COMPUTER VISION APPROACH FOR QUALITY INSPECTION OF STEEL SHEETS

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

by

Atharva Patil (13)
Kartiki Pande (12)
Aditya Kadam (36)
Rakshitha Krishnan (8)

Under the guidance of
Mrs. Madhumati Khuspe

Department of Instrumentation Engineering Vivekanand Education Society's Institute Of Technology Academic Year:2021-2022

<u>CERTIFICATE</u>

This is to certify that the project entitled "**A Computer Vision Approach For Quality inspection of steel sheets**" is a bonafide work of "**Atharva Patil(13), Kartiki Pande(12), Aditya Kadam(), Rakshitha Krishnan(8)**" submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Undergraduate** in **"Instrumentation"**.

Mrs. Madhumati Khuspe
Supervisor/guide

Mrs. Sangeeta Prasanna Ram
Head of Department

Dr. J.M. Nair
Principal

<u>Project Report Approval for B. E.</u>

This project report entitled (***A Computer Vision Approach For Quality Inspection of Steel Sheets***) by (***Atharva Patil, Kartiki Pande, Aditya Kadam, Rakshitha Krishnan***) is approved for the degree of Bachelor's (***Engineering***).

Examiners

1.--

2.--

Date:

Place:

DECLARATION

We declare that this written submission represents our ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

--------------------------------	------------------------------
Atharva Patil	Kartiki Pande
-------------------------------	-------------------------------
Aditya Kadam	Rakshitha Krishnan

Date:

Place: Mumbai

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our Project mentor, **Ms Madhumati Khuspe**, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report.

We would like to thank our Principal Dr.(Mrs). J. M. Nair & the Head of the department Prof. Dr. (Mrs) Sangeeta Prasanna Ram for allowing us to do this project and also the Staff and Management of the Vivekanand Education Society's Institute of Technology for providing their cooperation without which it would not have been possible to complete the work in time and successfully.

We would also like to thank our friends & family for their technical as well as moral support.

ABSTRACT

Steel is one of the most important building materials of modern times. 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. In this course of production damage caused to steel sheets is considerable. This has a considerable impact and hinders the quality of steel sheets.Defect detection techniques used in industries currently are manual . Manual defect detection has scope of human error and are comparatively more time consuming. Human error results in inaccuracy which results in decrease of efficiency. Our project is a deep learning to improve automation, increase efficiency, and maintain high quality in the production of steel sheets.In this project, we advance the steel defect inspection methods by designing **deep learning models that aim to detect multi-level defects from sample steel sheet images and classify them according to their corresponding classes**. We explore two deep learning methods including U-NET and MobilenetV2 to solve the steel defect detection problem with a **Dice coefficient accuracy of 0.3791 and 0.7562 correspondingly.**

INDEX

INTRODUCTION

The defect detection system in steel sheet surfaces plays a critical role in the steel sheets industry by detecting, localizing, recognizing, and subsequently correcting causative factors. The traditional steel surface defect detection is completed by manual visual inspection combined with traditional machine vision. There are some shortcomings in manual testing, such as low confidence and high labor intensity. The traditional target detection selects candidate regions on a given image; then, the features are extracted manually and the trained classifier is used for classification. This method has high time complexity and low precision and is difficult to meet the actual production needs of the steel industry. With the continuous development of the convolutional neural network, target detection based on deep learning has become the mainstream surface defect detection method.It is also quite necessary for controlling product quality and generating real-time analysis reports. The Detection process involves determining the existence of the steel surface defects from images taken from the industrial cameras. Localization locates all known content in the scene including the defective regions. Recognition takes the defect regions and infers the defect category according to the defect appearance. A defect detection system in steel sheet surfaces is thus a combined process of detection, localization, and classification. Typically, in steel mills, human inspectors manually perform the defect detection process on steel sheets. However, this procedure is very time-consuming, costly but lower efficient and does not meet up the requirement of real-time online defect detection. Many recent pieces of research are conducted on a combined approach of computer vision with machine learning methods to solve the requirements for real-time online defect detection on steel sheets. However, they apply some morphological operations on high-frequency images generated by low-cost industrial cameras and simple classifiers to solve the classification problem. Thus, these approaches lower accuracy and are unable to handle complex problems including multi-level classifications or localizing the defected area within a single image.

In this study, we advance the steel defect inspection methods by applying a modern segmentation approach to partition the image into various regions and designing a new machine learning model to feed the region pixels to detect the defect region from a single sample image of steel sheet and classify them according to their corresponding multi-level classes. We apply U-NET and MobilenetV2 to solve the given problem. The dice coefficient method is used to trace the accuracy of the selected machine learning models.

DATA VISUALIZATION

2.1 Classification

Data classification is done into four types of defects in stainless steel in our project as consideration for detection, namely - Pit defect, Edge Crack defect, Scratches defect, and Rolled In Scale defect. The data set comprises black & white pictures of Stainless Steel sheets that may or may not have desired defects. Hence data set in which abnormalities are present should be segregated first, resulting in an increase in the efficiency of the project by running the algorithm on images with defects.

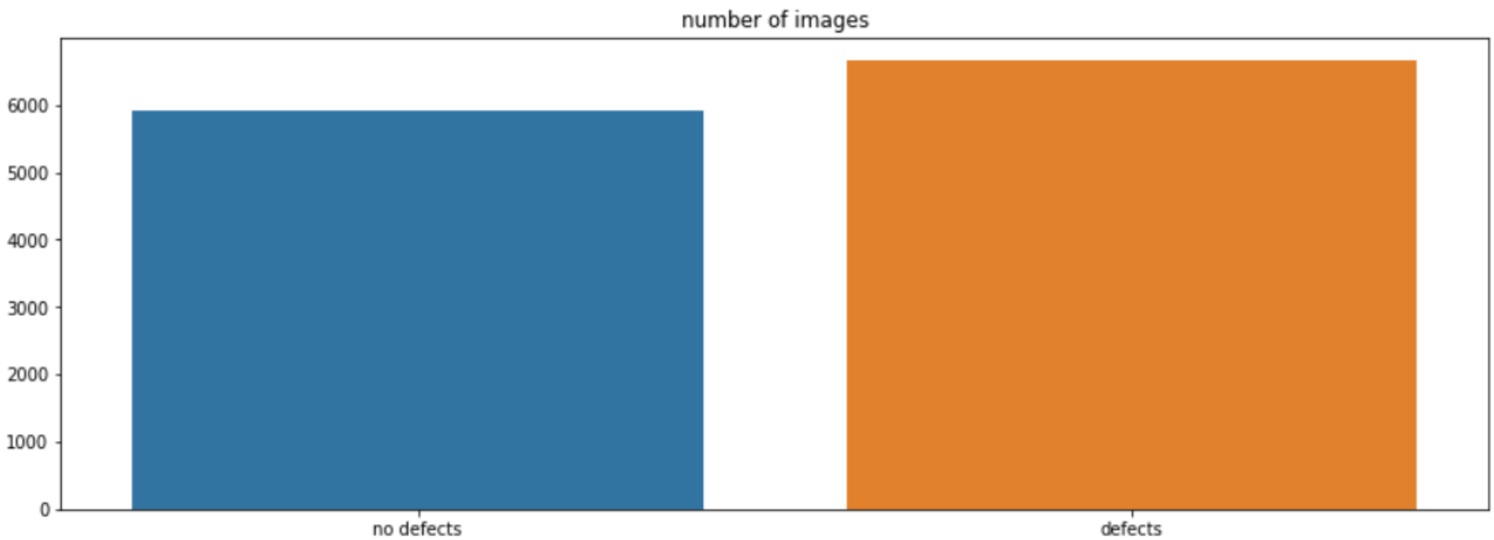


Fig. 2.1.1

The number of images with no defects is 5912 and the number of images with defects is 6666 respectively.
Hence the algorithm starts working on this prediction and considers the defective images first while detecting the classification of defects in them. More weight is given to the images with defects in them - which are 6666 here, rather than the images with no defects for predicting the defect classification in our project. Fig. 2.1.1 shows the data which is segregated using a bar graph plot here which consists of the number of images with no defects and with defects plotted here.

In the Fig. 2.1.2, pie diagram of the defects is plotted as shown and palettes containing the labels for each class are shown in the output for representation purpose wherein,

Class 1 defects comprises of 12.64%

Class 2 defects comprises of 3.84%

Class 3 defects comprises of 72.59%

Class 4 defects comprises of 11.29%

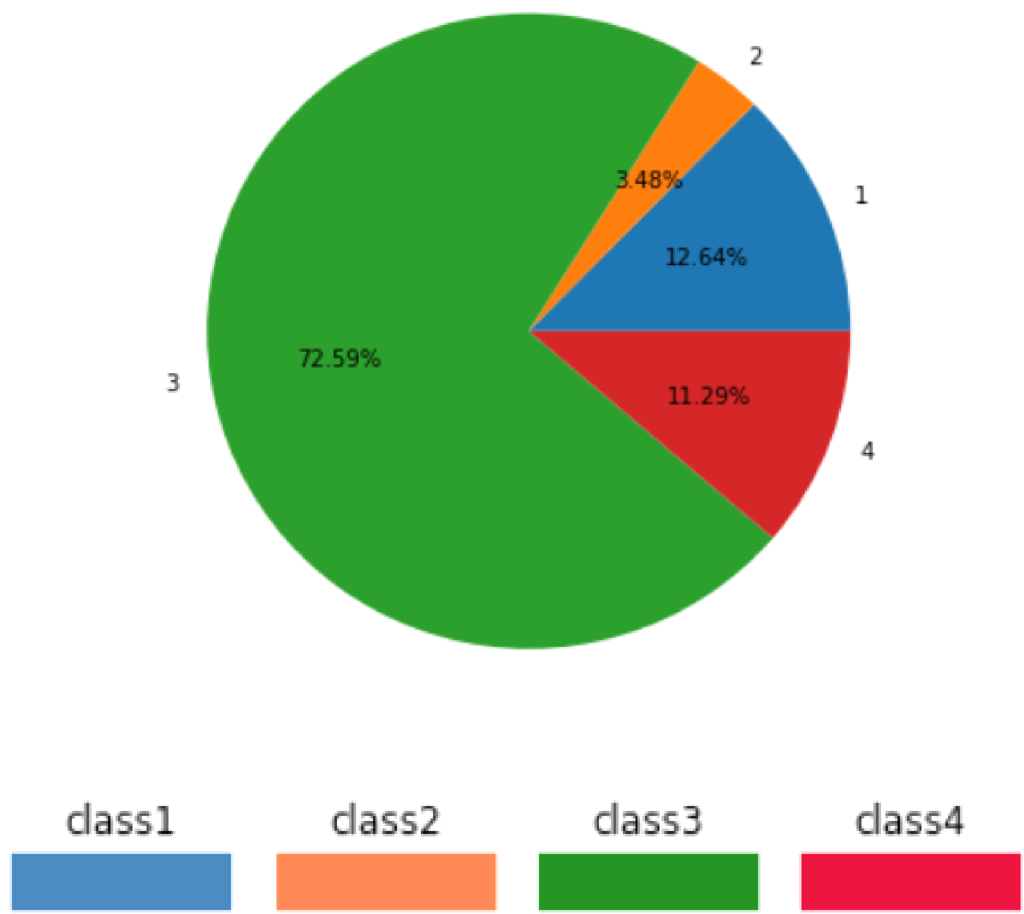


Fig. 2.1.2

2.2 Defects in Stainless Steel & Types

Surprisingly, the stainless steel's surface, which should be clean, smooth, and faultless, is what determines the "resistance" scale contained in the name. Corrosion resistance is dependent on this.
In reality, stainless steel is only protected against corrosion by a thin, impermeable surface layer known as the passive layer, which is mostly made of chromium oxide. The oxygen content in the atmosphere or oxygenated water solutions is often sufficient for the creation and maintenance of this passive layer. Surface flaws and imperfections caused by manufacturing processes, on the other hand, might sabotage the re-filming process and lower resistance to a range of localized corrosion.
As a result, cleaning is frequently required as the final stage in the stainless steel treatment process to restore acceptable surface quality.

