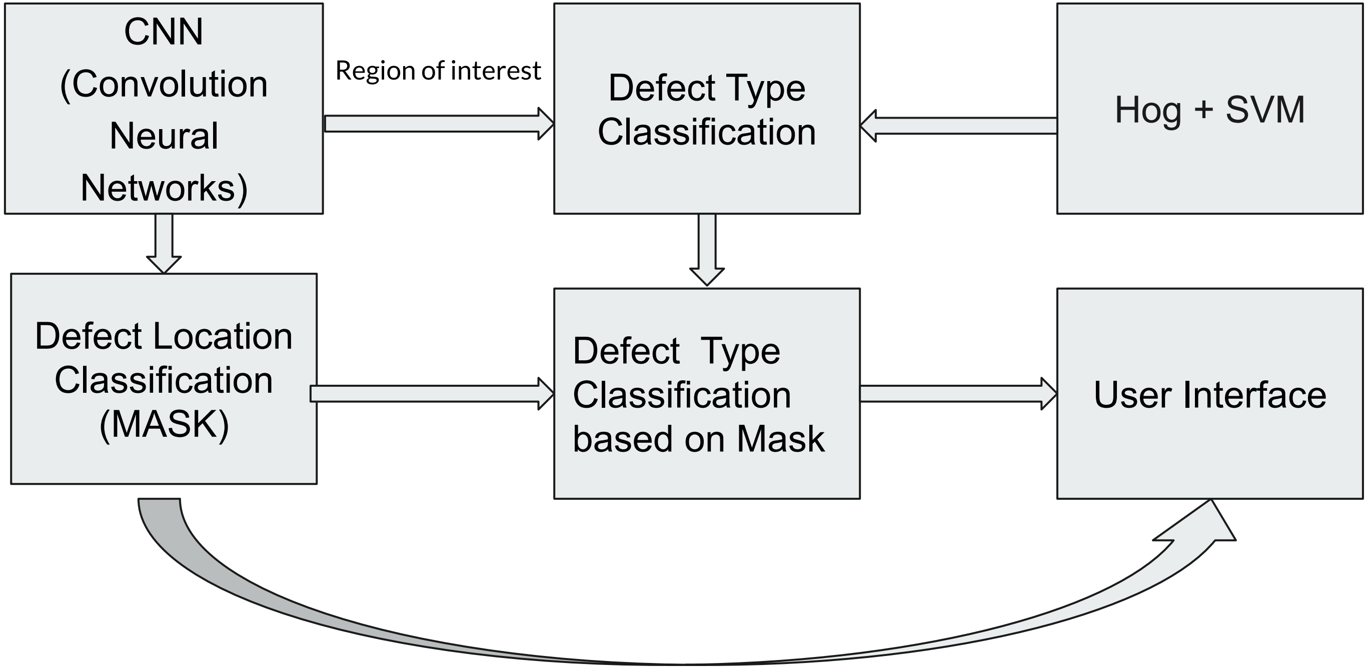
Steel Detection

Group 9 (A1): Brad Zhao, Danny Trinh, Xushan Hu

System Architecture



Components

-Defect location implementation in CNN

-Classification of defect either Hog - SVM, CNN or other methods

-User interface (Web vs Software interface)

