**Surface Defect Detection and localization**

**Using Deep learning(in hot-rolled steel strips)**

A Design Credit Project Report

Submitted by

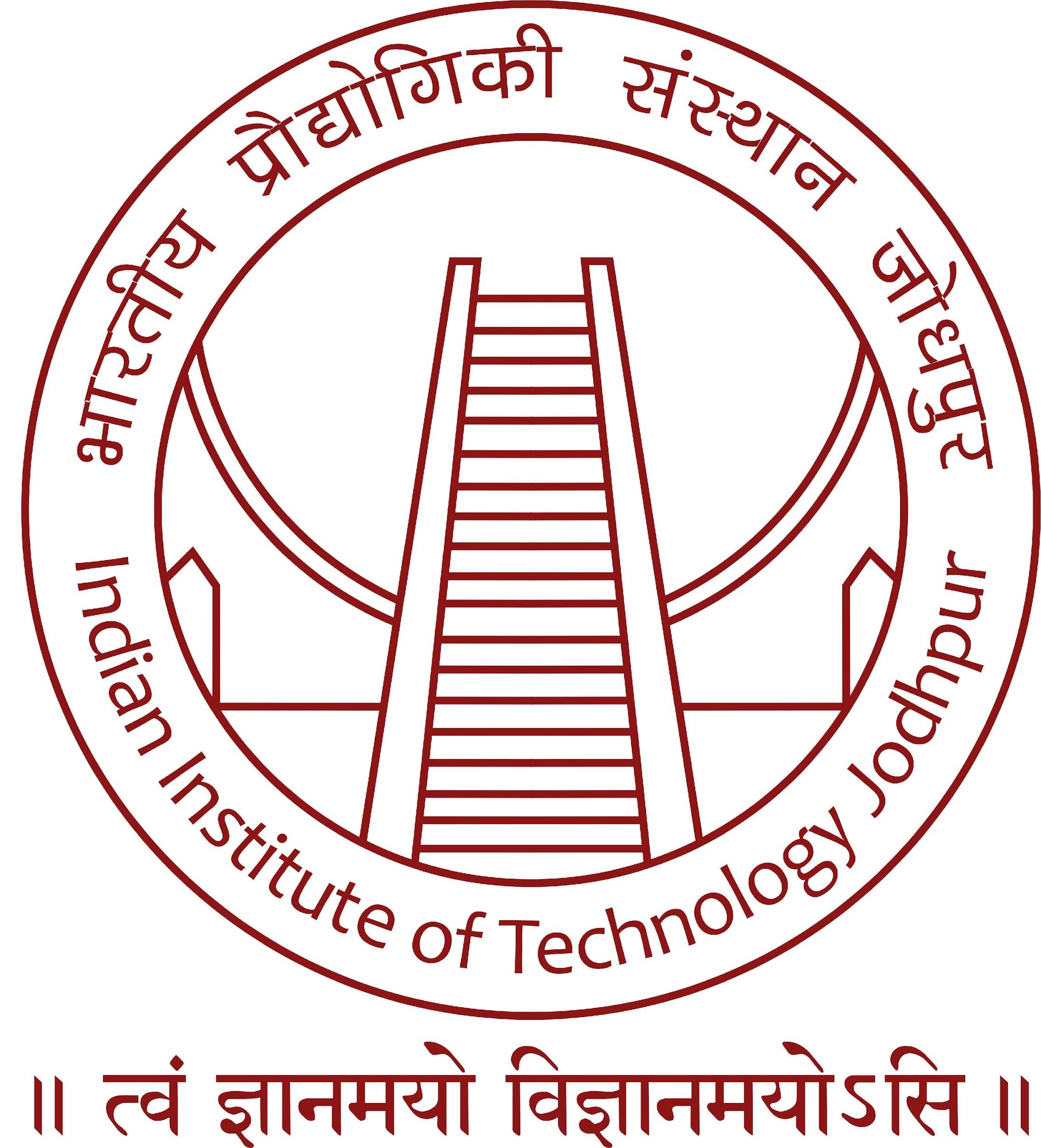
**Rishav Kumar(B19ME066)**

**Satyam Kumar (B19ME076)**

Under the Supervision

Of

**Dr. Binod Kumar**



Department of Mechanical Engineering

Indian Institute of Technology Jodhpur

January 2022

CONTENTS

Acknowledgements

Abstract

1. Introduction

1.1 Objective

1.2 Introduction to the problem

2. Work done

2.1 Methodology

1. Dataset
2. Data Analysis

2.2 Theoretical/Experimental work

1. Data Preprocessing
2. Convolution Neural Network(CNN)
3. Transfer Learning

IV. Feature Pyramid Network(FPN)

V. U-Net

VI. YOLO

2.3 Implementation

1. Classification Models
2. Localization Models

3. Results

3.1 Results for different classification models

3.2 Results for different Localisation models

4. Conclusion

References

**Acknowledgements**

We are grateful to Dr. B. Ravindra sir for enlightening us about the numerous mechanical domain challenges and for his guidance on how to solve them by implementing embedded technology and machine learning.

He has been incredibly encouraging and supportive of the idea, which has enabled us to finish this project.

During a number of productive sessions, we looked at a variety of approaches to the problem while keeping the demands and practicality of the concepts in mind. He also exhorts us to take part in a number of sessions that are useful to the project and are pertinent to it. For us, this was an excellent opportunity to learn.

Finally, we want to thank the mechanical engineering department for giving us the opportunity to work with cutting-edge technology, which, in our opinion, is crucial for a student in any area.

**Abstract**

It remains difficult to use defect detection in practice since a full task seeks to determine the precise class and position of each fault in an image. The task of defect detection involves both categorization and localization, making it challenging for related algorithms to consider the accuracy of both. A specialized detection data set with pricey hand annotations is required for fault detection implementation. In this project we have implemented three different methods for classifying the defect into one of the six classes and also find out the location of the defects by using four different methods. For both the task we have used deep learning techniques for automation of the task. The dataset used for the project is **NEU-DET** for training and evaluating our method. We have created individual models for both the tasks - classification and localisation.

1. **Introduction**

**1.1 Objectives:**

* Automate repetitive processes to improve productivity and efficiency by automating the quality control process in industries.
* To design a Deep learning model for classifying the image into different classes of defects.
* To design a model which can identify defect locations on the steel surfaces.

**1.2 Introduction to the problem**

With the ongoing development of the production requirements for contemporary industrial goods, the detection of surface flaws on machined components has emerged as a powerful method and tool for product quality inspection and control. Companies must quickly implement industrialized automation of production due to the variety and volume of industrial components. Hot rolled steel surface flaws pose potentially major issues during the rolling process since they might lead to mill shutdowns and the rejection of the rolled products.

Hot rolled steel is used in a variety of industries, including JSW Steels and LINXU Materials, and is available in a number of grades, each of which has its own distinct properties. It is economical, durable, fast to make, and ductile. This makes hot rolled steel perfect for heavy-duty industrial uses including mining, agriculture, and building infrastructure and bridges. The product must be defect-free to be used in these high-risk applications, but during the process, variables like rolling equipment performance decline and production process fluctuations may result in different types of surface defects showing up on the surfaces of hot-rolled strips, which will also lead to financial losses. These factors make the need for quality control and surface flaw identification in industries critical.