Stainless steels are highly corrosion resistant, however surface damage can still occur in stainless steel applications. Without normal cleaning and maintenance, oxidation, corrosion, rusting, or discoloration can occur over time in hostile conditions. Repeated mechanical damage causes the metal to degrade more quickly. At least 10.5 percent chromium by weight is present in all stainless steels. Stainless steel, unlike other steels, is protected from corrosion by a layer called the passive layer, which is made up of chromium. The corrosion resistance increases as the chromium level rises. When the passive layer is broken and there isn't enough chromium for it to regenerate, stainless steel rusts.

Fig 2.1.3 highlights the defects considered in this project which are as follows :

A	Pit Defect	*(Class 1 defect)*
B	Edge Crack Defect	(*Class 2 defect*)
C	Scratches Defect	*(Class 3 defect)*
D	Rolled in Scale Defect	(*Class 4 defect)*

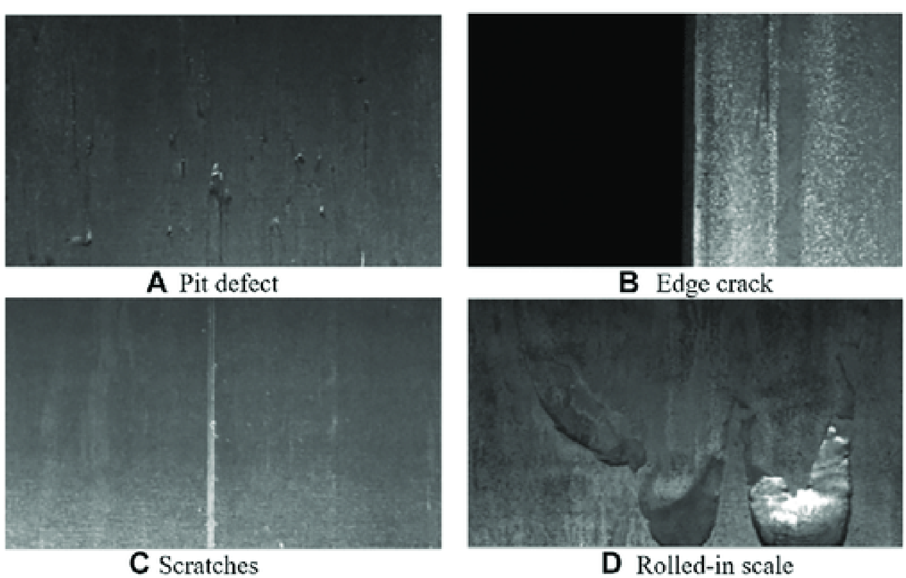


Fig. 2.1.3

2.2.1 Pit Defect

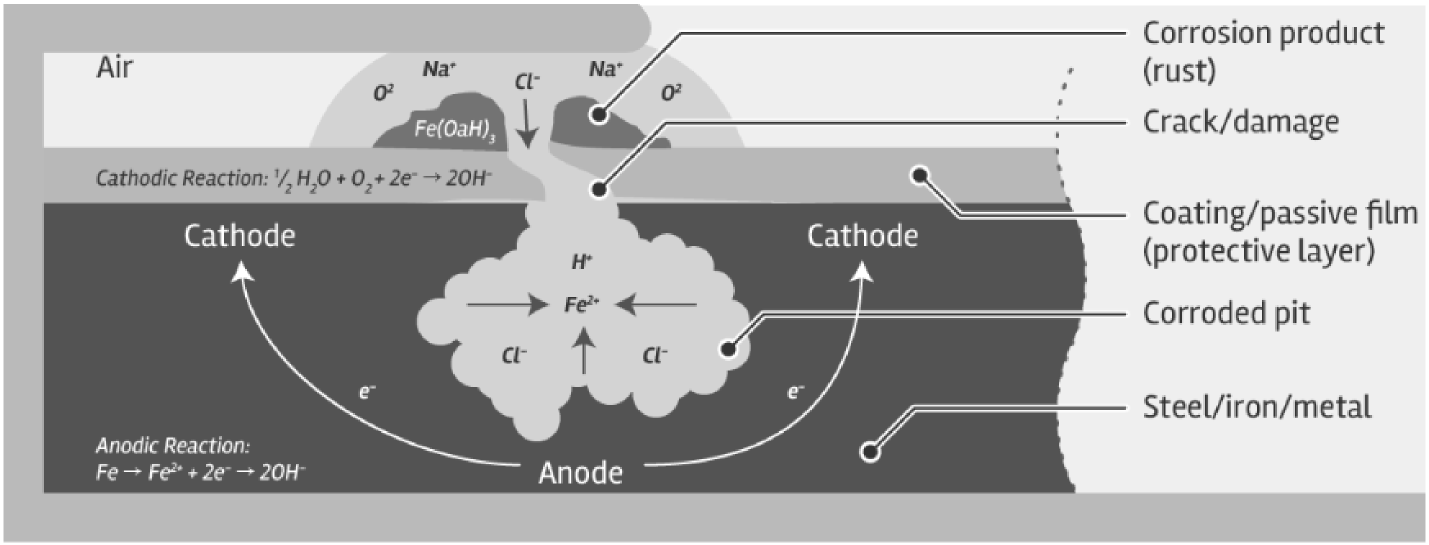


Fig. 2.1.4

What is Pitting Corrosion, and how does it happen?

Steel, iron, aluminum, and other metals and alloys are affected by pitting corrosion. It is usually limited to a specific geographical location. It is tough to detect since it penetrates and attacks quickly. It usually happens when the passive coating layer is physically or chemically degraded. This produces a weak area in the substrate where water or corrosive solutions can attack it.
Often, adjacent materials will appear to be unaffected. Pitting corrosion can be deadly to roof systems and any metal structure if left unchecked. Because it happens quickly and is readily overlooked, many people consider it to be the most harmful type of corrosion.

Pitting corrosion is the formation of a cavity, hole, or pit in a narrow area or point. A minor quantity of corrosion product (rust) on the surface hides the pits or holes. A pit, cavity, or small hole forms when a cathodic reaction in a large area (coating) is sustained by an anodic reaction in a small area (exposed metal). Even when there is no oxygen available, the metal oxidizes as shown in Fig. 2.1.4 respectively. The small anode is subjected to high electron demand from the huge cathode, resulting in severe pitting corrosion. It will be subtle and happen quickly, with devastating consequences. Only a small bit of rust can be seen on the surface, but damage is occurring deep within the metal structure.

When the cathode (damaged coating) is large and the anode (exposed metal) is small, pitting corrosion develops. When the surface protective layer or film is broken and cracked, it usually becomes the cathode. The anodic is then formed by exposing a tiny piece of metal. When the fluid on the metal surface contains chloride, hypochlorite, or bromide ions, pitting is severe. Fluorides and iodides-containing solutions are also hazardous. Pitting is also known to be aided by sulfides and water.

Pitting corrosion can also occur when a metal is neglected and exposed to water droplets and dust particles. The area beneath the droplet is under-oxygenated, whilst the surrounding portions are well-oxygenated. Differential aeration corrosion occurs as a result, with the surrounding portions becoming cathodic and the little area below the drops and dust particles becoming anodic. Water and oxygen meet the electrons as they travel through the metal. Rust is created when ions combine and defuse. As rust forms, pits, fissures, and crevices appear in the metal.

The following are the most typical causes of pitting corrosion:

Protective coating cracks
Scuffs, scratching, and small chips
Stress that isn't uniform
Metal substrate with flaws
Protective covering that isn't uniform
Protective coating is attacked by chemicals.

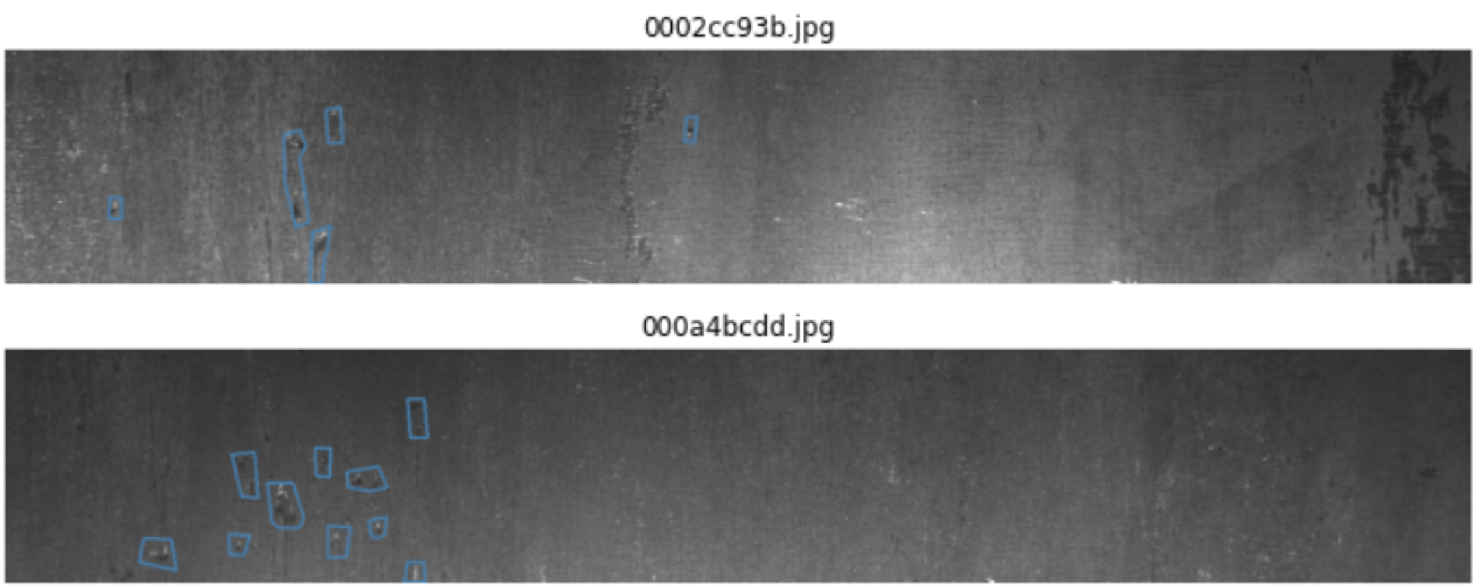


Fig. 2.1.5

Fig. 2.1.5 shows Class 1 defects annotated with Blue color contour on images of Stainless Steel sheets used in our project's training data set.