CNN

-Tensorflow

-Keras

-OpenCV

-Pandas

-Python

CNN Difficulties

-Lack of data

- -Looking to augment data to resolve

-Long training times

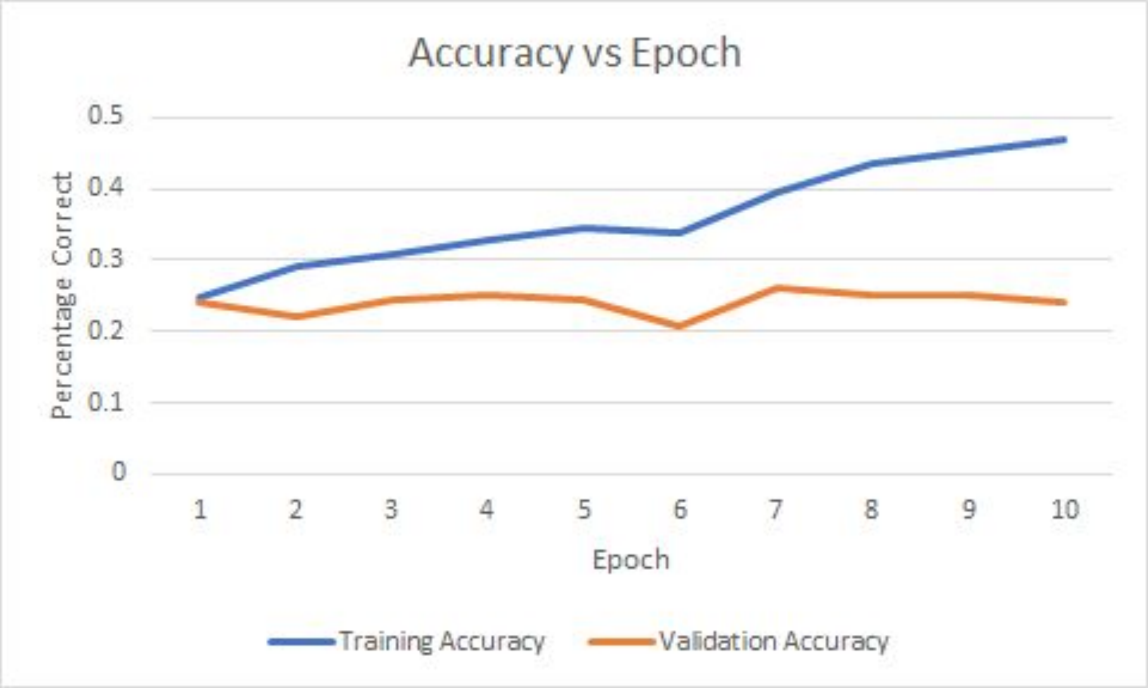
-BU SCC

- -Package version mismatches
- -Bugs in preloaded versions

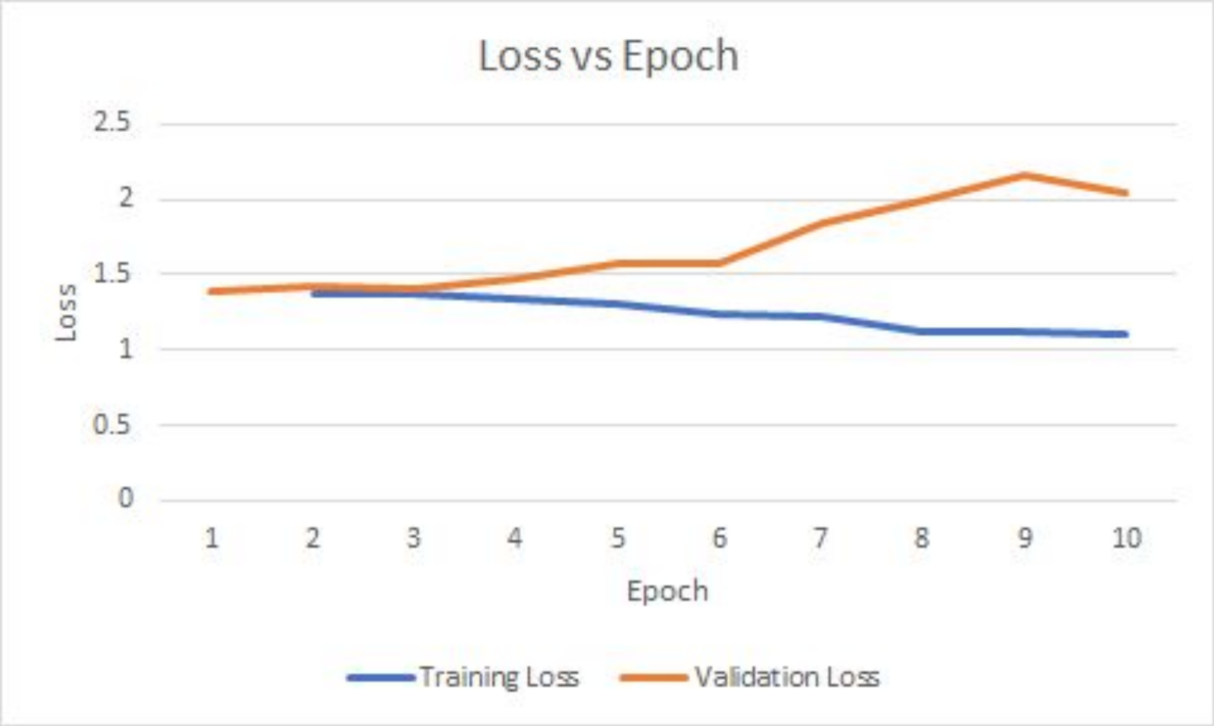
CNN Difficulties (cont.)