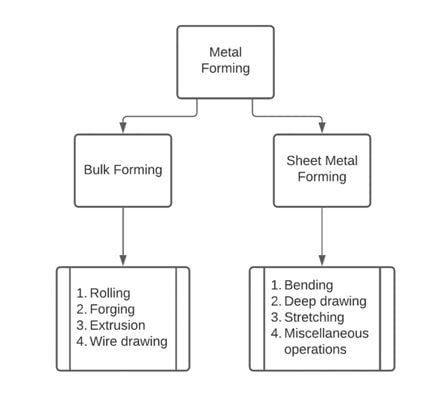
**Why is there a need for automation?**

The old methods of manual detection and conventional pattern recognition heavily rely on the operators' or algorithms' skill in parameter-setting. These techniques frequently work well for strip steel examination under particular circumstances. The **detection will not work if the external environment somewhat alters, such as an increase in noise or a change in lighting intensity**. Additionally, many businesses have established operations in several places, and the data at each site varies. This type of circumstance also necessitates the use of an inspection model that can hold up in the presence of some noise in the data. A **deep learning-based solution** has been suggested to automatically detect faults in steel surfaces, **replacing manual inspection,** as a result of deep learning's **improved performance in many visual tasks in recent years.**

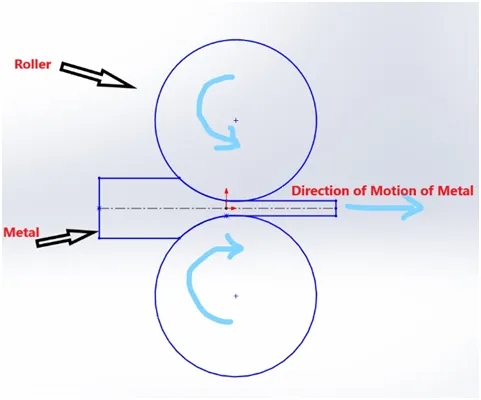
**2. Work done**

**2.1** **Theoretical work for problem**

**Metal forming** is a process of manufacturing components of desired shapes by deforming the material plastically, by the application of compressive force, bending or shear force, tensile force, or combinations of these all forces together, without adding or removing material.

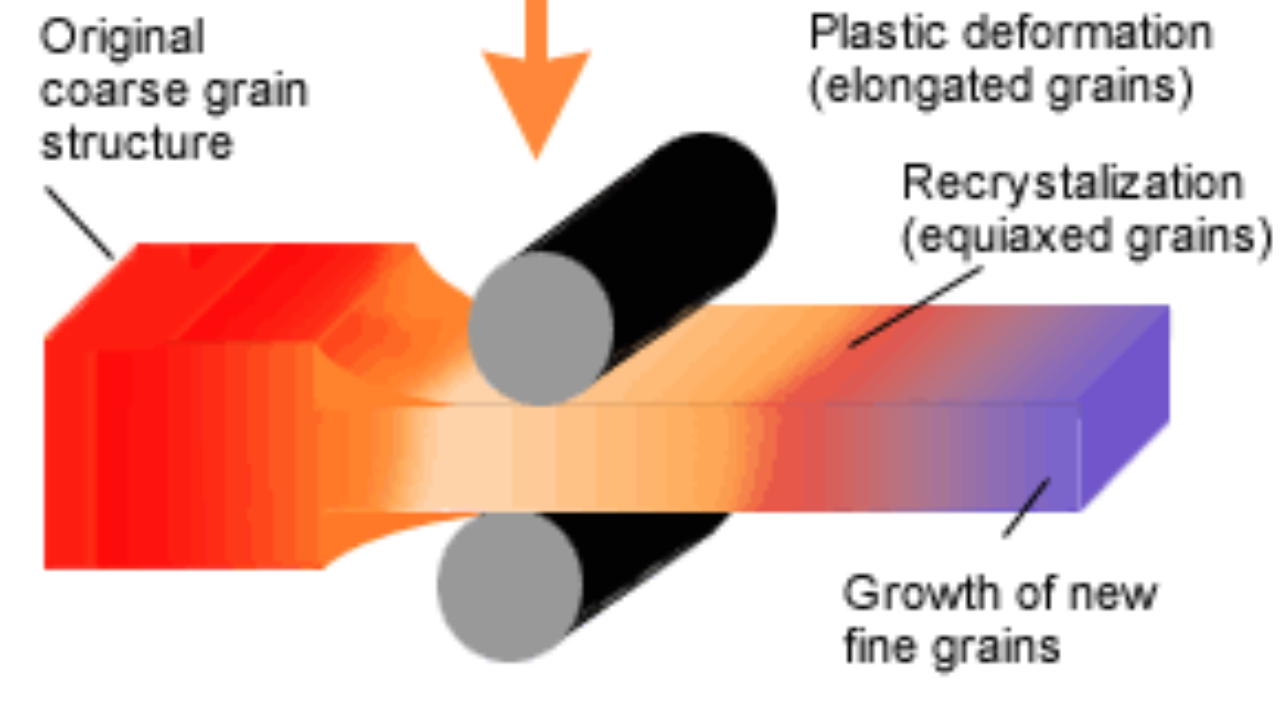


**Rolling process** is a deformation process in which metal(s) in its semi-finished or finished form is passed between the two opposing rollers, which reduces the metal’s thickness through the compression process. The rollers rolls around the metal as it squeezes in between them.



**Hot rolling** is a metalworking process that occurs **above the recrystallization temperature** of the material. After the grains deform during processing, they recrystallize, which maintains an equiaxed microstructure and prevents the metal from work hardening.

For steels, hot rolling is generally conducted at 850–1200 °C, in austenite phase.



As compared to this, **cold rolling** is a metalforming process where rolling is done below the recrystallization temperature of the metal. The metal is compressed and squeezed, increasing the yield strength and hardness of the metal.

**Recrystallisation** : Under the action of heat and force, when the atoms reach a certain higher energy level, the new crystals start forming, which is termed as recrystallisation. Recrystallization destroys the old grain structure deformed by the mechanical working, and entirely new grain structures, which are strain free, are formed.

Now, the minimum temperature at which complete recrystallisation of a cold worked metal occurs within a specified time is known as **recrystallisation temperature**.

The recrystallisation temperature of pure iron is 450°C, and that of steel is between 400°C to 700°C.

**Advantages of hot rolling -**

* Requires less force for deformation as the bonds between the molecules are already weakened because of the applied temperature
* Can be done with brittle materials, as the ductility increases when increasing temperature
* Final grains formed would be strain free, and with better mechanical properties

**Defects** -

Defects in the appearance, form, dimension, macrostructure/microstructure and/or chemical properties of steel products are classified as deviations compared to the requirements stated in the technical standards or any other relevant regulatory documents. Defects are found either by means of visual inspection or with the assistance of tools and devices.

The steel defect data collected had many kinds of defects like crazzing, pitted surface, scratches, folding, inclusion, rolled-in-scale etc. These all can arise due to hot and cold rolling of structural steels. Such defects can easily be seen on the surfaces of steels hence it is possible to make use of artificial intelligence for the identification of these defects. The dataset can be broadly divided into 4 categories of defects.

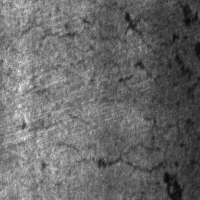
1. Patches
2. Crazing and pitted surface
3. Scratches
4. Inclusion
5. Rolled in scale

* **Patches** - These are material discontinuities produced while hot rolling. These patches and spots can appear on the surface due to any kind of impurities.



