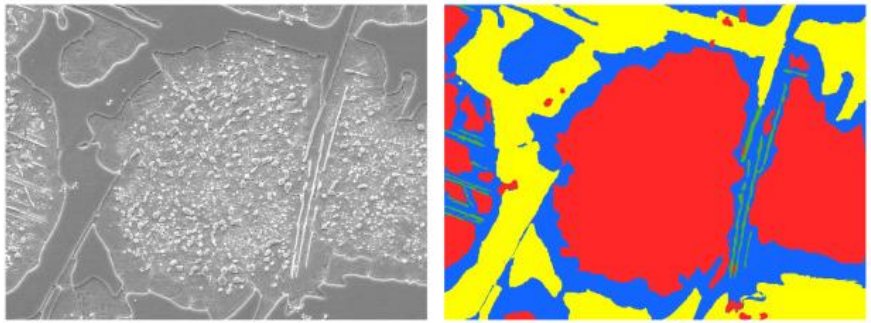
D3 Type



H2 Type

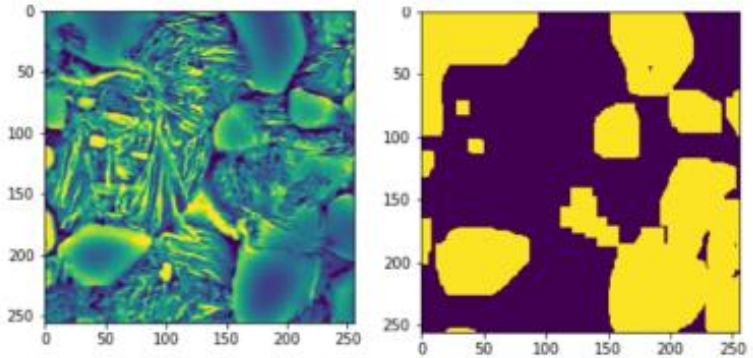
Previous Work

- A tutorial on the segmentation of metallographic images



  - Trains and predicts on same set of images.

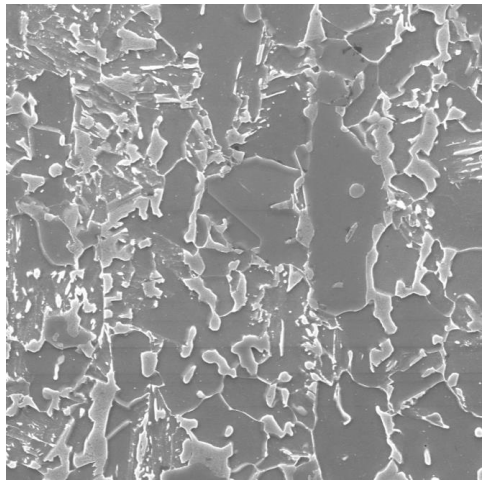
- Machine Learning Based Carbon steel image segmentation



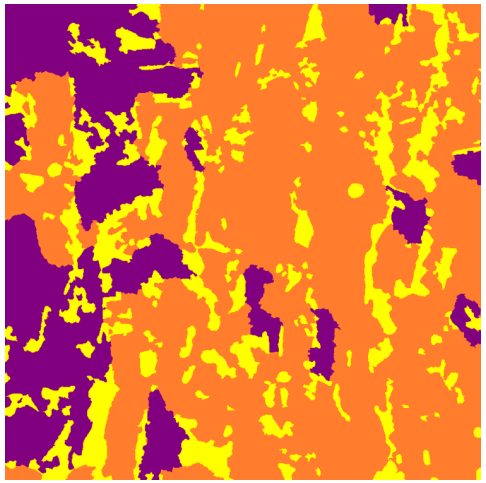
Aim

To identify and label the components of magnified steel images by training on one pair of known set of input and predicting results on different type and magnification of images.

1. Predicting Magnified Images of same steel type
2. Predicting images of different steel types



Steel Image



Labeled Image

Ways to mitigate -

Update - 2023 May 30

- ✓ Finding the appropriate augmentations to perform so as the model can generalize well on all inferencing images.
- ✓ Creating appropriate model to train.
- Finding the appropriate Loss function, activation function, and other hyper parameters to optimize the model
- Finding appropriate metrics for evaluation