- Random classification (~25% classification)
    - Idea of using regions of interest (ROI)

CNN Epoch vs Correct Classification Rate



CNN Epoch vs Loss

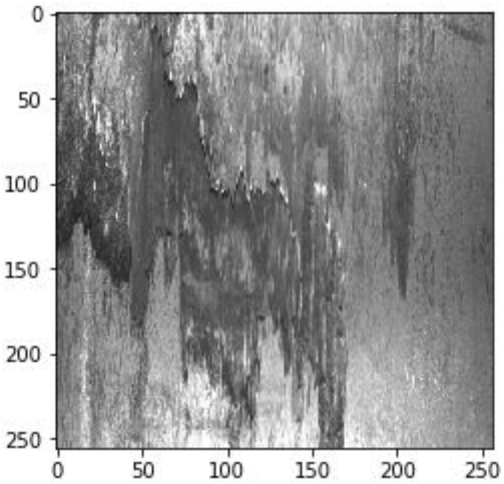


CNN Classification Results

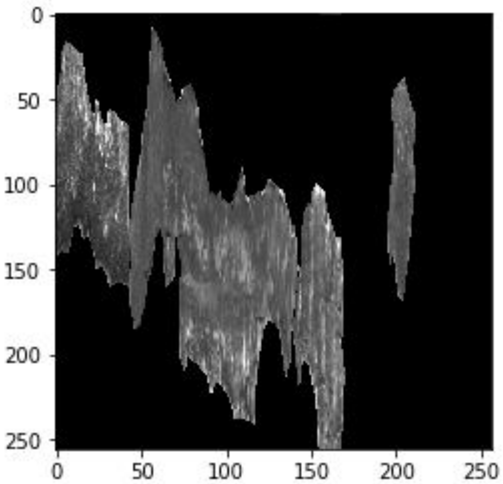
```
Test loss: 2.040714062638019
Test accuracy: 0.24120603501796722
199/199 [==============================] - 4s 22ms/step
[3 3 3 3 3 2 3 3 3 3 3 3 1 0 2 3 3 3 3 3 3 3 3 1 3 3 2 3 3 0 3 1 3 3 2 3 3
 3 3 3 2 3 2 3 3 0 3 3 2 3 3 3 1 3 3 3 0 3 3 0 3 3 3 3 0 3 0 3 3 3 3 3 3 3
 3 3 3 3 2 3 3 3 3 0 3 1 1 3 3 3 3 3 3 3 0 3 3 2 1 0 3 3 3 3 3 2 3 3 3 3 3
 1 3 3 3 3 1 3 0 3 1 3 0 2 1 3 3 3 2 1 3 3 3 3 0 3 0 3 3 3 3 3 3 3 3 3 3 3
 3 3 3 1 0 3 3 3 3 3 3 3 3 0 3 3 3 3 1 2 2 3 3 3 3 3 3 3 0 3 3 3 3 2 3 1 2
 3 3 3 3 2 3 3 3 3 3 3 3 1 0]
[3, 3, 3, 2, 2, 3, 3, 2, 2, 2, 3, 3, 3, 3, 0, 2, 3, 0, 3, 0, 3, 2, 0, 2, 0, 0, 2, 0, 3, 2, 3, 0, 2, 0, 3, 2, 2, 2, 3,
3, 0, 3, 2, 3, 3, 2, 0, 3, 3, 2, 0, 2, 0, 2, 0, 3, 3, 0, 0, 0, 0, 3, 3, 3, 0, 2, 2, 2, 2, 0, 2, 0, 0, 3, 3, 3, 3, 3,
2, 0, 2, 2, 2, 0, 0, 3, 3, 3, 2, 3, 0, 3, 3, 2, 2, 3, 1, 3, 1, 3, 0, 0, 0, 0, 3, 1, 0, 3, 0, 2, 3, 3, 3, 1, 3, 0, 3,
0, 1, 1, 1, 2, 1, 0, 0, 0, 2, 2, 3, 3, 0, 2, 0, 2, 0, 2, 1, 2, 0, 0, 2, 0, 0, 0, 0, 0, 2, 1, 0, 1, 0, 0, 1, 2, 2, 1,
1, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1]
```