**Patches on steel surface**

* **Crazing and Pitting -** In metallic surface regions that undergo high stress, crazing is propagated, which is most likely during rolling and contributes to the formation of microvoids and tiny cracks. If there is ample applied tensile stress, then these cracks elongate and break, causing the microvoids to develop and expand. Pitting is a corrosion defect and is a localized type of corrosion created in the material by cavities or "holes".

****

**Crazing and pitted surface**

* **Scratches -** Scratches are visually observed and triggered during rolling by accidental contact with the build-up of mechanical parts and mill components. One of the steel surface defects is the scar-like foil performance of the metal surface. Rolling resulting in scratch is potentially a defect.

****

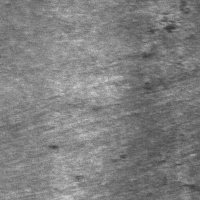
**Scratches**

* **Inclusions** - Inclusions are non-metallic compounds and precipitates which form in steel during its production and processing.



**Inclusions on steel surface**

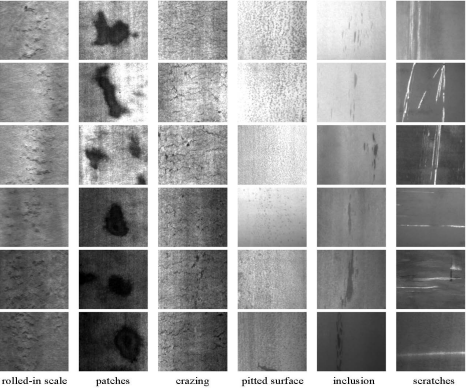
* **Rolled-in scale -** It ranges from fractured streaks to irregular blemishes. Oftentimes scale will remain in the marking. May be deep or very shallow, depending on when (and how much of) the scale was rolled in during processing.



**Rolled-in scale defect**

**2.2 Methodology**

* **Dataset:** The data we are using is unstructured data in the form of images and there is one XML file corresponding to each image which contains the location of the defect. It is taken from Northeastern University (NEU) surface defect database. It includes six kinds of typical surface defects of the hot-rolled steel strip collected, i.e. 1. crazing (Cr) 2. patches (Pa) 3. pitted surface (PS) 4. rolled-in scale (RS) 5. inclusion (In) 6. scratches (Sc).



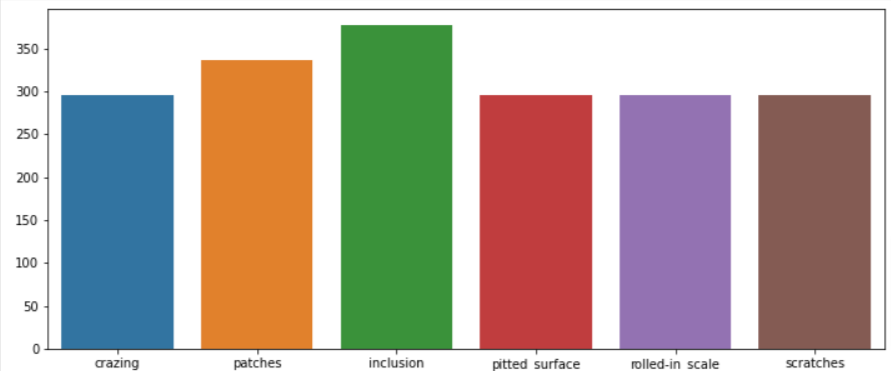
The database includes 1800 grayscale images: 300 samples each of six kinds of typical surface defects. Fig. shows the sample images of six kinds of typical surface defects, the original resolution of each image is 200×200 pixels

* **Data Analysis:**

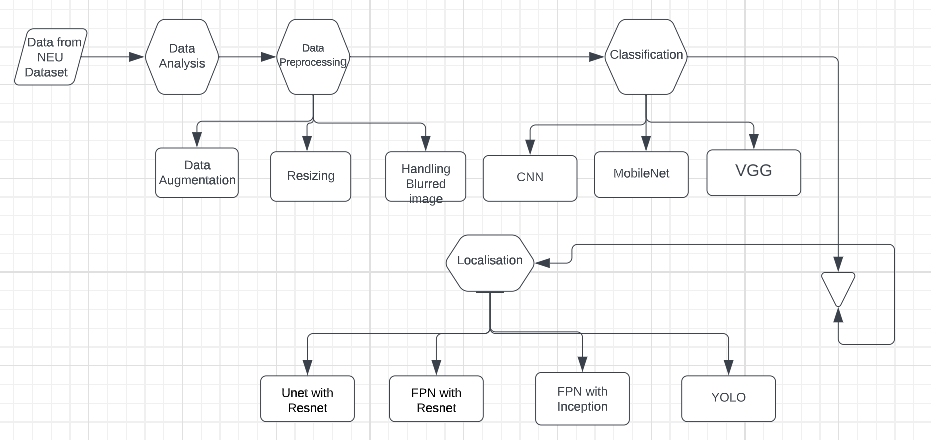
On plotting the number of defects on each image in the pie-chart



From the above results it seems the majority of the images have a single type of defect but some have 2 types of defects and even some have 3 types of defects. But, samples with 3 types of defects are very less i.e only 0.3%. Next, let's draw a count plot of each defect.



So, from the above result, it seems that there is an equal number of crazing, pitted surfaces, rolled-in scale, and scratches on the samples. However, inclusion and patches are present more in the samples.



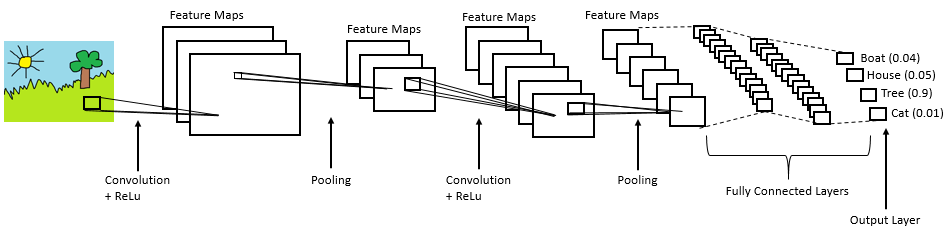
**2.3 Theoretical / Experimental work**

**Data preprocessing**: After the data analysis, the main step in every machine learning problem is data preprocessing. Data preprocessing is the process of transforming raw data into something that can be used by a machine learning model. It is the initial and most important stage in developing a machine-learning model. The format of the real-world data we have is useless and cannot be utilized directly in machine learning models. Data cleaning and preparation are necessary in order to make the data appropriate for a machine learning model, which also improves the model's efficacy and accuracy.

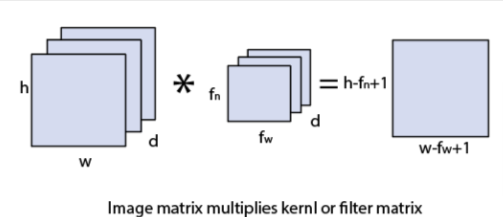
If we preprocess the data efficiently we can get better results from simple algorithms also. As the dataset contains some noisy images or in the ideal case, there may be a chance of getting blurred or noisy images handled that there must be some noisy images in the original dataset.