In Brief,

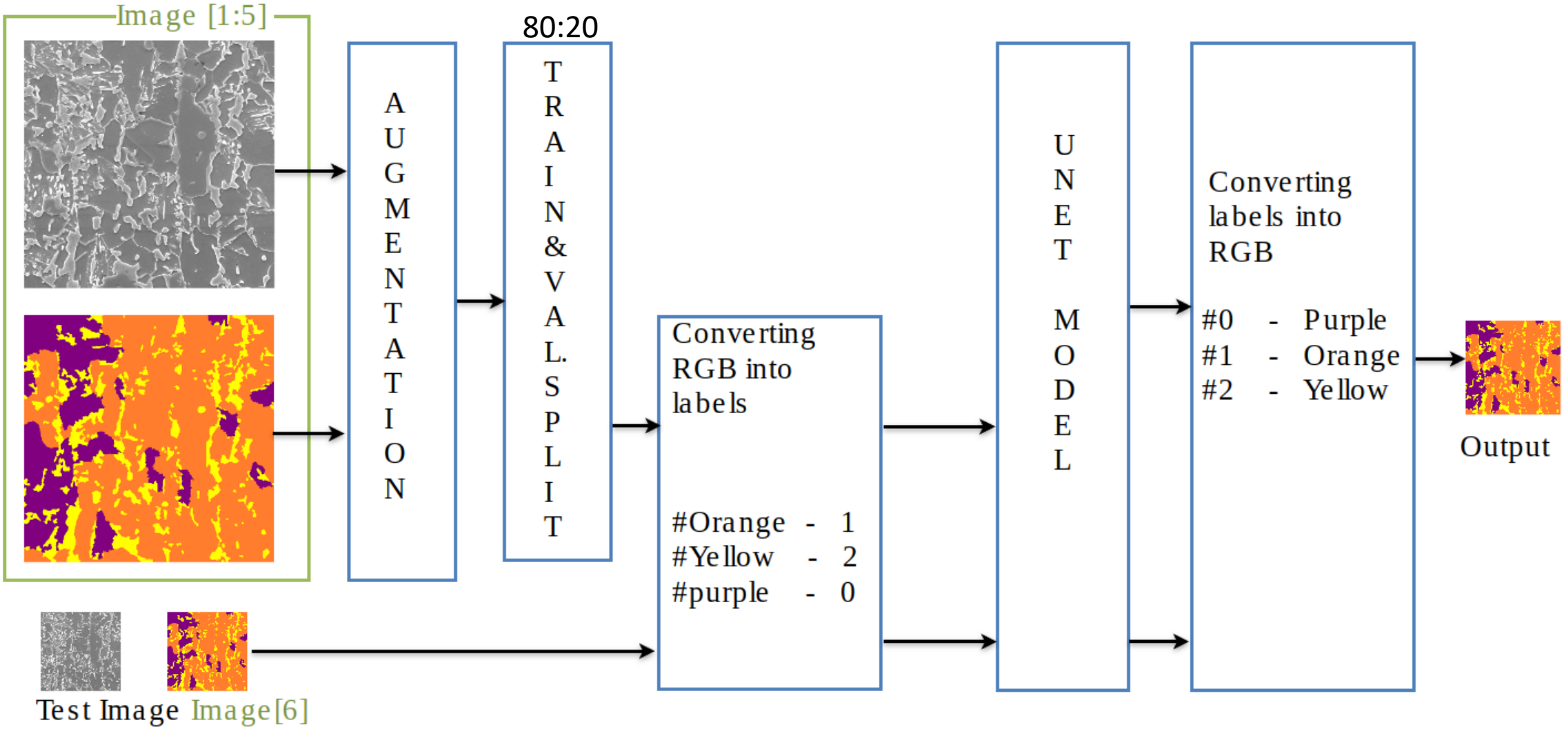
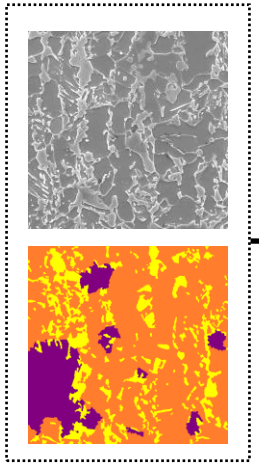


Image [1:5]
AUGMENTATION
80:20
TRAIN&VAL. SPLIT
Converting RGB into labels
#Orange - 1
#Yellow - 2
#purple - 0
UNET MODEL
Converting labels into RGB
#0 - Purple
#1 - Orange
#2 - Yellow
Output
Test Image
Image[6]

Augmentation

- The following Augmentations were performed on selected 5 image pairs
  - **Magnification (1-2.5x)**
    - To train the model on scalable image patterns
  - **Sliding Window (5 Pixels)**
    - To generate more number of images which slight change in image characteristics
  - **Random Intensity (0-10)**
    - As the intensity of Yellow/Orange/Purple label varies, generating more random intensity images will help train model better
  - **Random Gamma (1-10)**
    - To generate images and train the model on variable brightness and contrast as different steel types have different brightness and contrast properties.
  - **Horizontal and Vertical Flip**
  - **Rotation (10 degrees)**
    - Flip and Rotations are applied to train the model for detecting microstructures in images

Model Description



Input Images

Convert to corresponding labels of {0, 1, 2}

reshape

Set Model

Set Optimizer

Set Loss

Loop on Epoch

Train

Calculate Loss

Optimize

Run Validation

If val_acc[i]> val_acc[i-1]