CNN Training

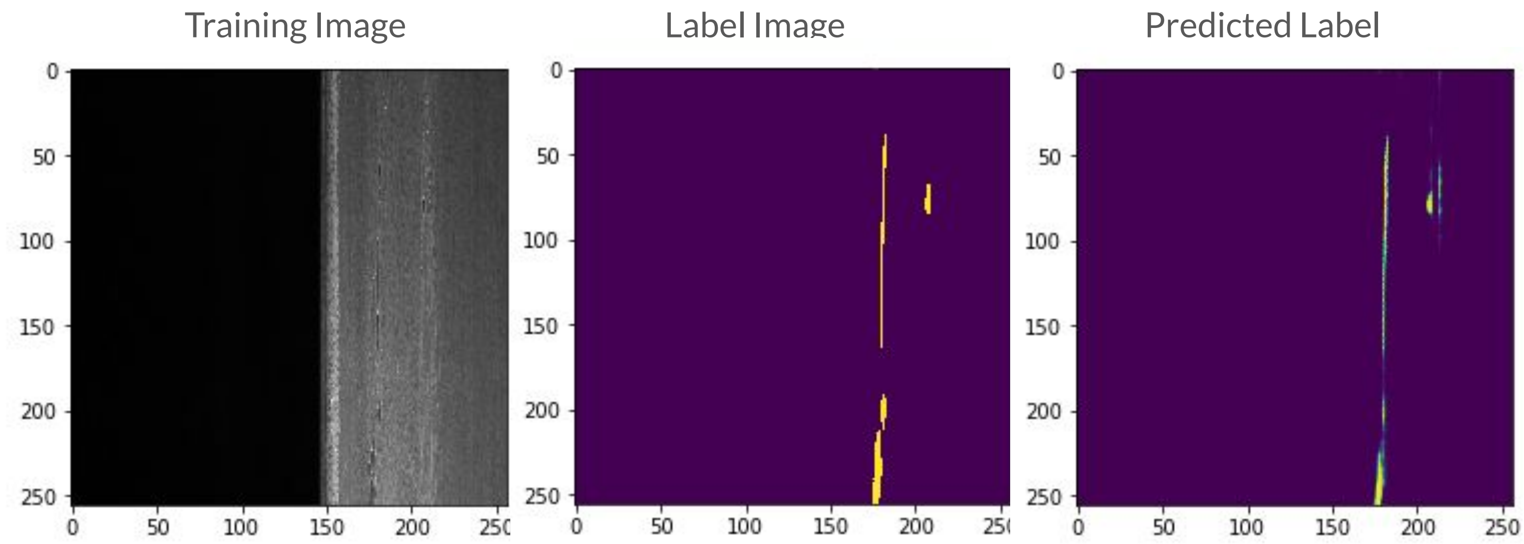
Training Image



ROI Training Image



CNN Defect Location Classification



Classification -- Hog + SVM model

Step 1: Analyse the dataset and decide the direction

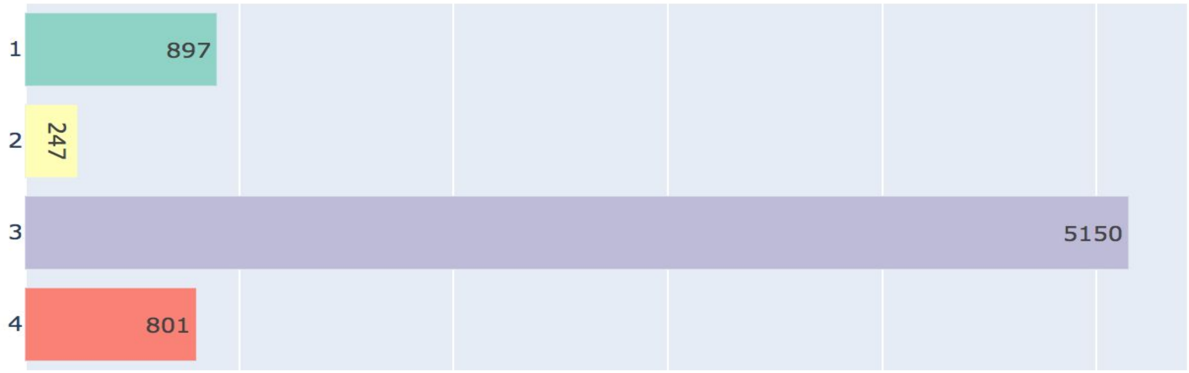
SVM model is a linear classifier. Linear classifier can be understood as mapping a series of data to classes. So we considered the image as feature and classId as label which is used for training the classification model. At first, we analysis the dataset and know more about the dataset. The results are as follows:

Classification -- Hog + SVM model

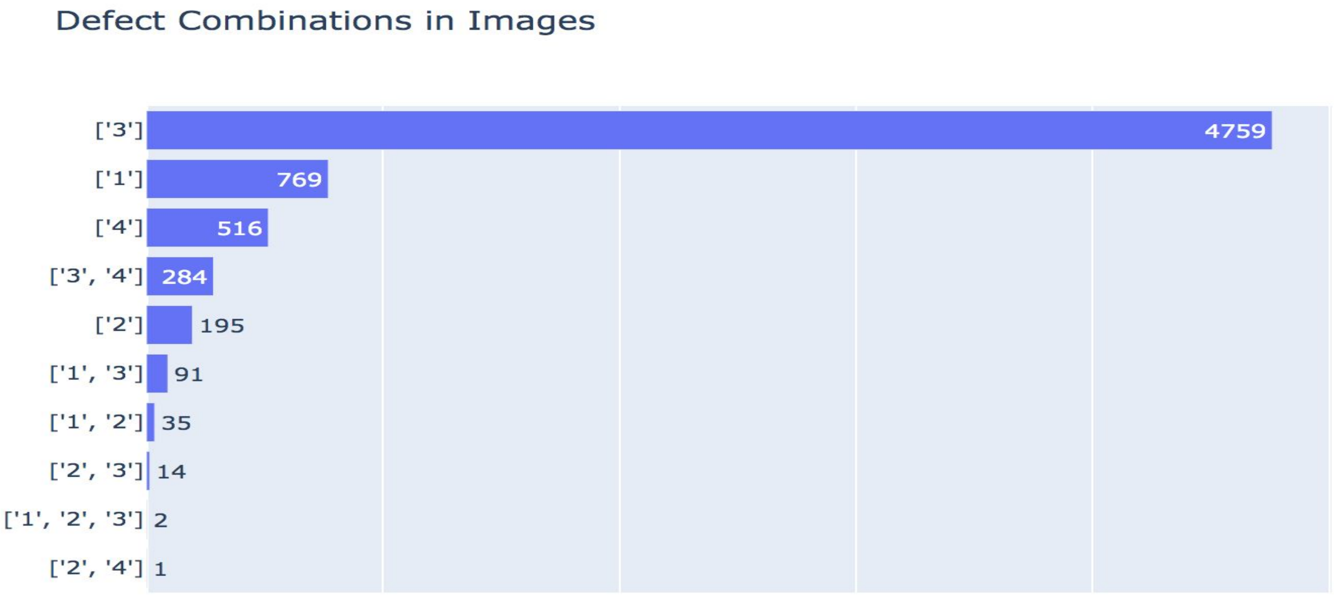
	ImageId	EncodedPixels	ClassId	Distinct Defect Types
0	0002cc93b.jpg	[29102 12 29346 24 29602 24 29858 24 30114 24 ...	[1]	1
1	0007a71bf.jpg	[18661 28 18863 82 19091 110 19347 110 19603 1...	[3]	1
2	000a4bcdd.jpg	[37607 3 37858 8 38108 14 38359 20 38610 25 38...	[1]	1
3	000f6bf48.jpg	[131973 1 132228 4 132483 6 132738 8 132993 11...	[4]	1
4	0014fce06.jpg	[229501 11 229741 33 229981 55 230221 77 23046...	[3]	1
5	0025bde0c.jpg	[8458 14 8707 35 8963 48 9219 71 9475 88 9731 ...	[3, 4]	2
6	002af848d.jpg	[290800 6 291055 13 291311 15 291566 18 291822...	[4]	1
7	002fc4e19.jpg	[146021 3 146275 10 146529 40 146783 46 147038...	[1, 2]	2
8	0030401a5.jpg	[186833 1 187089 3 187344 6 187600 7 187855 10...	[4]	1
9	0046839bd.jpg	[152926 1 153180 4 153434 6 153689 8 153943 11...	[3]	1