1. **Data Augmentation:** We use 5 data augmentation techniques on training images so that our model becomes more generalized. These techniques include inverting, rotating, Flipping, adding contrast, and changing the saturation of the image dataset.
2. **Dealing with Blurred images:** We have used image sharpening and a gaussian blur filter. Also, we have found that if we add blurred image to the dataset the model efficiency for handling blurred images increases and hence it improves the accuracy of the model.

**Convolution Neural Network(CNN)**: Convolutional Neural Network is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, object detections, face recognition, etc., are some of the areas where convolutional neural networks are widely used. Each input image in CNN will be processed by a series of convolutional layers, pooling, fully connected layers, and filters (also known as kernels). The Soft-max function will then be used to categorize an item using probabilistic values between 0 and 1.

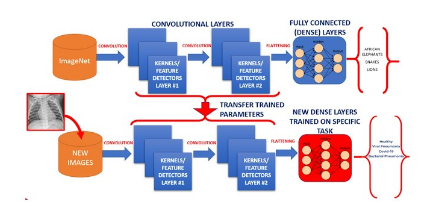


1. **Architecture:** The first layer to extract features from an input picture is the convolution layer. The convolutional layer maintains the link between pixels by learning visual properties using a tiny square of input data. An image matrix and a kernel or filter are two inputs that are used in this mathematical technique.

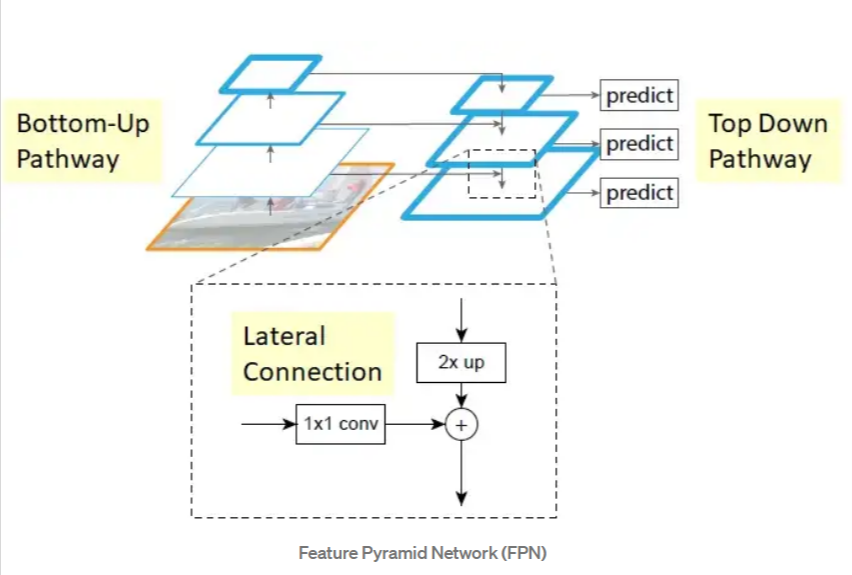


1. **Pooling Layer:** The pooling layer plays an important role in pre-processing of an image. The pooling layer reduces the number of parameters when the images are too large.
2. **Fully connected layer:** The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

**Transfer Learning:** A model that has been trained for one job is repurposed for a different, related task using the machine learning approach known as transfer learning. In the initial layers, forms are often sought after by neural networks, followed by certain task-specific properties in the intermediate layers. The early and intermediate layers are utilized in transfer learning, whereas the latter layers are just retrained. It aids in utilizing the labeled data from the first job it was trained on. Although transfer learning has other advantages as well, its key advantages include **reducing training time**, improving neural network performance (in most circumstances).



**Feature Pyramid Network (FPN):** FPN is not an object detector by itself. It is a **feature extractor** that works with object detectors. Through lateral connections and a top-down approach, it blends low-resolution, semantically robust characteristics with high-resolution, semantically weak ones. This feature pyramid is generated fast from a single input image scale and has extensive semantics at all levels, doing so without losing representational strength, speed, or memory.

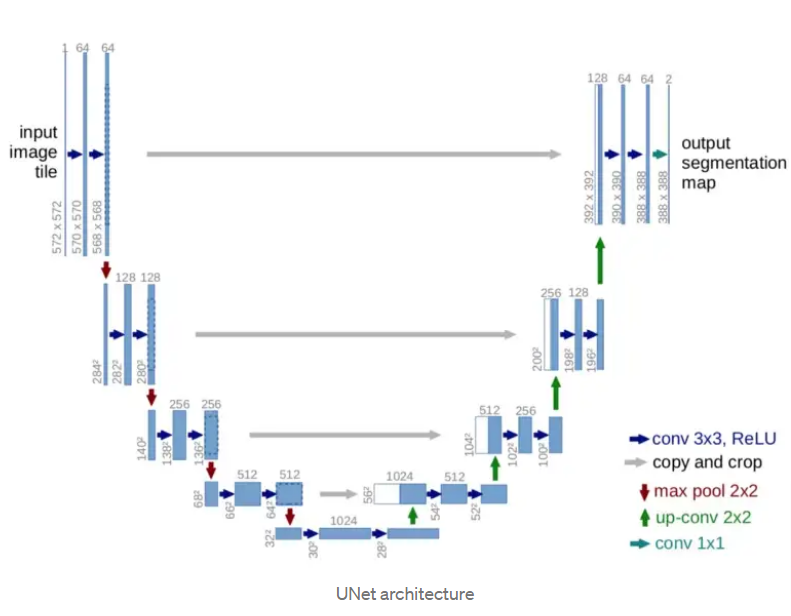


1. **Bottom-up pathway**: Feedforward computation of the ConvNet's backbone is the bottom-up approach. Each step is said to have its own pyramid level. In order to enhance the top-down route through lateral connection, the output of each state's final layer will serve as the reference set of feature maps.
2. **Top-down pathway**:

* The feature maps from higher pyramid levels are upsampled to provide higher resolution features that are geographically coarser but semantically stronger. For ease of use, the spatial resolution is particularly upsampled by a factor of 2 using the nearest neighbor
* Each lateral link combines feature maps from the top-down and bottom-up pathways that are the same spatial size. To specifically decrease the channel dimensions, 11 convolutions are applied to the feature maps from the bottom-up pathway
* And through element-wise addition, the feature maps from the top-down and bottom-up pathways are combined

FPN extracts feature maps and later feeds into a detector, say ResNet, for object detection.

**UNet:** UNet, which developed from the conventional convolutional neural network, was created and used for the first time in 2015 to process pictures used in biomedicine. A standard convolutional neural network focuses on classifying images, with an input of an image and an output of a single label. However, in biomedical applications, it is necessary to identify both the presence of a disease and the location of the abnormality. UNet is devoted to finding a solution to this issue. It helps to localize and identify boundaries since every pixel is classified, ensuring that the input and output are of the same size.