2.2.2 Edge Crack Defect

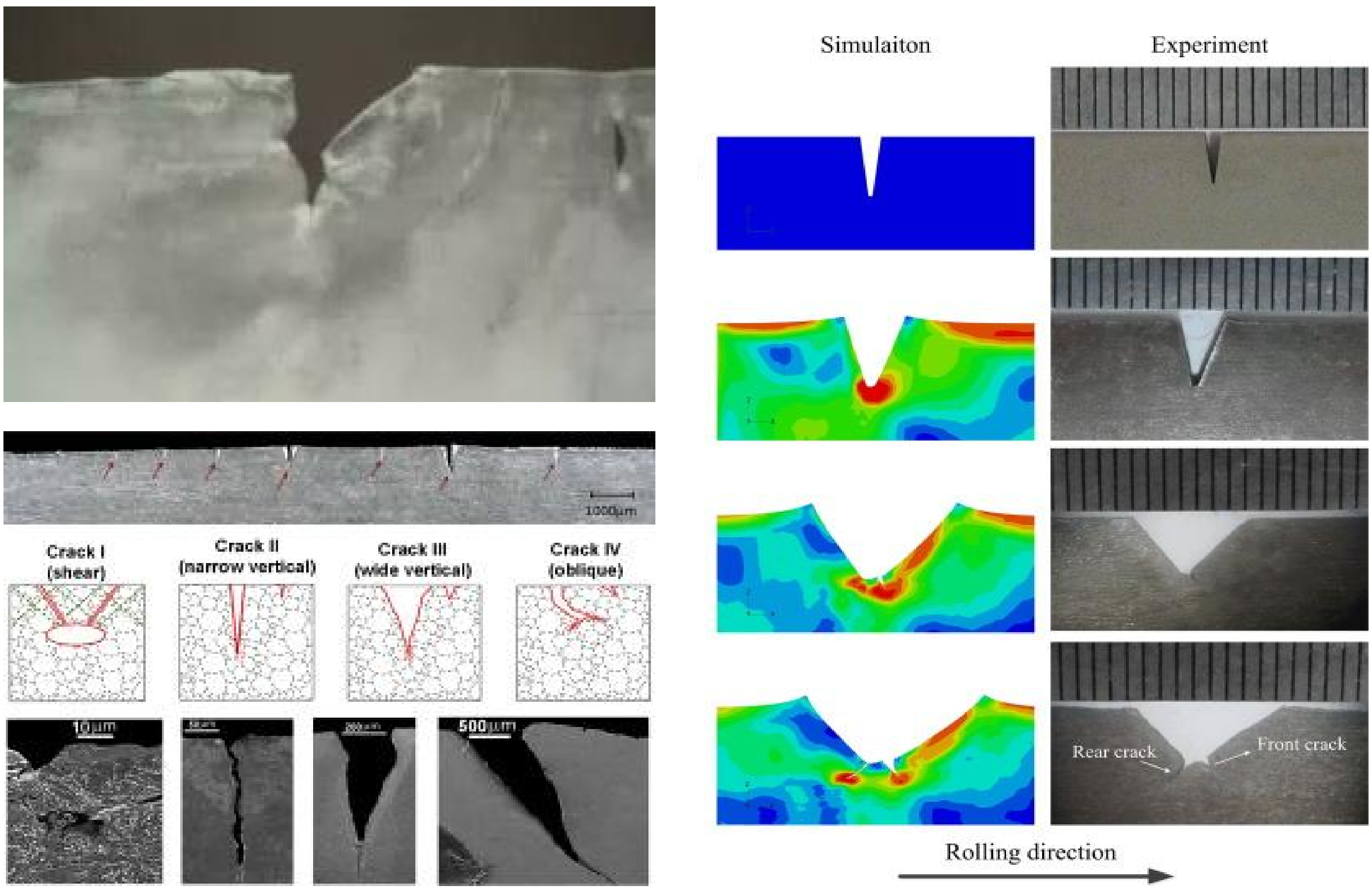


Fig. 2.1.6

During hot and cold rolling, many metals acquire edge cracks. Trimming is required to remove edge cracks, which might cause the sheet to rupture in the rolling mill. One of the fractures in strip rolling is edge cracking, which occurs when the strip's edge cracks in the width direction of the strip during rolling. The causes of edge cracking in cold rolling were reviewed by Dodd and Boddington (1980). The following is a summary of several studies on edge cracking in strip rolling.

Based on published data, Schey (1966) hypothesized that edge cracking was caused by three factors: the strip's limited ductility, unequal deformation at the edges (bulging or concave edges), and differences in stresses over the strip's width, particularly near the edges.

The influence of edge form on edge cracking during strip rolling was discovered by Cusminsky and Ellis (1967). Longitudinal strain at edge cracking is seen in Table 1.2. (Cusminsky and Ellis, 1967). The longitudinal strain at edge cracking increased with increasing the chamfer angle at the strip's edge in the breadth direction of the strip. The squared edge is implied by the chamfer's edge with an angle of 180 degrees. Cusminsky and Ellis computed the stress distribution in the central plane after measuring the strain distribution in the central plane in the thickness direction of the strip by riveting two strips together to produce a specimen. Edge cracking was said to be regulated by longitudinal stress, according to Cusminsky and Ellis.Edge cracking, according to Cusminsky and Ellis, is regulated by longitudinal stress; edge cracking occurs when the longitudinal stress at the strip's edge is equal to the stress calculated from the strip's stress–strain relationship. Edge cracking occurs when the stress distribution at edge cracking is nearly identical to the stress distribution in the tensile test, according to this agreement.

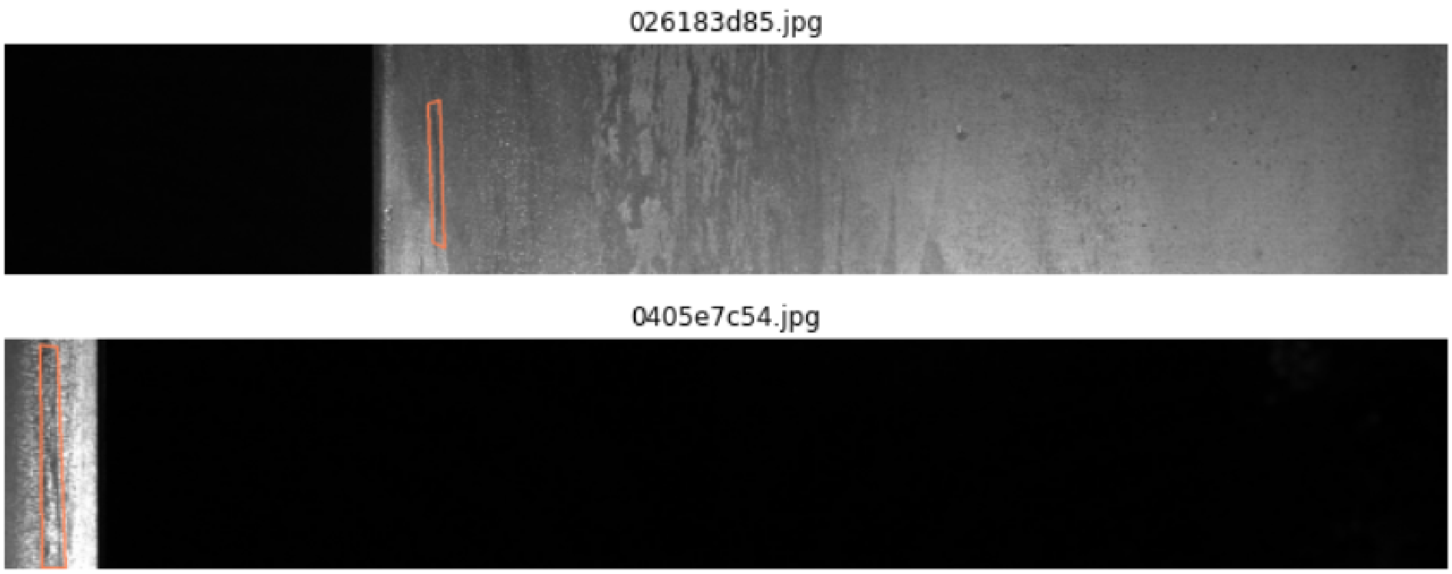


Fig. 2.1.7

Fig. 2.1.7 shows Class 2 defects annotated with Orange color contour on images of Stainless Steel sheets used in our project's training data set.

2.2.3 Scratches Defect

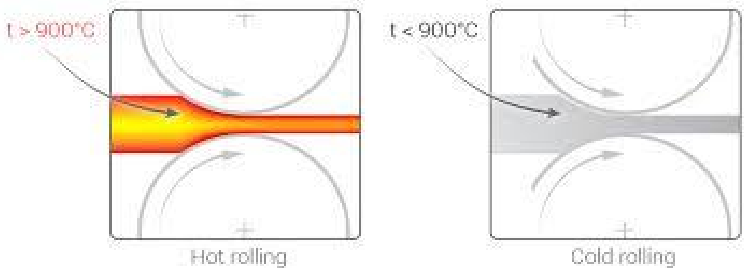


Fig. 2.1.8

Scratches are visible and result from unintended contact with buildup on mechanical parts and mill components while rolling. Scratches, unlike seams and cracks, have a more rounded bottom and less scale.Small, acute indentations to wide gouges with partially protruding edges are all possible. The metal displaced by mechanical works frequently appears to have an amorphous form. A Bielby layer is what this is known as.As steel is displaced into the depression during the cold drawing process, hot roll scratches might 'fill in'. The shining surface, lack of scale, and lack of decarb behind the displaced metal indicate that the area beneath the misplaced material was pickled or descaled, and that the displacement occurred during cold drawing and the scratch occurred prior to that operation.

This metal can then flake away (typically during straightening), causing the material to be labeled as having slivers. The correct identification is still a work in progress. A brilliant scratch on the cold drawn bar might be caused by a foreign object (wire) or material buildup in the cold drawing die.

While a scratch should be considered a "mechanical defect," most scratches do not go deeper than the minimal stock removal restrictions.
Scratches tend to appear on softer materials. Scratches rarely open up during upset or torsion tests because of their wider radius. Most mill hands, in my experience, will call any longitudinal flaw a seam, hence scratches are frequently mislabeled as seams.

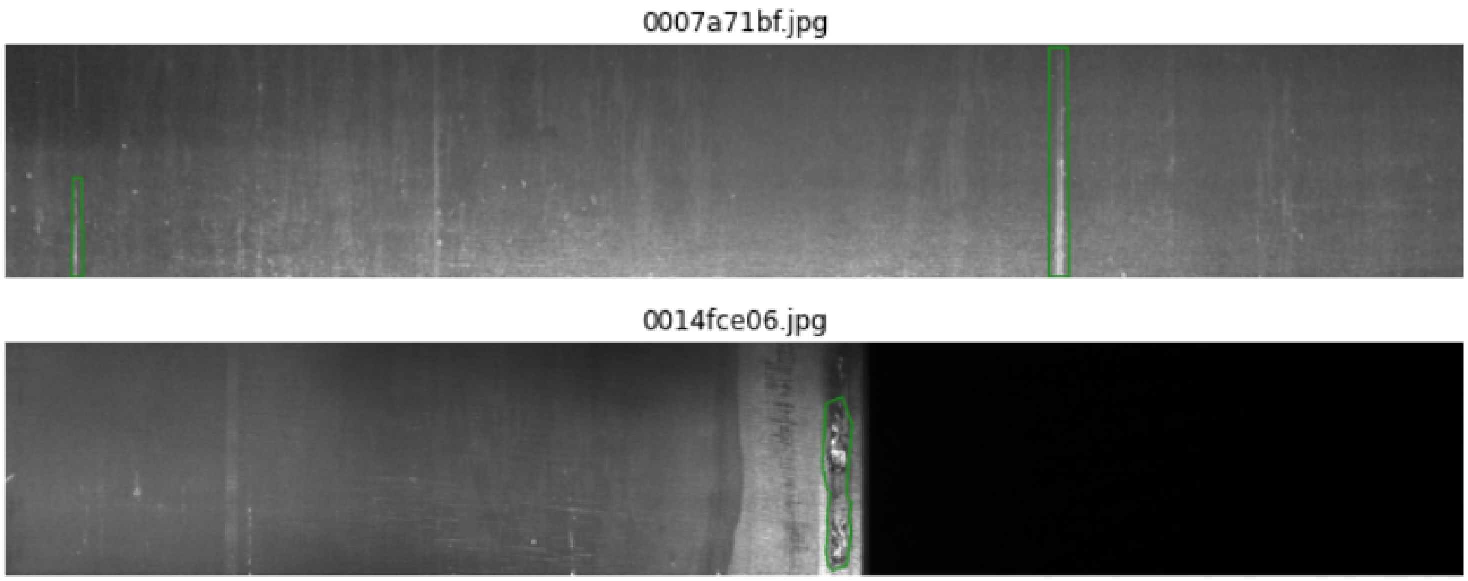


Fig. 2.1.9

Fig. 2.1.9a shows Class 3 defects annotated with Green color contour on images of Stainless Steel sheets used in our project’s training data set.