Classification -- Hog + SVM model

Defect: Count & Frequency with different color



Classification -- Hog +SVM model



Classification -- Hog + SVM model

Step 2: Why choose Hog + SVM model

The histogram of oriented gradient (HOG) feature is a feature descriptor used to detect objects in computer vision and image processing. It constructs the feature by calculating and counting the gradient direction histogram of the local region of the image. In an image, the appearance and shape of the local target can be well described by the directional density distribution of the gradient or edge.

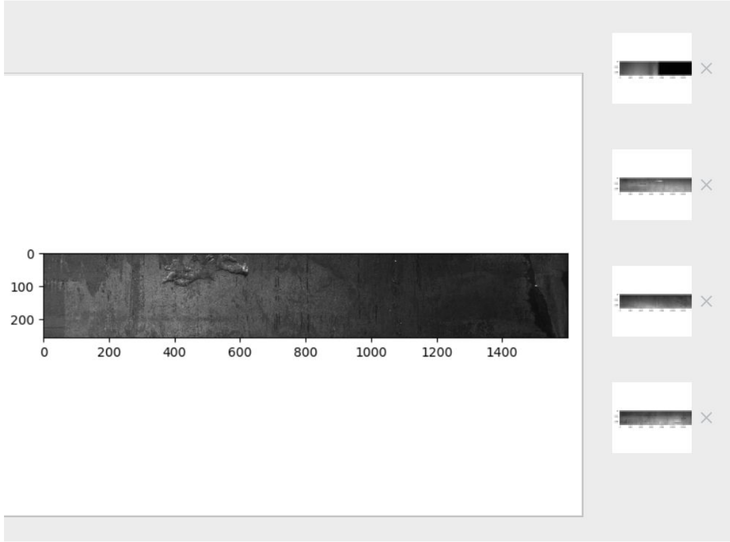
Classification -- Hog + SVM model

Advantages of Hog:

Because Hog is operated on the local grid element of image, it can keep good invariance to the geometric and optical deformation of image, and these two kinds of deformation will only appear in the larger space field. This point is suitable for detecting the defeat on the metal in our project which can improve the accuracy.

Classification -- Hog + SVM model

Step 3: Load train images and get Hog feature



Classification -- Hog + SVM model

907/5095	2864/5095
908/5095	2865/5095
909/5095	2866/5095
910/5095	2867/5095
911/5095	2868/5095
912/5095	2869/5095
913/5095	2870/5095
914/5095	2871/5095
915/5095	2872/5095
916/5095	2873/5095
917/5095	2874/5095

KNN(*k-nearest neighbors algorithm* **)**

This algorithm simply relies on the distance between feature vectors.we have the *labels* associated with each image so we can predict and return an actual *category* for the image

Cons:

- Classifier should store all the dataset of train data to compare them with the test data, which will take too much time predicting(testing) the result.
- It is hard to calculate the distance metric in high dimension.(overfitting)

Sprint 3 Next Steps

-Work on improving defect location detection

-Joint effort to try and get defect classification model working

-CNN: try region of interest training and testing to see if classification rate improves

- Hog + SVM: Extract hog features of positive and negative samples. Input them into SVM classifier training and get model

-Try other models(Alex-Net,etc)

-Try to implement web interface