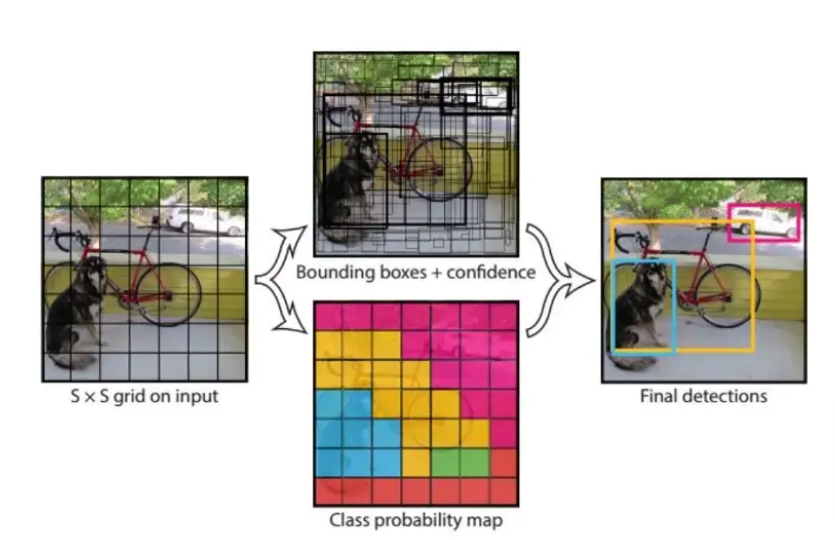
It appears to be in the shape of a "U." The design is symmetrical and is divided into two main sections: the left section is known as the **contracting route** and is made up of the basic convolutional process; the right section is known as the **expanding path** and is made up of transposed 2D convolutional layers.

The creator of UNet asserts in his research that the network is robust enough to make accurate predictions based on even small data sets by employing excessive data augmentation techniques. UNet can do image localization by predicting the picture pixel by pixel.

**YOLO:** Yolo architecture is more like a fully convolutional neural network (FCNN), which processes an image (nxn) just once before producing a prediction (mxm), as opposed to other region proposal classification networks (fast RCNN), which perform detection on a variety of region proposals and ultimately end up performing prediction for various regions in an image multiple times. This architecture **divides the input picture into mxm grids**, with two bounding boxes and class probabilities for each grid generation. Keep in mind that the bounding box is more likely to be larger than the grid.

YOLO directly enhances detection performance while training on complete photos. Comparing this unified model to conventional object identification techniques provides a number of advantages. First off, YOLO moves really quick. We don't require a sophisticated process as we define detection as a regression problem. To forecast detections, we only execute our neural network on a fresh picture at test time.

Second, while generating predictions, YOLO thinks broadly about the image. Because YOLO views the full image during training and testing, unlike sliding window and area proposal-based approaches, it implicitly stores contextual information about classes in addition to their appearance. Fast R-CNN, a popular approach for object recognition, misinterprets background patches in an image as objects because it lacks context awareness. Compared to Fast R-CNN, YOLO creates fewer than half as many background mistakes.



Additionally, because YOLO is so generalizable, it is less likely to fail when used with unfamiliar contexts or inputs. For each bounding box prediction, our network makes use of characteristics from the full image. Additionally, it forecasts all bounding boxes for a picture across all classes. The YOLO architecture maintains excellent average precision while enabling end-to-end training at realtime speeds.

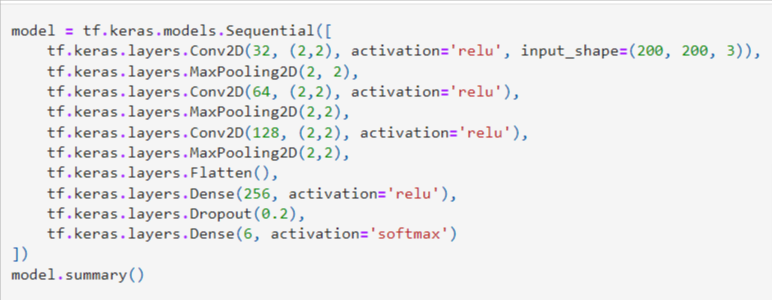
The input image is divided into a SxS grid by our method. A grid cell is in charge of detecting an object if its center falls within that grid cell. The confidence ratings should be 0 if there is no item present in that cell. In the absence of it, we want the confidence score to be equal to the intersection over union (IOU) between the projected box and the actual data.

**2.4 Implementation**

**Classification Models**

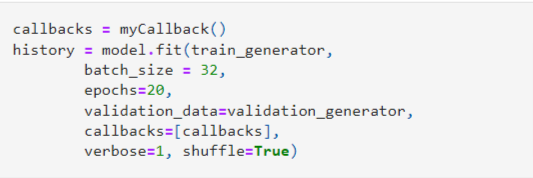
**CNN from scratch:**

We have started our classification task by creating convolution neural network (CNN) from scratch. For this network we use loss function as categorical cross entropy, and optimiser as root mean square propagation. We use 3 convolution layer with activation function as Relu and one dense layer with activation function Relu and output layer with softmax as activation function.



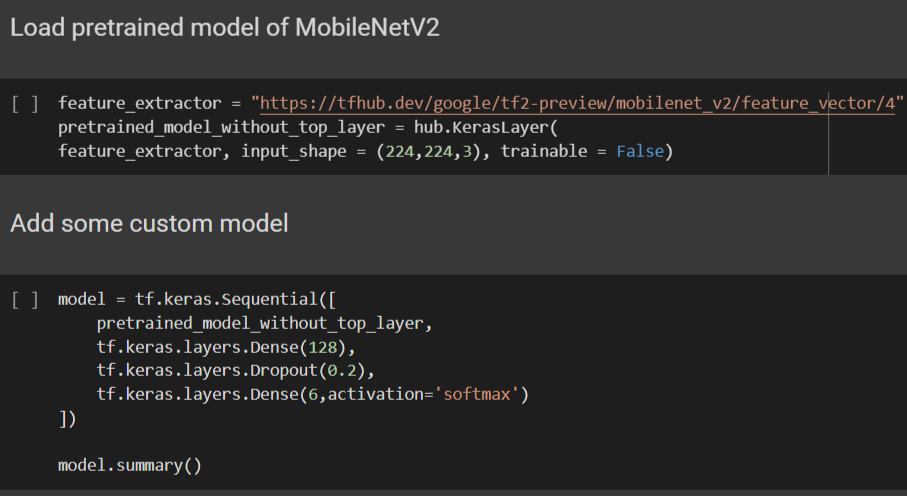


This model is trained for 20 epochs.