2.2.4 Rolled-in Scale Defect

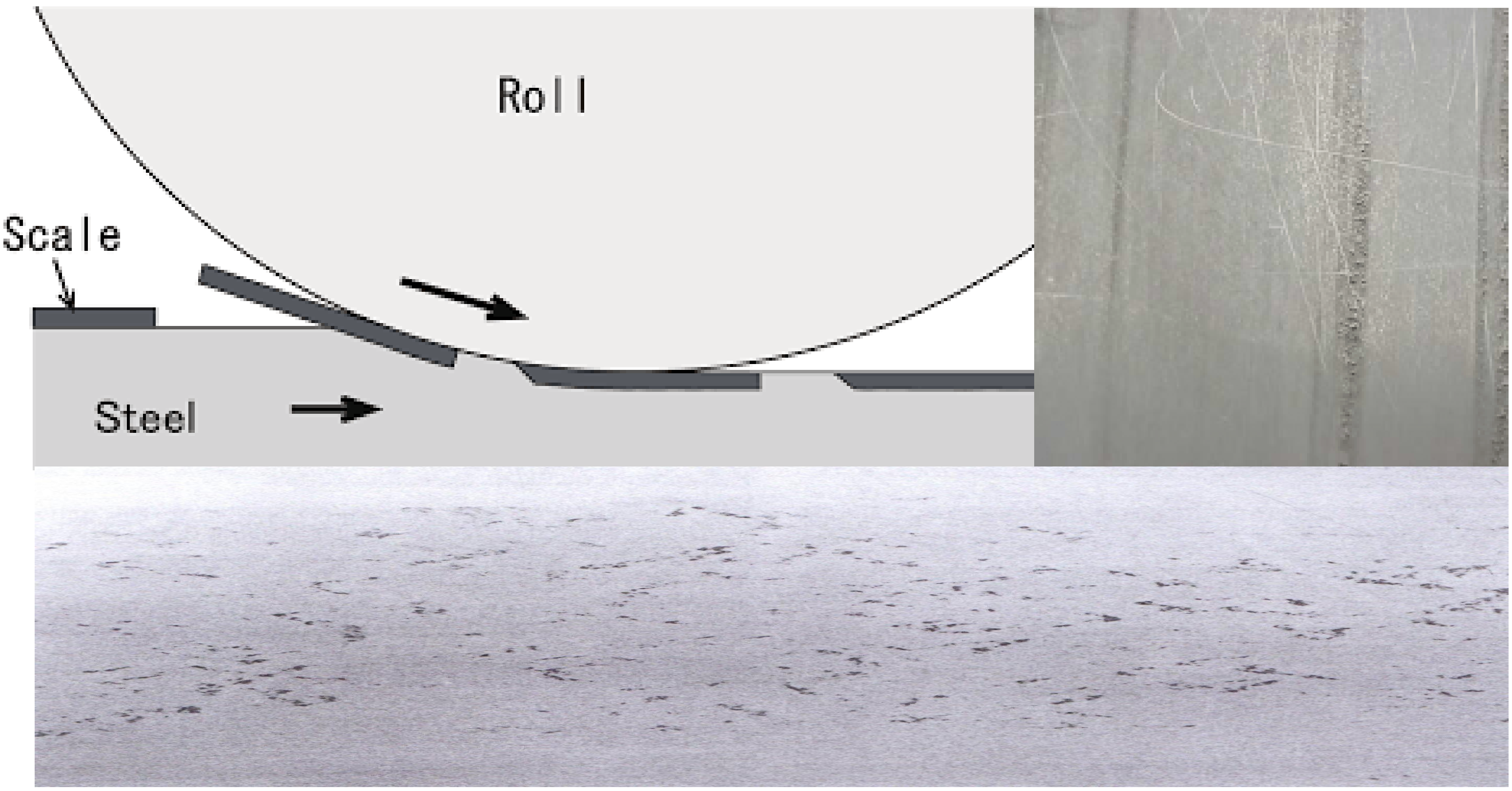


Fig. 2.1.9a

Mill scale is rolled into the metal during the rolling process, resulting in a rolled-in scale fault. The flaky mixture of iron oxides that forms on the surface of heated steel is known as mill scale.

Mill scale is typically found on raw steel and is mistaken for a blue-colored primer. Mill scale is an iron oxide that forms on the steel's surface during the hot-rolling process. The combination of a high surface temperature and high roller pressures produces a smooth, bluish gray surface. Mill scale is less reactive (or "noble") than the steel beneath it, and when two dissimilar metals come into contact, the more reactive metal (in this example steel) oxidizes (rusts) at the expense of the less reactive metal (in this case mill scale) (mill scale).The mill scale resembles a scale, and it has the ability to pop out of the surface, splitting the coating and enabling moisture to permeate.

This causes a "galvanic reaction," which causes pitting corrosion (rust) on the base steel. This will cause the corrosion under the scale to spread, causing further cracking in the covering and exposing more areas, resulting in more corrosion.This mill scale comes under rolling and causes Rolled-in Scale defect as shown in Fig. 2.1.9a

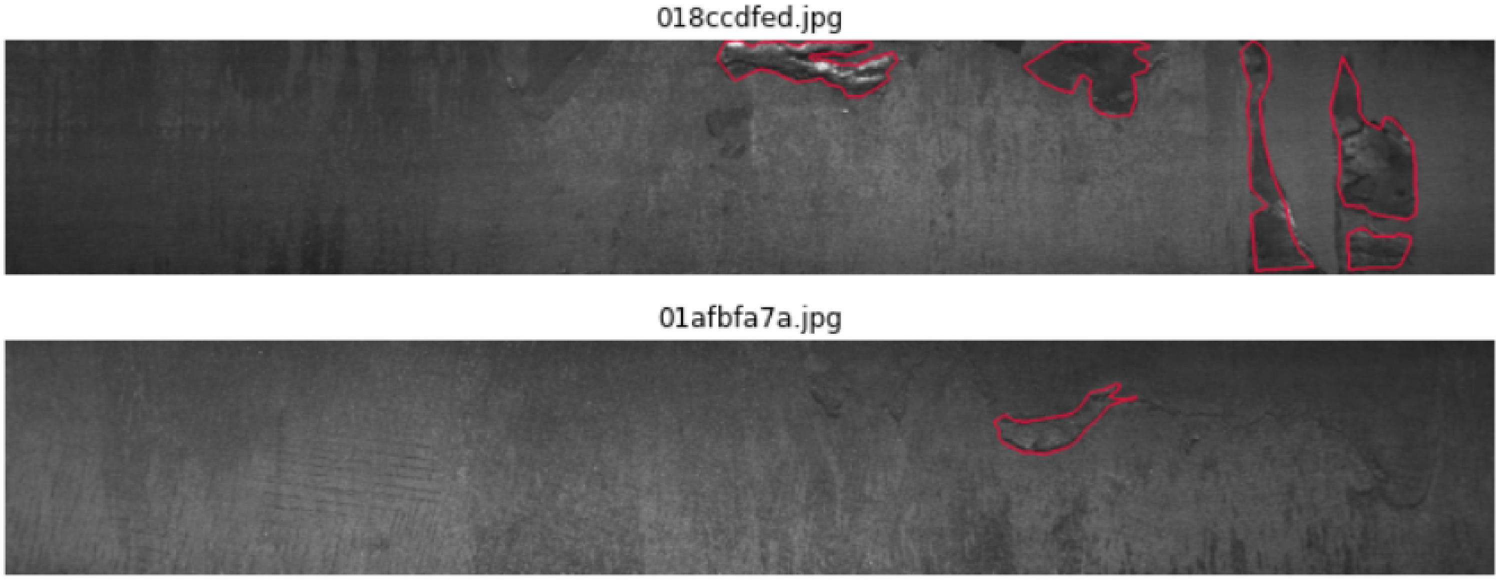


Fig. 2.1.9b

Fig. 2.1.9b shows Class 4 defects annotated with Red color contour on images of Stainless Steel sheets used in our project's training data set.

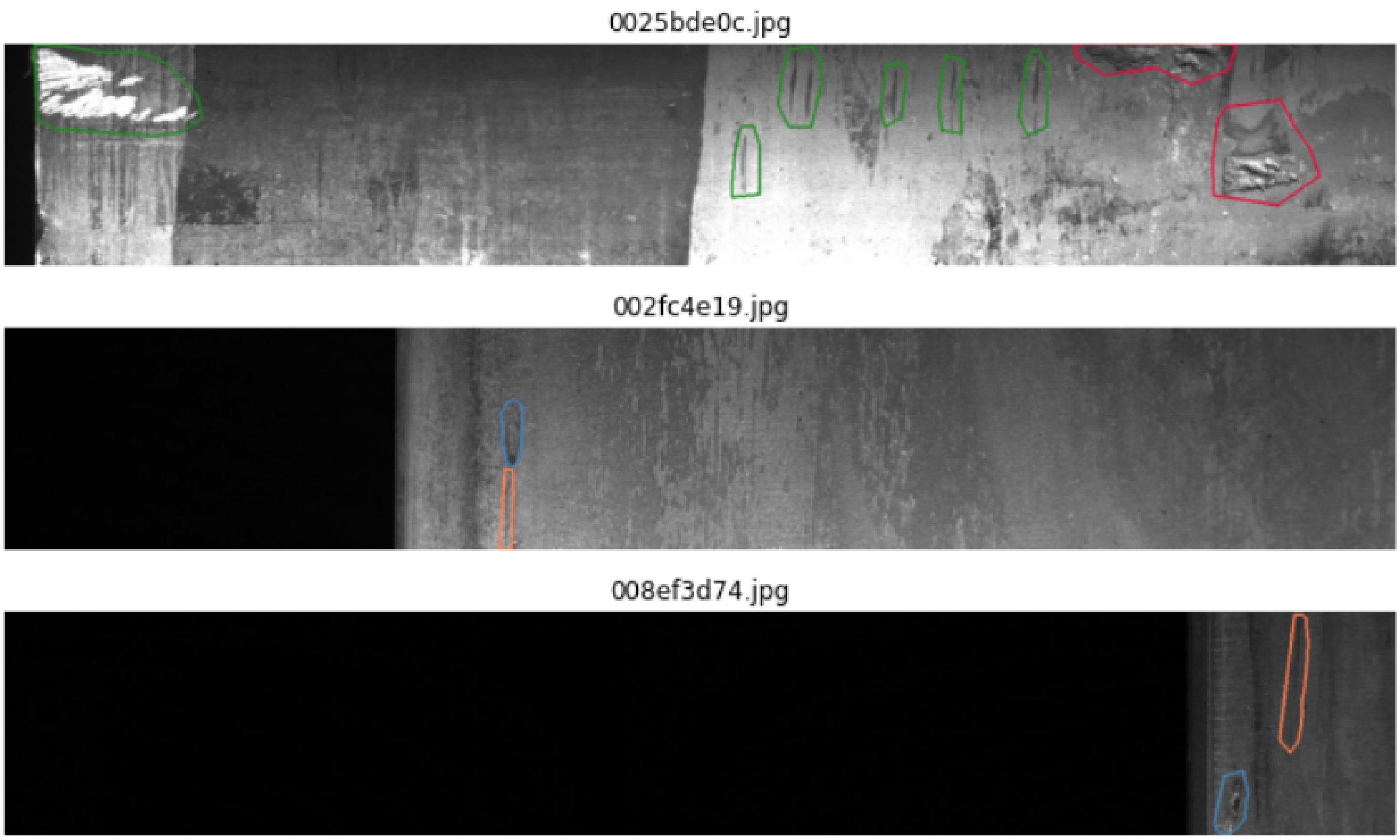


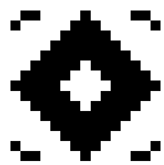
Fig. 2.1.9c shows Class 1 2 3 & 4 defects annotated with color contour on images of Stainless Steel sheets used in our project's training data set.