Save Model State

Calculate Metrics

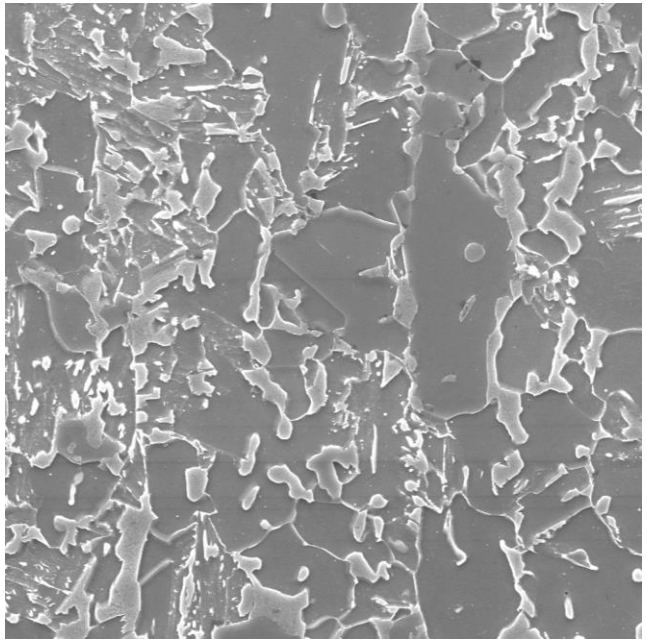
Model	UNET
Optimizer	Adam
Loss	Focal (0.25)

Setup
kernel size - 3
Batch Size - 16
Learning Rate - 0.00001
Channel - 16
Depth - 4

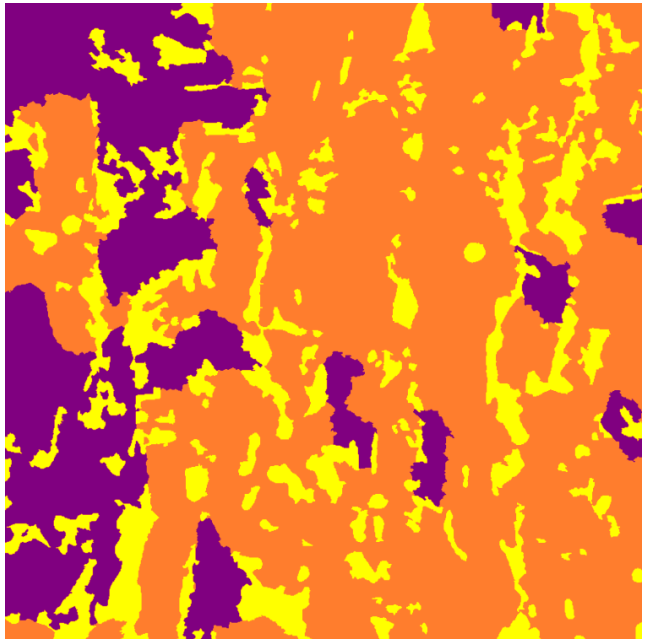
Metrics

Loss, Confusion Matrix, mIoU, Accuracy

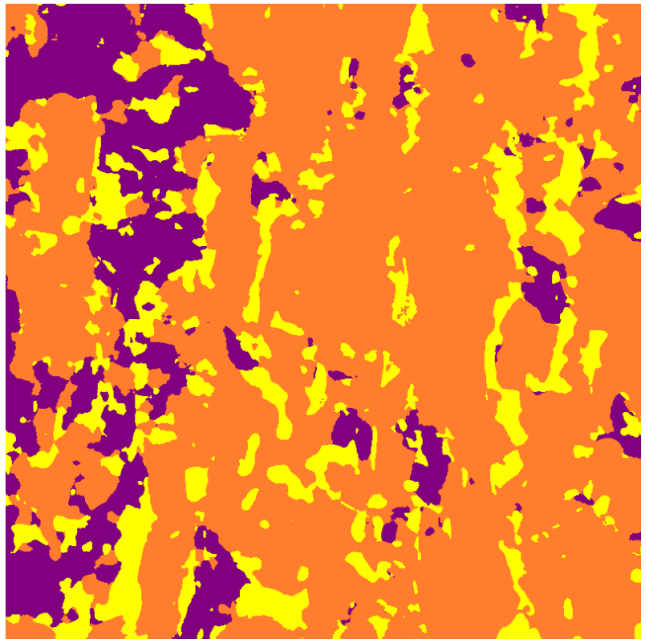
Output Results



Original



Label



Prediction

Test Image - 6.bmp

Training Accuracy- 91.68
Validation mIOU - 0.4684
Validation Acc - 95.31
Test Accuracy - 87.98
Dice Coefficient - 0.8609

Image train size – 800x800
Image test size – 1600x1600
Training time – 26 hours

Activation	ReLU
BlurPooling	None
Loss	Focal

Next to do -

- Make inference on other steel images.
- Need to perform various experiments to find the best loss and activation function
- Check the effects of addition of Blur Pooling
- Find appropriate metrics for evaluation.