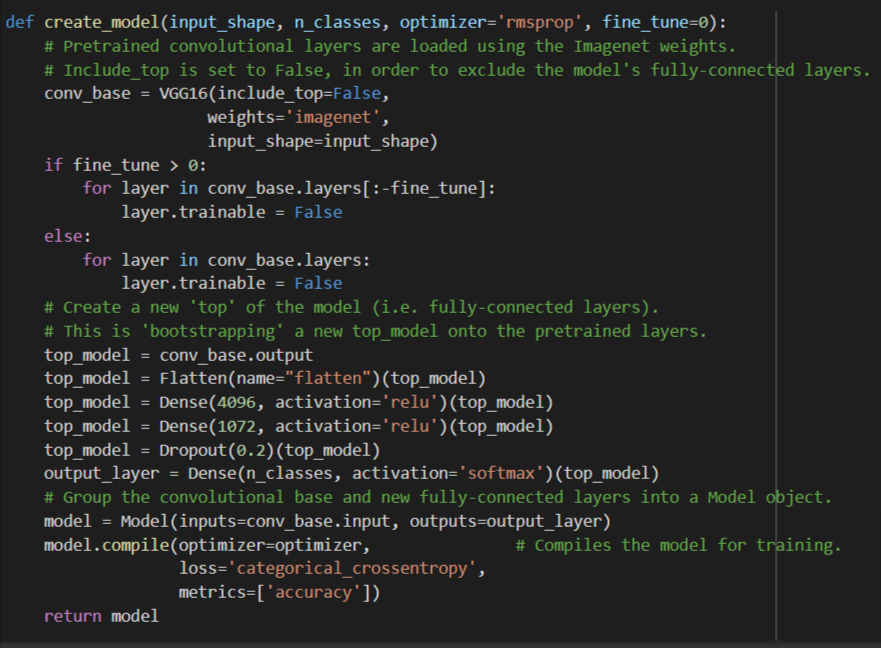


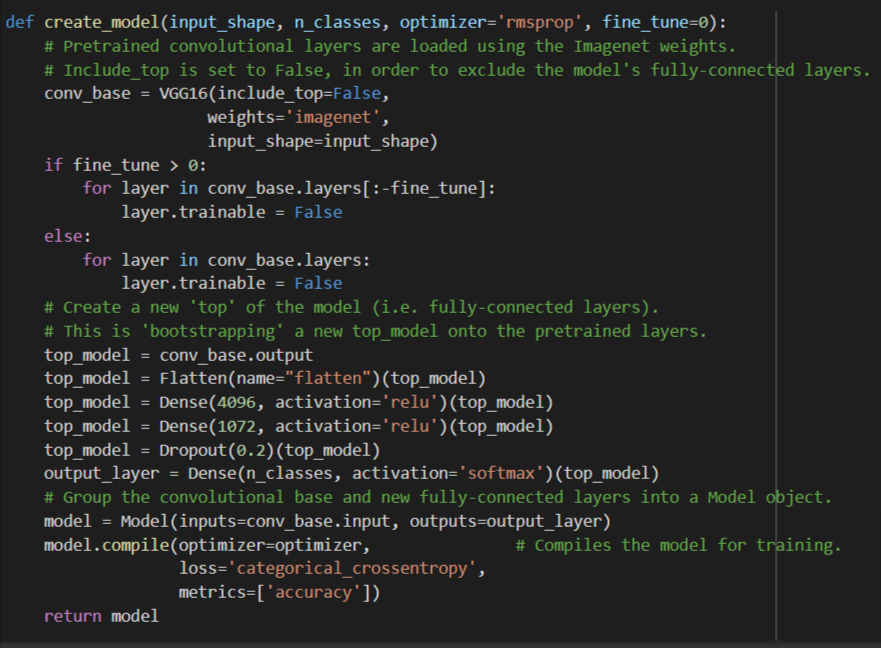
**CNN With Transfer Learning:** We have tried multiple models using transfer learning techniques for finding the best accuracy. These models are **VGG16** and **MobileNet**.The best accuracy we got among them is with MobileNet, around **99.72%**. The complete results of all the models will be discussed in results section.

**MobileNet:**

****

**VGG:**





**Localisation Task**

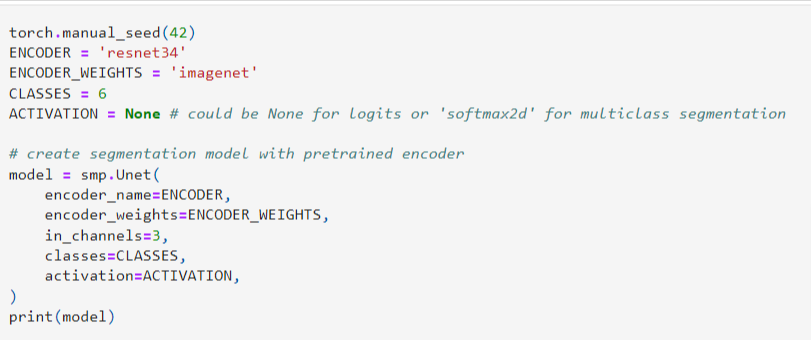
Before starting the localisation task we have to convert the xml file which contain the coordinates of the defect to the csv file so that we can feed that to our network.There is one xml file corresponding to the each image.

**Converting xml files to csv files**: For this task we have used imguag library

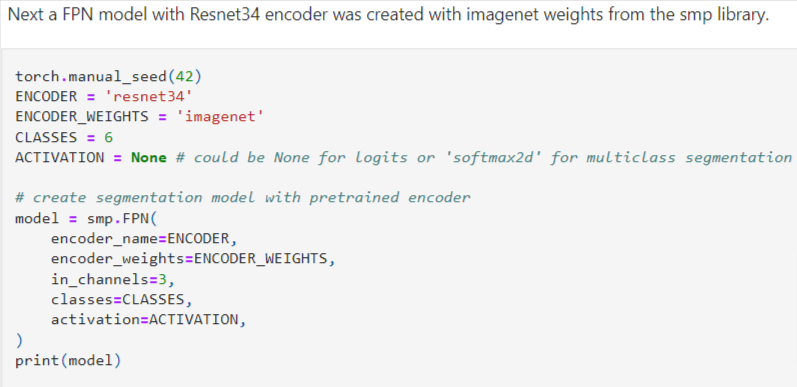


After converting the xml file to csv file we used various approach for defect localisation in the images. The best results have been found using YOLO model. All the models are described below:

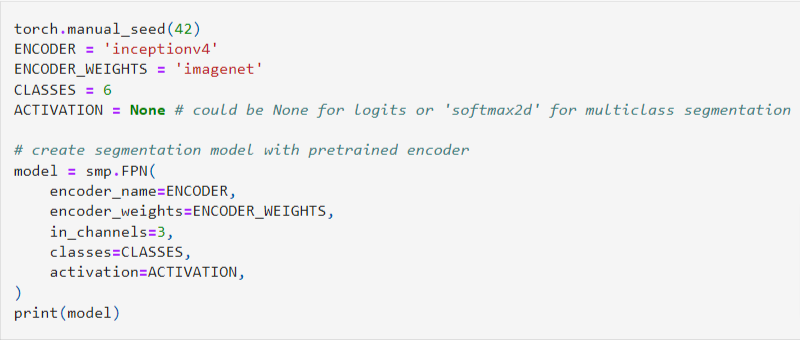
**Unet with Resnet34 encoder**:



**FPN with Resnet34:**

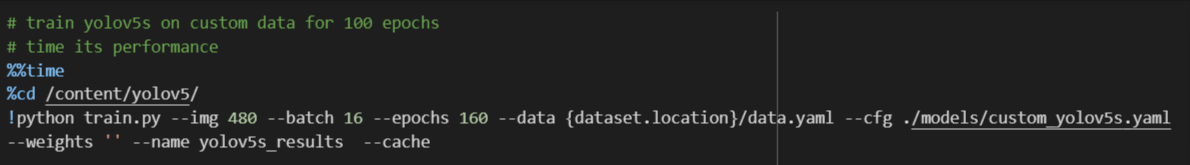
****

**FPN with InceptionV4:**

****

**YOLO:**

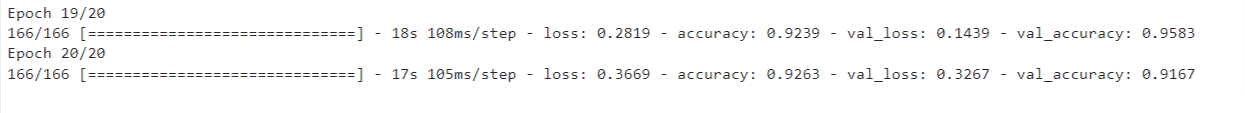


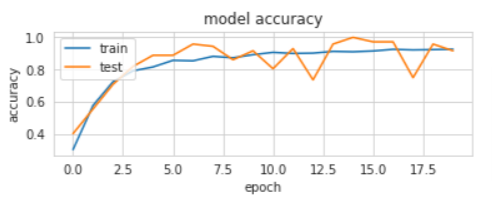


3. **Results**

**Classification Task**

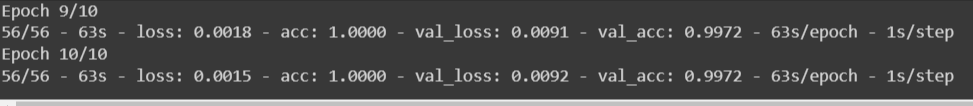
**CNN Scratch:**

****

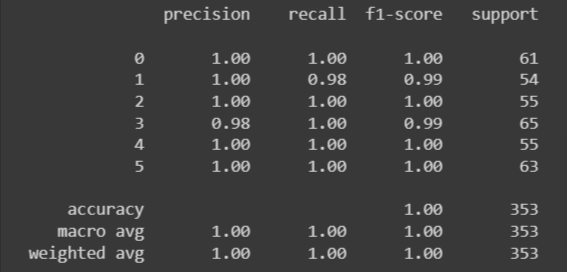
****

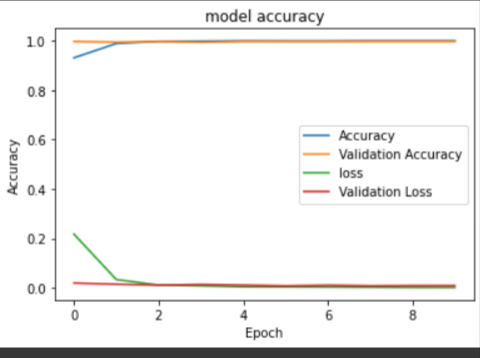
On training CNN from scratch we got training accuracy around 92% and validation accuracy around 91 % and on increasing the epoch accuracy does not increase significantly which can be seen from the graph above as the training line flatten.

**MobileNet:**

****

Classification report:

****

****

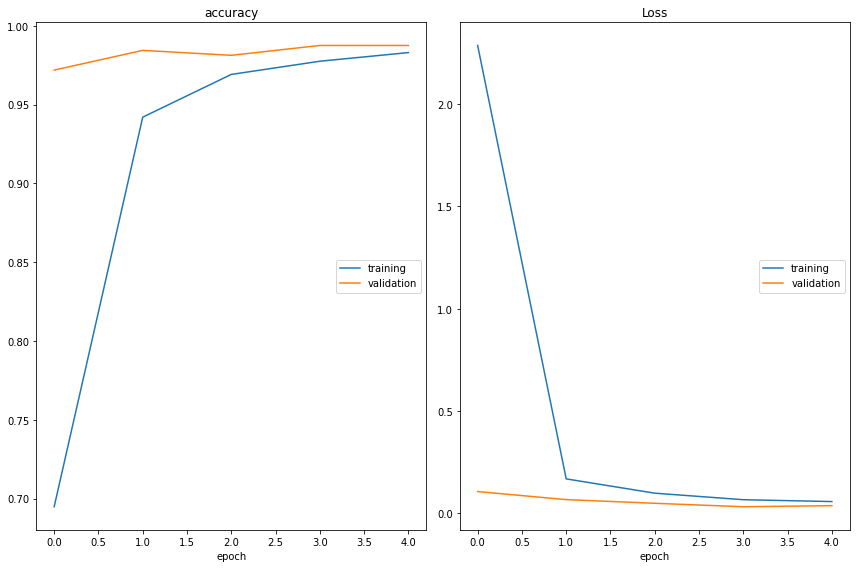
On training the mobilenet model we observe that we get accuracy around 99% within 10 epochs only and training and testing accuracy line becomes constant so in less epochs only it gives better result.

**VGG:**

****

Classification report:

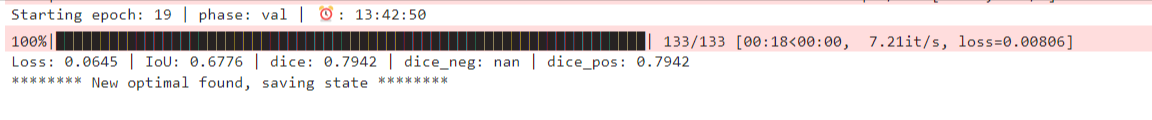




On training the VGG model we observe that we get accuracy around 98% within 10 epochs only and training and testing accuracy line becomes constant so in less epochs only it gives better results than scratches CNN but it takes more training time than MobileNet model.

| **Model** | **Accuracy** | **Training Time** |
| --- | --- | --- |
| **Simple CNN** | **91.67%** | **9 min 12 sec** |
| **MobileNetV2** | **99.72%** | **10 min 50 sec** |
| **VGG16** | **98.8%** | **12min 57sec** |

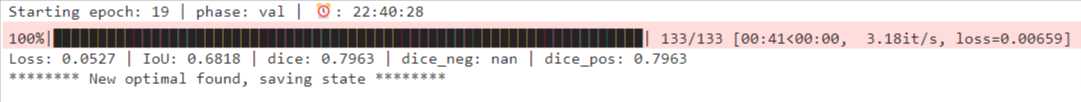
**Localisation Task**

**Unet with Resnet34 encoder**:



From the above results it can be concluded that the loss IOU get's reaches a plateau after epoch 15. So, it can be concluded that it's not worth training the model after 20 epochs.Also we got IOU score around 67.76% and dice score as 79.42%.Its training time is around 2hr46min.

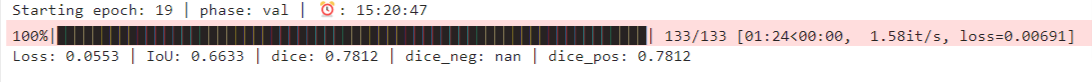
**FPN with Resnet34:**

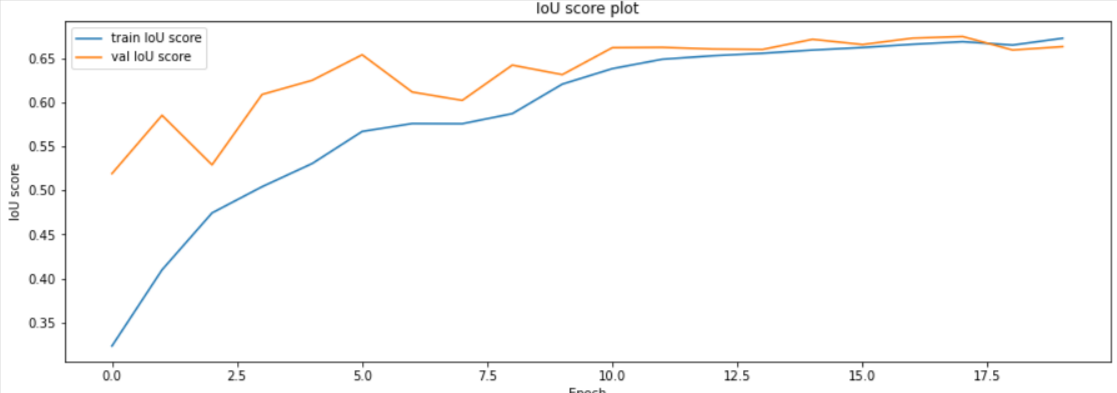
****

****

The above results show that overall training and validation IoU goes up with epochs and training and validation loss goes down with epochs.Also we got IOU score around 68.18% and dice score as 79.63%.But it takes significantly less time as compared to previous model.Its training time is around 1hr27min.So we can say that using FPN reduce the training time.