DATA PREPROCESSING

3.1 RUN LENGTH ENCODING AND IMAGE MASK

Run-length encoding (RLE) is a form of lossless data compression in which runs of data (sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most efficient on data that contains many such runs, for example, simple graphic images such as icons, line drawings, Conway's Game of Life, and animations. For files that do not have many runs, RLE could increase the file size.

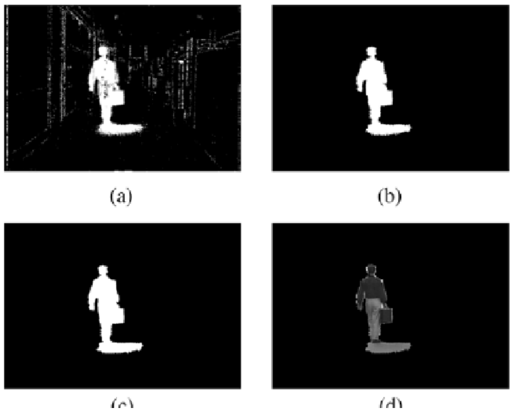
Imagine we have the following simple black and white image.



One very simple way a computer can store this image in binary is by using a format where '0' means white and '1' means black (this is a "bitmap", because we've mapped the pixels onto the values of bits). Using this method, the above image would be represented in the following way: 01101000110101

The main place that black and white scanned images are used now is on fax machines, which use this approach to compression. One reason that it works so well with scanned pages is that the number of consecutive white pixels is huge. In fact, there will be entire scanned lines that are nothing but white pixels. A typical fax page is 200 pixels across or more, so replacing 200 bits with one number is a big saving. The number itself can take a few bits to represent, and in some places on the scanned page only a few consecutive pixels are replaced with a number, but overall the saving is significant. In fact, fax machines would take 7 times longer to send pages if they didn't use compression.

Mask:
Masking is an image processing method in which we define a small 'image piece' and use it to modify a larger image. Masking is the process that is underneath many types of image processing, including edge detection, motion detection, and noise reduction. When there is more than one object in view in a scene at the same time, we need to figure out which pixels of the image belong to which objects. This process is called 'image segmentation'. We accomplish image segmentation by using a 'connected pixel' algorithm to find an 'object matrix', in which the value of each pixel tells which object that pixel belongs to.



3.2 RLE TO MASK

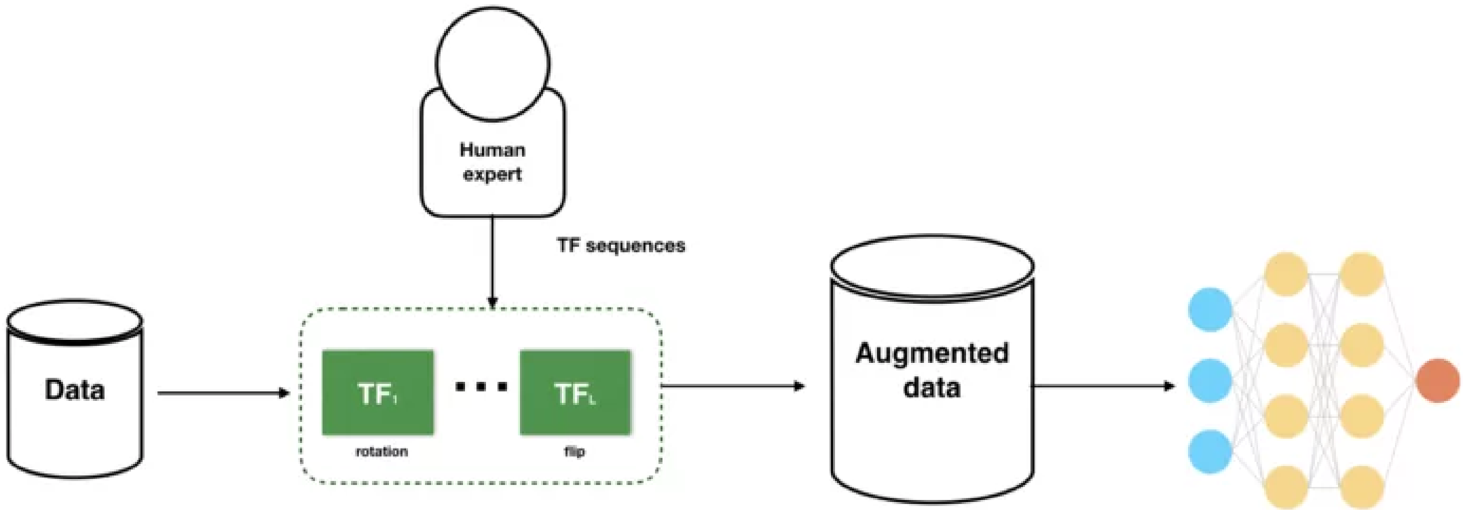
In our project the dataset contains encoded pixel values of image but we need to convert these encoded pixel values also known as RLE to mask and for that we use our very own rle2mask function which is a customized function that we created for this project in order to convert rle image to mask image. The height and width is specified in order to create a one dimensional array. This one dimensional array contains 0s of length obtained by multiplying the height and width of the image. After the 1-D mask is obtained, it’s dimension is converted to the dimensions of the image using the reshape function followed by transpose.

3.3 DATA AUGMENTATION

The performance of most ML models, and deep learning models in particular, depends on the quality, quantity and relevancy of training data. However, insufficient data is one of the most common challenges in implementing machine learning in the enterprise. This is because collecting such data can be costly and time-consuming in many cases. Companies can leverage data augmentation to reduce reliance on training data collection and preparation and to build more accurate machine learning models faster.

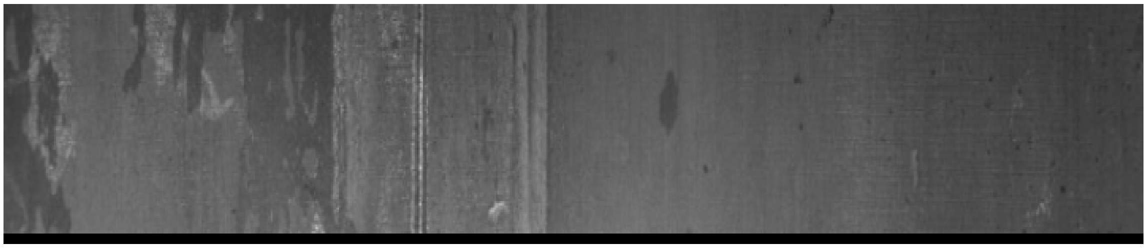
Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data. This includes making small changes to data or using deep learning models to generate new data points. Machine learning applications especially in the deep learning domain continue to diversify and increase rapidly. Data augmentation techniques may be a good tool against challenges which the artificial intelligence world faces. Data augmentation is useful to improve performance and outcomes of machine learning models by forming new and different examples to train datasets. If the dataset in a machine learning model is rich and sufficient, the model performs better and more accurately. For machine learning models, collecting and labeling of data can be exhausting and costly processes. Transformations in datasets by using data augmentation techniques allow companies to reduce these operational costs.

One of the steps into a data model is cleaning data which is necessary for high accuracy models. However, if cleaning reduces the representability of data, then the model cannot provide good predictions for real world inputs. Data augmentation techniques enable machine learning models to be more robust by creating variations that the model may see in the real world.

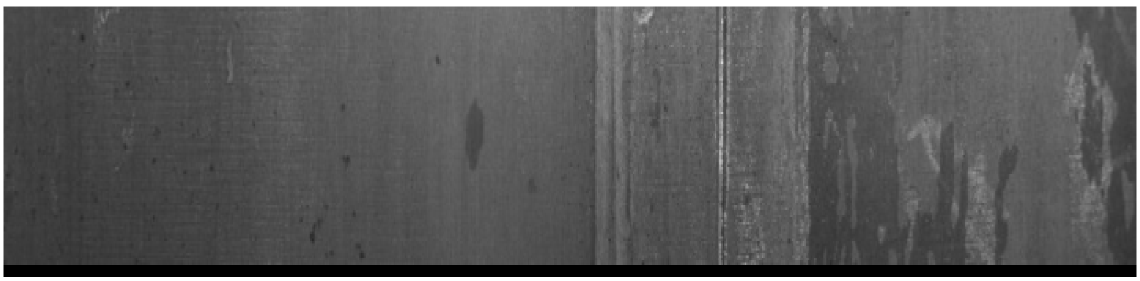


- Benefits of data augmentation include:
- Improving model prediction accuracy
    - adding more training data into the models
    - preventing data scarcity for better models
    - reducing data overfitting (i.e. an error in statistics, it means a function corresponds
    - too closely to a limited set of data points) and creating variability in data
    - increasing generalization ability of the models
    - helping resolve class imbalance issues in classification
- Reducing costs of collecting and labeling data
- Enables rare event prediction
- Prevents data privacy problems

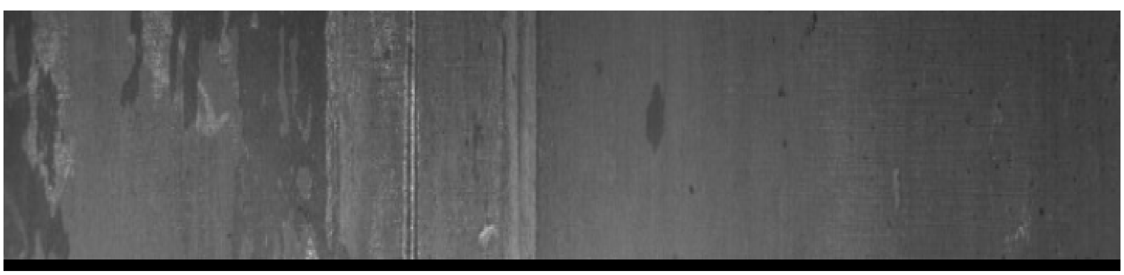
In the below images, First image represents our original image, in this original image two data augmentation techniques have been performed namely horizontal flip and vertical flip.



ORIGINAL IMAGE



HORIZONTAL FLIP



VERTICAL FLIP

LOSS FUNCTION

LOSS FUNCTION

Binary Cross Entropy :

Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value.

Dice Coefficient :

Dice coefficient, which is essentially a measure of overlap between two samples. This measure ranges from 0 to 1 where a Dice coefficient of 1 denotes perfect and complete overlap. It is a popular loss function for image segmentation tasks based on the Dice coefficient, which is essentially a measure of overlap between two samples. This measure ranges from 0 to 1 where a Dice coefficient of 1 denotes perfect and complete overlap. The Dice coefficient was originally developed for binary data, and can be calculated as:

$$Dice = \frac{2\,|A \cap B|}{|A| + |B|}$$

Where A = Predicted Value , B = Ground Truth Value

To calculate the dice coefficient, the predicted image Value and the ground truth value are first converted into arrays of 0's and 1's.

Intersection is calculated by performing element-wise multiplication of array .

Dice Loss = 1 - Dice coefficient.

We prefer the model with a lower loss value and hence the preferred value of Dice coefficient such be higher.

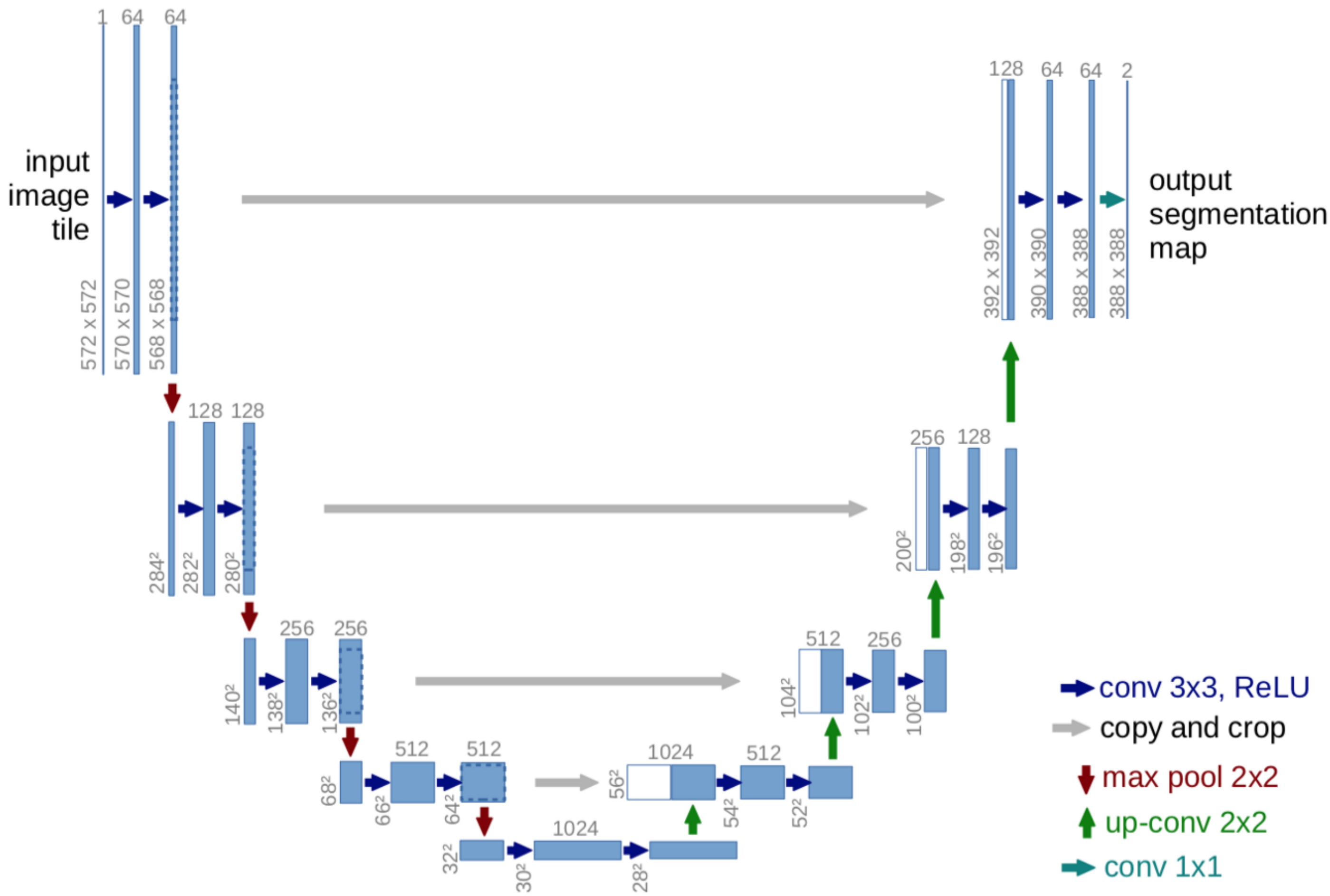
As per computation the value of dice coefficient for UNET segmentation is 0.3791 and for MobileNetV2 is 0.7562.

Which results in the loss value of 0.62 and 0.2438 respectively.

Hence MobileNetV2 is a better segmentation model for the given database and problem statement.

MODEL ARCHITECTURE

5.1 UNET



We use UNET for carrying out the image segmentation task. The reason why we use it is because it is able to localize and distinguish borders by doing classification on every pixel, so the input and output share the same size.

The architecture is symmetric and consists of two major parts — the left part is called contracting path, which is constituted by the general convolutional process; the right part is expansive path, which is constituted by transposed 2d convolutional layers.

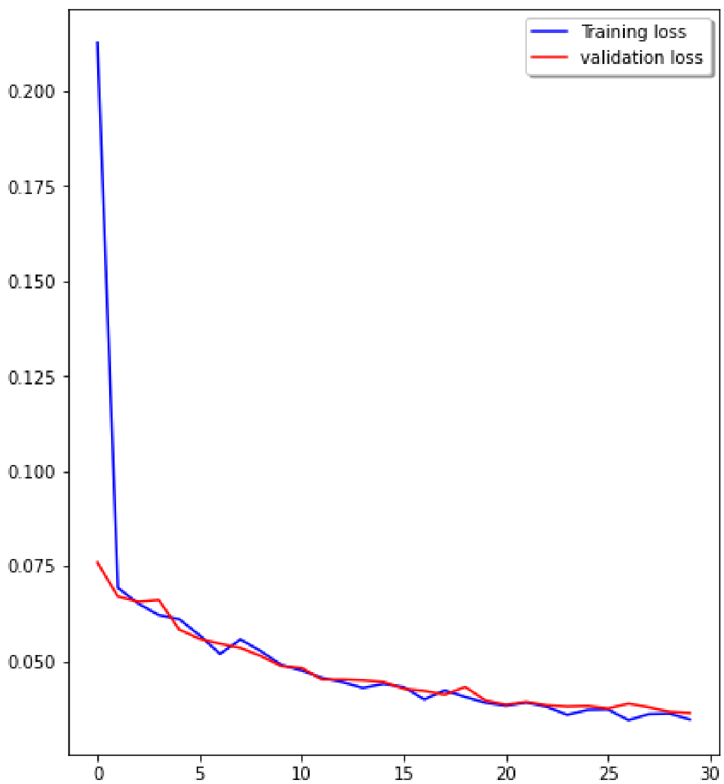
Contracting path

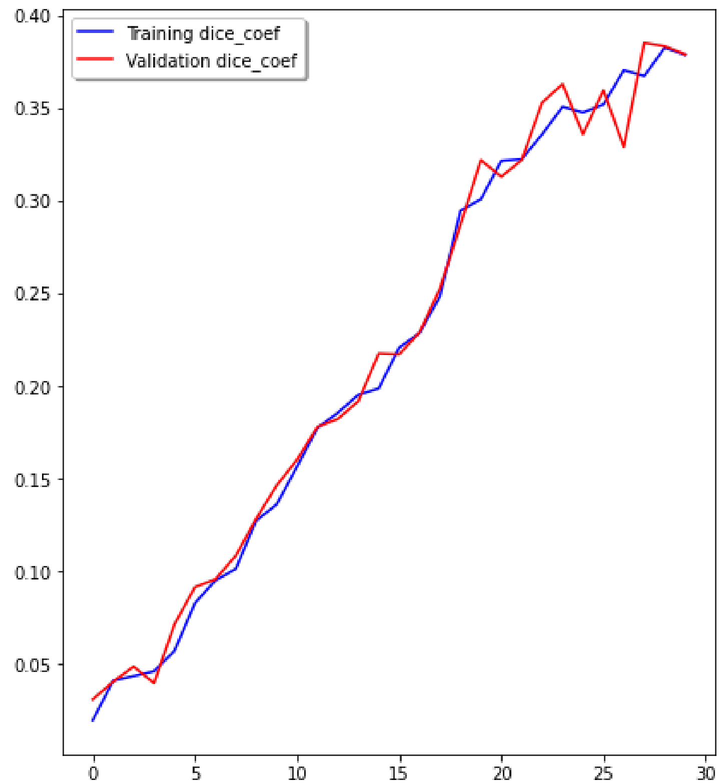
Each process constitutes two convolutional layers, and the number of channels changes from 1 → 64, as the convolution process will increase the depth of the image. The red arrow pointing down is the max pooling process which halves the size of the image. This entire process is further repeated three more times.

Expansive path

The image is going to be upsized to its original size. The transposed convolution is an upsampling technique that expands the size of images. After the transposed convolution, the image is upsized then this image is concatenated with corresponding image from the contracting path. The reason to combine the information from previous layers is to increase spatial information and to get more precise information.

Results





After training the data on UNET model, we get the following results:

Training dice coefficient is 38.46%
Validation dice coefficient is 36.33%

In order to achieve greater dice coefficient we use the Pre-trained MobileNet V2 Unet model.

5.2 PRE TRAINED MOBILE NET V2 UNET

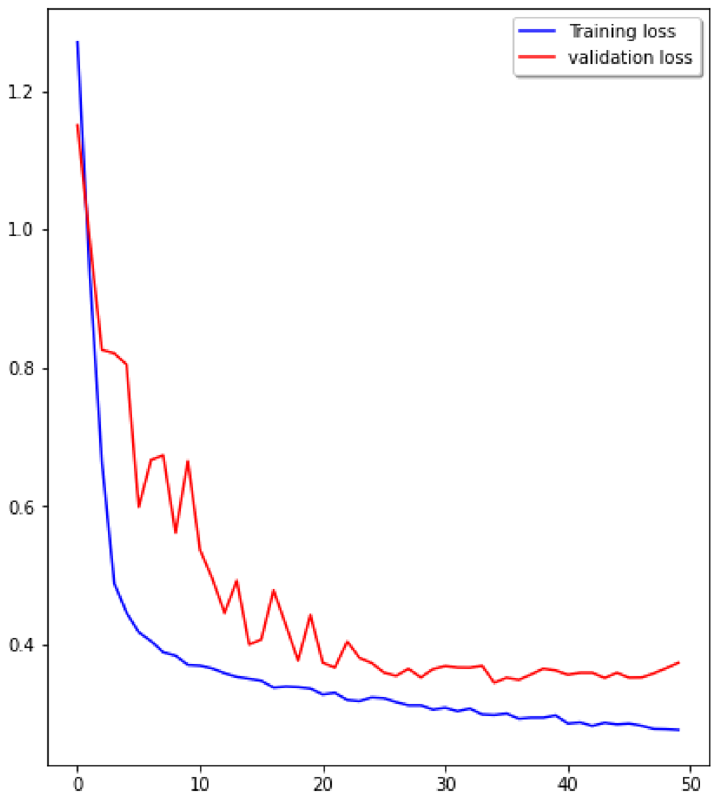
Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

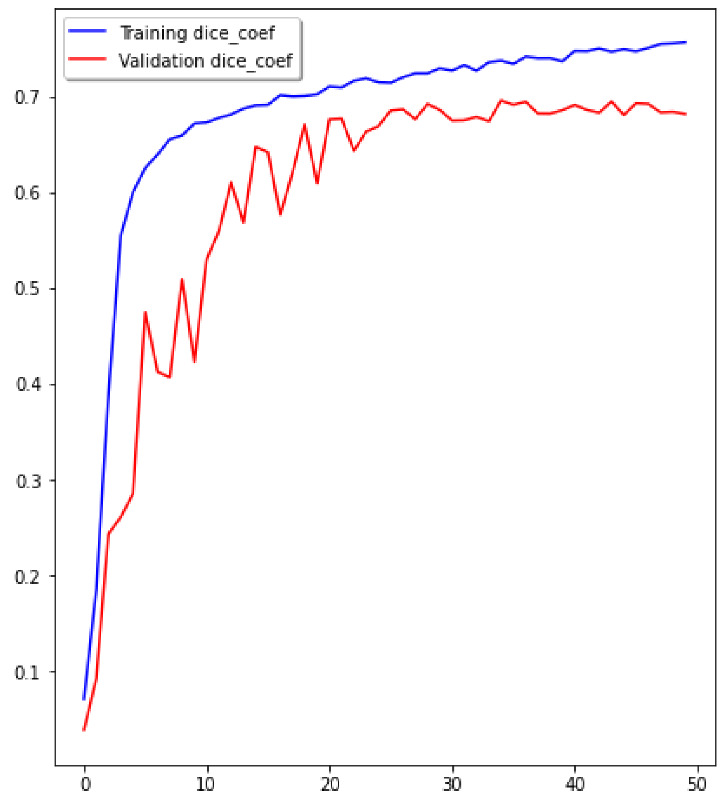
In this model we replace the encoder path with the pre-trained MobileNet V2 model which is trained on a large imagenet dataset.