**FPN with InceptionV4:**

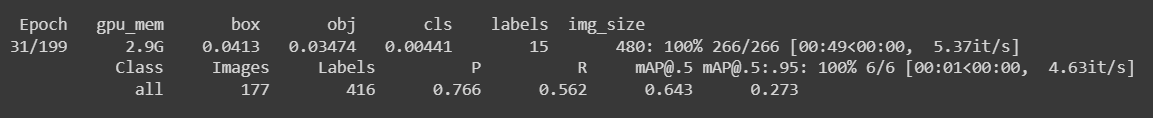


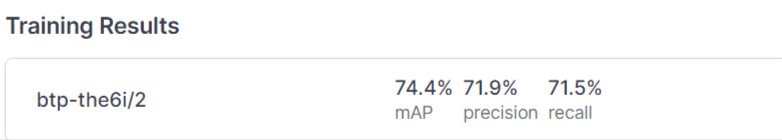


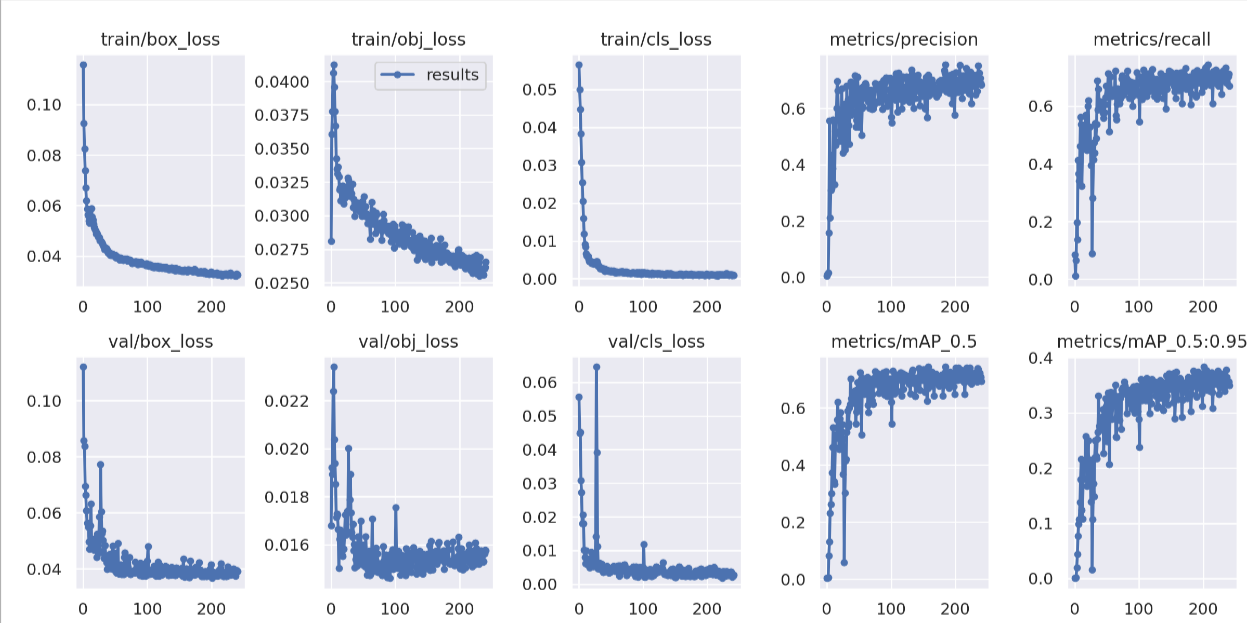
The above results the overall training and validation IoU goes up with epochs and it becomes constant around 15 epochs.Also we got IOU score around 66.33% and dice score as 78.12%.But it takes much more time as compared to previous model.Its training time is around 4hr16min.So we can say that Inception model takes much more time for training the model.

In the above 3 models we got around the same IOU and Dice score but we know that YOLO can perform better from all of this as it is researched that YOLO takes less training time and gives better results than sliding window techniques.

**YOLO:**

****

****

****

On training the YOLO model for 20 epochs we got IOU around 68.4% but it takes much less time as compared to sliding window technique and accuracy is also more.Also on training it for 200 epochs we got significant increase in accuracy our accuracy increases to 74.4% and we found out from the plot that the accuracy becomes constant after that so there is no need of further training of the model.

****

The above image is the result of the test image that is predicted by our YOLO model.Each bounding box tells that there is a defect and and type of defect is in the index form whose values are as follows.

Here indexes are as for the defects :0. crazing (Cr) 1. patches (Pa) 2. pitted surface (PS) 3. rolled-in scale (RS) 4. inclusion (In) 5. scratches (Sc).

| **Model** | **IOU** | **Epochs** | **Training Time** |
| --- | --- | --- | --- |
| **Unet with Resnet34** | 67.78 | 20 | 2hr 46 min |
| **FPN with Resnet34** | 68.18 | 20 | 1hr 27 min |
| **FPN with InceptionV4** | 66.33 | 20 | 4hr 16 min |
| **YOLO** | 68.4 | 20 | 19 min 40 sec |
| **YOLO** | 74.4 | 200 | 3hr 5 min |

**4. Conclusion**

1. We have observed that increasing the size of the datasets by generating our own datasets using data augmentation and adding blur images increases the accuracy of the model.
2. We got maximum accuracy for classification by using transfer learning technique and also it takes less time for training as it is expected and between different transfer learning we got maximum accuracy in **MobileNet** model.
3. For Localisation tasks we got maximum accuracy by using **YOLO** algorithm and also it takes less training time for training the model.

**References:**

1. NEU surface defect database by kitchen Song and Yunhui Yan

2.<https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c>

3.<https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006>

4. Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville(2017, MIT)

5. An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features Yu He, Kechen Song , Qinggang Meng , and Yunhui Yan

6. M. Chu, J. Zhao, X. Liu, and R. Gong, “Multi-class classification for steel surface defects based on machine learning with quantile hyperspheres,” Chemom. Intell. Lab. Syst., vol. 168, pp. 15–27, Sep. 2017

7. <https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5>

8. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Neural Inf. Process. Syst. (NIPS), Las Vegas, NV, USA, Dec. 2012, vol. 60, no. 2, pp. 1097–1105

9. <https://github.com/rozumden/ShapeFromBlur>

10. <https://www.youtube.com/watch?v=O5jECHOjK3s>

11. <https://www.mechanicalfunda.com/2017/01/advantage-disadvantage-hot-rolling.html>

12. [https://themechanicalengineering.com/forming](https://themechanicalengineering.com/forming/)

13. <https://www.youtube.com/watch?v=1Vi3gv09wbY>

14. <https://www.steelwarehouse.com/hot-rolling-defects/>

15.<https://github.com/asetkn/Tutorial-Image-and-Multiple-Bounding-Boxes-Augmentation-for-Deep-Learning-in-4-Steps>