Advantages of using pre trained encoder are:-

- MobileNet V2 has less parameters due to which it is easy to train.
- Using a pre-trained model, it converges much faster in comparison to a non pretrained model.
- A pre-trained encoder helps to achieve high performance as compared to non pretrained models.

Results





After training the data on Pre-trained MobileNet V2 UNET model, we get the following results:

Training dice coefficient is 79.54%
Validation dice coefficient is 68.89%

Callback Functions:

A callback is a function that can run after each epoch. It takes as arguments the epoch number and any metrics you have your model keeping track of. They can be used to do such useful things as scheduling reductions in the learning rate, saving the model between epochs.

Some of the callback functions we used are:

1) Model Checkpoint

   We initialize the class object with the file path to which to save, the conditions under which we want it saved, and how transparent the process should be. For example, let's say we want only the very best version of the model and we define 'best' as the one with the good validation dice coefficient . You can choose as a trigger either your loss function or any of the metrics you passed to the metrics argument when you compiled the model.

2) ReduceLRonPlateau

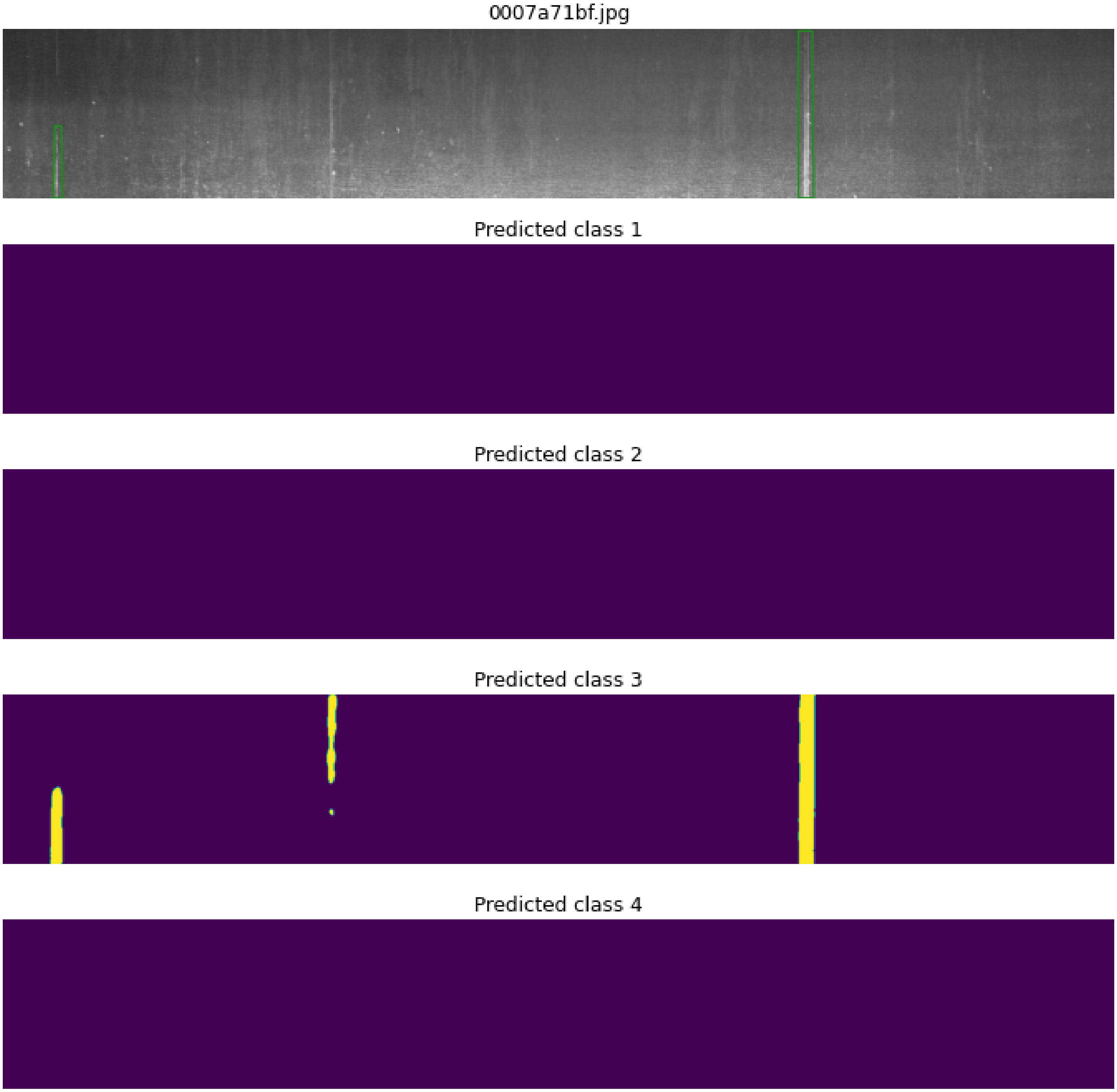
   If the initial learning rate is too large, it will cause oscillation. The initial learning rate is too small, resulting in slow convergence. The later learning rate is too large, it will cause overfitting. Therefore, in the training process, a dynamically changing learning rate is generally set according to the number of training rounds. The ideal strategy is to start with a large learning rate and gradually decay. In this project , the ReduceLROnPlateau method is used to dynamically update the learning rate, which is based on the number of epoch training times and some measurement values (loss, accuracy, etc.) to dynamically decrease the learning rate.

Optimizer:

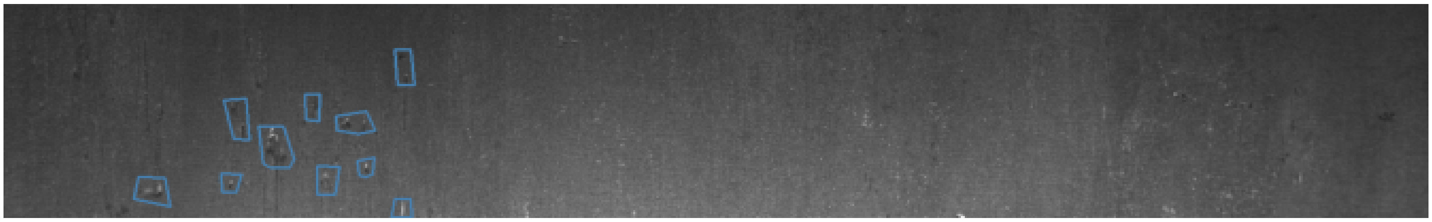
Adaptive Moment Estimation (Adam) is an optimizer that converges quickly and is often used. Adam uses the first-order moment estimation and the second-order moment estimation of the gradient to dynamically adjust the learning rate. It is an optimization method of adaptive learning rate. In this project we used Adamax optimization method which is a variant of Adam optimizer to continuously optimize the learning rate.

OUTPUT

6.1 PREDICTIONS OF UNET



000a4bcdd.jpg

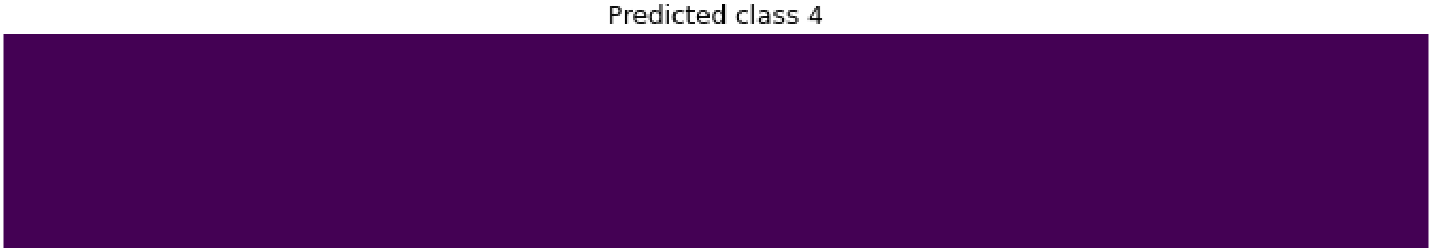


Predicted class 1

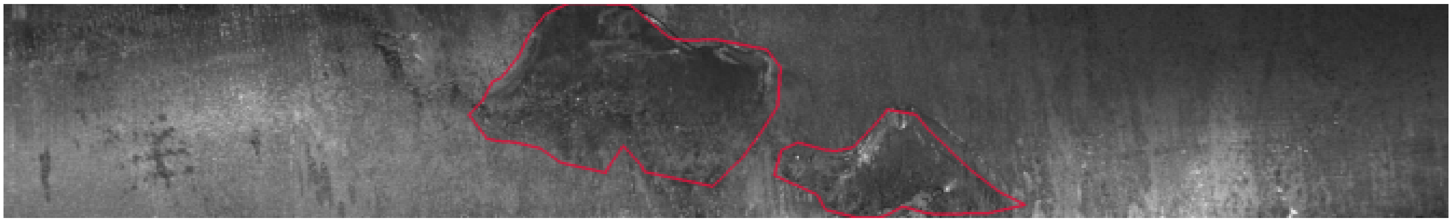
Predicted class 2

Predicted class 3

Predicted class 4



000f6bf48.jpg

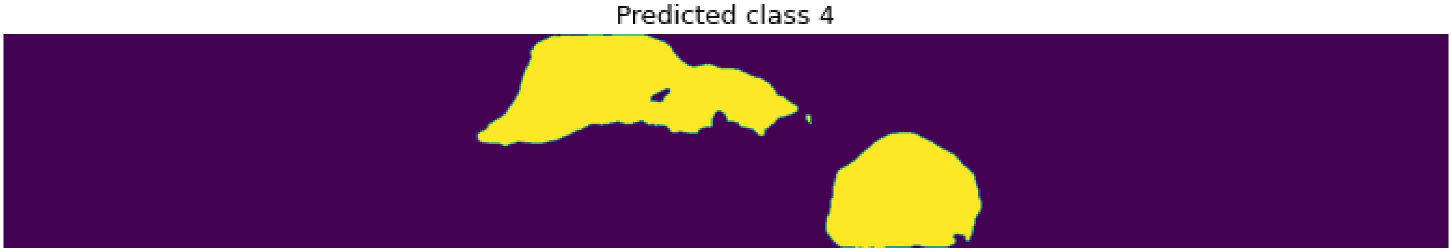


Predicted class 1

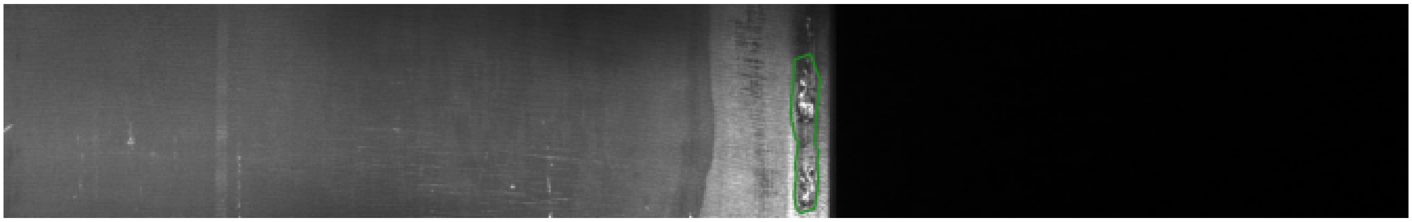
Predicted class 2

Predicted class 3

Predicted class 4



0014fce06.jpg

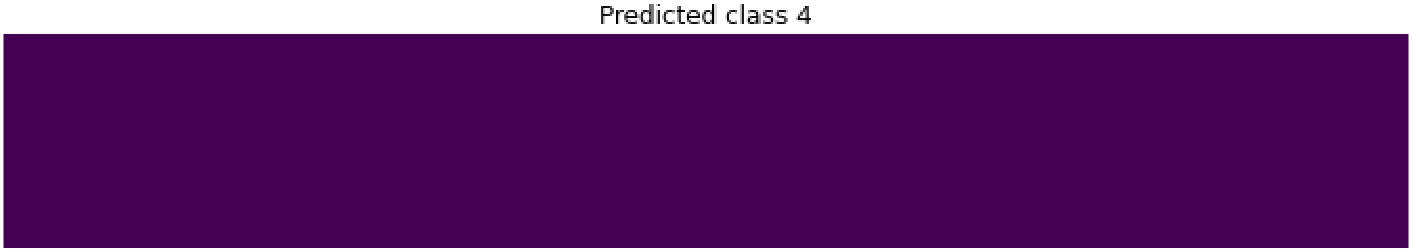


Predicted class 1

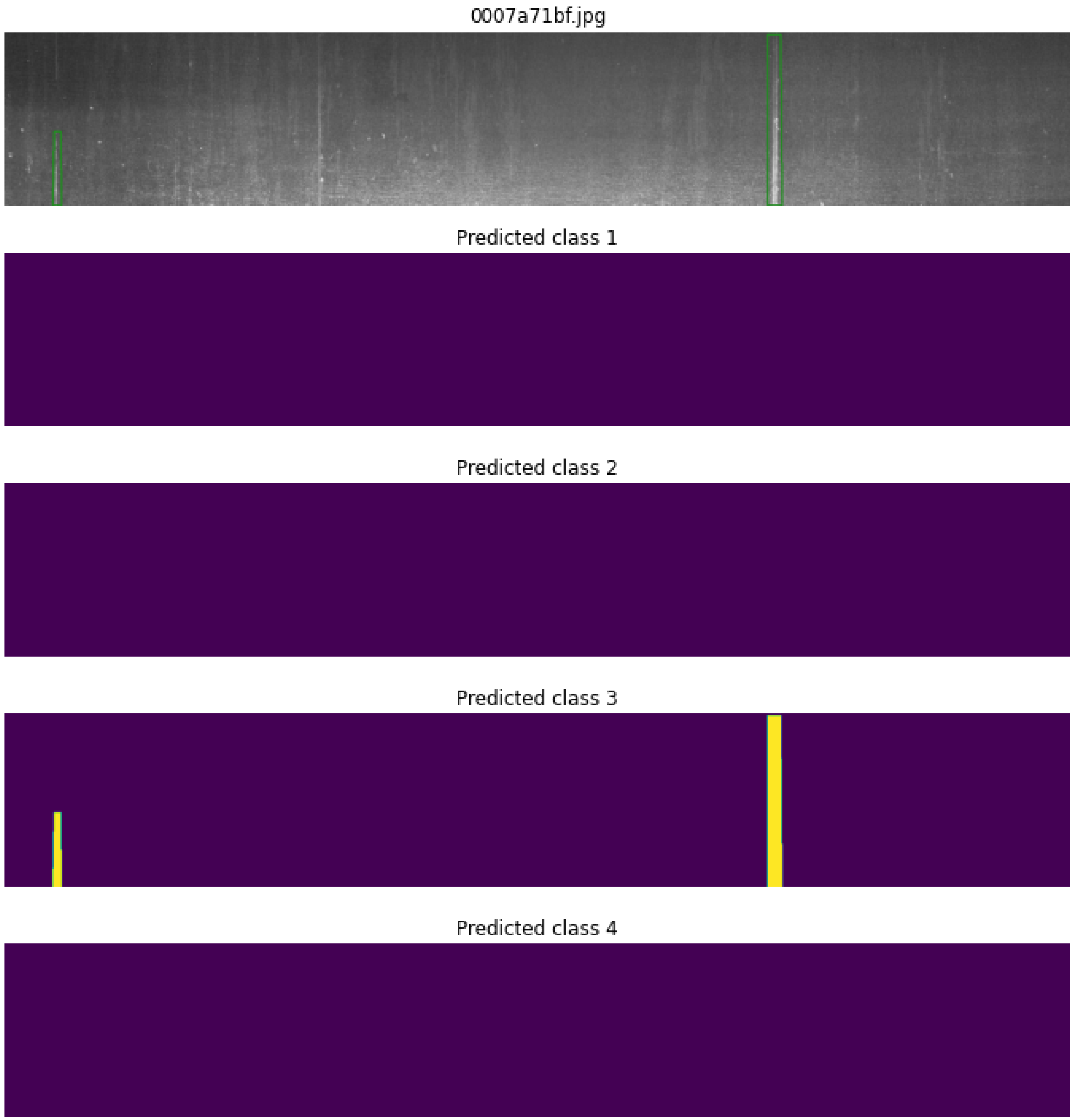
Predicted class 2

Predicted class 3

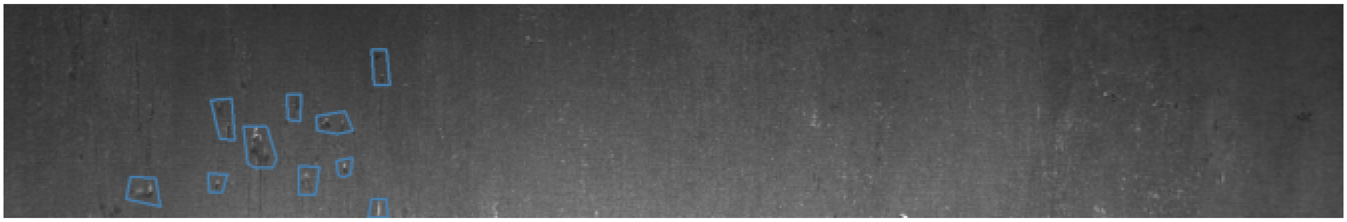
Predicted class 4



6.2 PREDICTIONS OF PRE TRAINED MOBILENET UNET



000a4bcdd.jpg

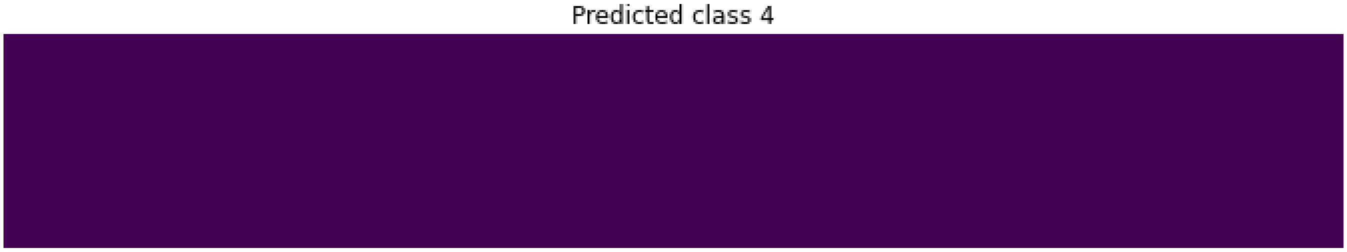


Predicted class 1

Predicted class 2

Predicted class 3

Predicted class 4



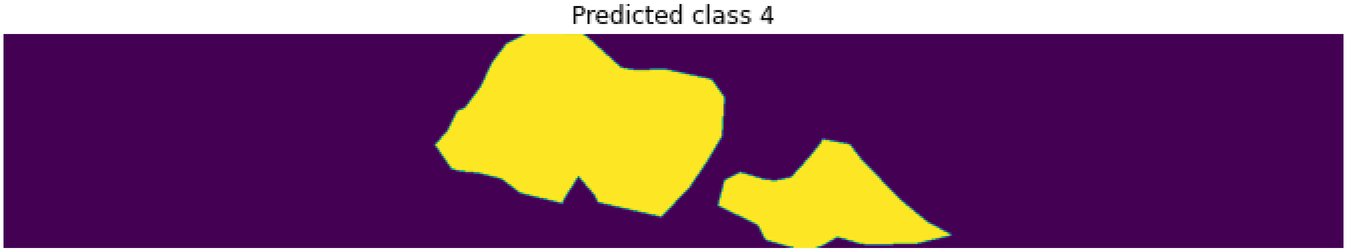
000f6bf48.jpg

Predicted class 1

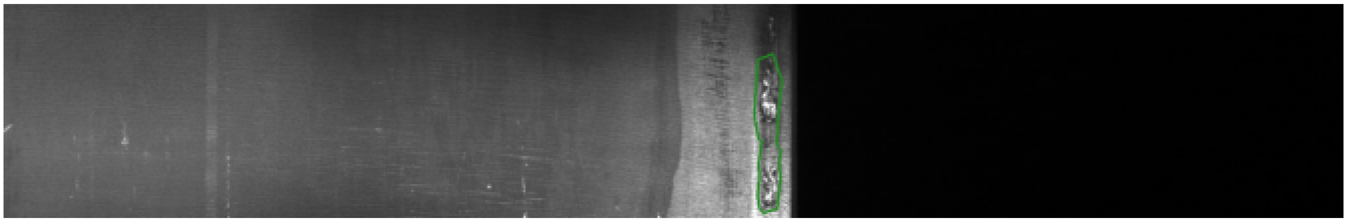
Predicted class 2

Predicted class 3

Predicted class 4



0014fce06.jpg

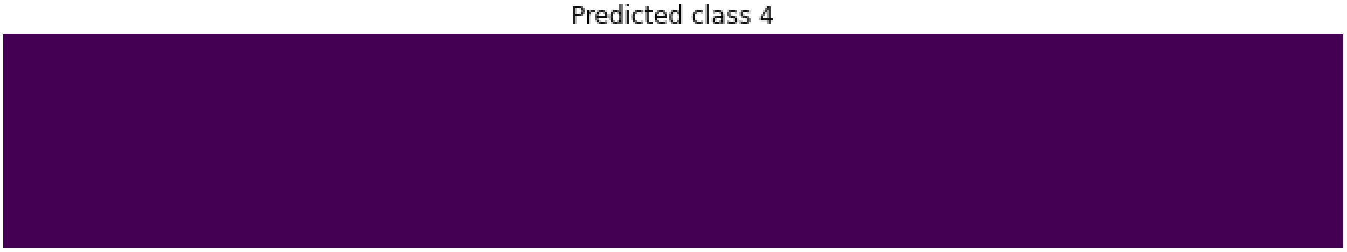


Predicted class 1

Predicted class 2

Predicted class 3

Predicted class 4



BIBLIOGRAPHY

1) Adaptive Recognition Of Surface Defects On Steel Sheet Using Transfer Learning
Jingwen Fu , Xiaoyan Zhu∗ , and Yingbin Li Xian Jiaotong University, Shaanxi , China

2) Deep Learning Based Defect detection System in Steel sheet Surfaces

3) Didarul Amin AISIP Lab, Dept. Of Computer Science and Engineering International University of Business Agriculture and Technology Dhaka, Bangladesh didarul38@gmail.com

4) Shamim Akhter AISIP Lab, Dept. of Computer Science and Engineering International University of Business Agriculture and Technology Dhaka, Bangladesh

5) Intelligent Detection of Steel Defects Based on Improved Split Attention Networks

6) https://towardsdatascience.com/u-net-b229b32b4a71