

**B. TECH. PROJECT
REPORT
On**

**Fractographic Classification
of Boron modified Ti64 alloy**

By
Yash Kothekar



**DEPARTMENT OF METALLURGY
ENGINEERING AND MATERIALS SCIENCE
INDIAN INSTITUTE OF TECHNOLOGY INDORE**

May 2022

Fractographic Classification of Boron modified Ti64 alloy

PROJECT REPORT

*Submitted in partial fulfillment of the
requirements for the award of the degrees*

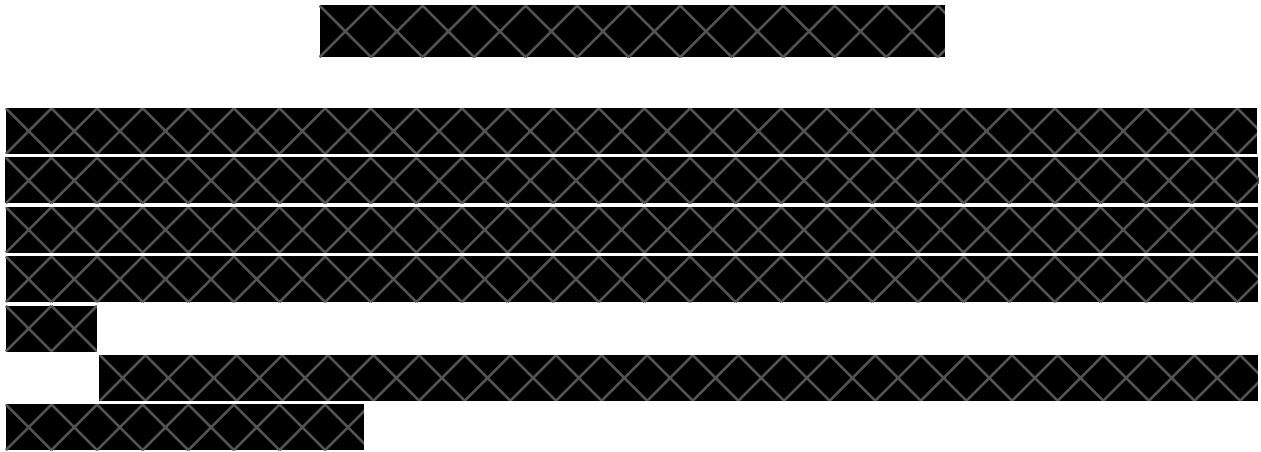
of
BACHELOR OF TECHNOLOGY
in
METALLURGICAL ENGINEERING AND MATERIALS SCIENCE

Submitted by:
Yash Kothekar (180005036)

Guided by:
Dr. Eswara Prasad Korimilli (IIT Indore)



INDIAN INSTITUTE OF TECHNOLOGY INDORE
May 2022



A photograph of a handwritten signature in cursive script, appearing to read "Mark". The signature is written in black ink on a light-colored background. It is enclosed within a rectangular frame. Three black rectangular redaction boxes are placed below the frame: one centered above the signature and two stacked to the right of the bottom edge.



A photograph of a handwritten signature in cursive script, appearing to read "K. Emmer manad". The signature is written in black ink on a light-colored background. It is enclosed within a rectangular frame. A single black rectangular redaction box is located at the top center of the frame.



PREFACE

This B.Tech. project report on “**Fractographic Classification of Boron modified Ti64 alloy**” is prepared under the guidance of Dr. Eswara Prasad Korimilli.

Through this thesis, I have tried to create an approach that will assist in the problem of fractographic classification. Deep learning models have been used to increase the accuracy and efficiency of classification along with providing a way to analyze the results. Through this thesis, efforts have been made to present the methodology, results, and conclusions of the study in a lucid and comprehensible manner. Figures, graphs, and tables have been included to make the content more illustrative.

Yash Kothekar

B.Tech. IV Year

Discipline of Metallurgy Engineering and Materials Science

IIT Indore

Acknowledgements

I extend my heartfelt gratitude to Dr. Eswara Prasad Korimilli for his valuable guidance in structuring the project and providing valuable feedback throughout the course of this project. His confidence and optimism in the project were the driving force for my research interest.

I am thankful to Ms. Tulika Dixit for helping me understand the material and its fracture features better. She contributed immensely to the annotation process and also provided a path for analysis.

The SEM analysis used to procure the images in the dataset was performed by Ms. Tulika Dixit and Mr. Venkata Sandeep at the IIT Indore facility. I am grateful to the Institute for providing this opportunity to be exposed to research in the inter-disciplinary domain of metallurgy and machine learning and for providing the necessary utilities to complete this project. Lastly, I offer my sincere thanks to everyone who helped me complete this project, whose name I might have forgotten to mention.

Yash Kothekar
B.Tech. IV Year
Discipline of Metallurgy Engineering and Materials Science
IIT Indore

Abstract

Fractographic classification is generally performed manually by analyzing the SEM images of fracture surfaces. This work aims to assist humans in this area, making the process less time-consuming and maintaining result consistency by using the capability of computer vision.

Various deep learning-based models, specifically CNNs, were used to classify the SEM images of fracture surfaces. The low amount and type of data provided challenges to the classification task. Results of several of these models were analyzed and compared, with each result giving us a probability of a particular class of fracture based on the image fed to the model. Heat Maps proved to be very useful in analysis, as they highlighted areas of the image the model had considered to give a specific outcome.

Modified VGG16 and ResNet50 proved to be best suited for the dataset provided.

Table of contents

CANDIDATE'S DECLARATION	5
CERTIFICATE by BTP Guide(s)	5
PREFACE	7
Acknowledgements	9
Abstract	11
Table of contents	13
List of Figures	16
List of Tables	18
Chapter 1	21
Introduction	21
1.1 Motivation	21
1.2 Outline of the Report	21
Chapter 2	22
Literature review	22
2.1 Fractography	22
2.2 Machine learning	22
2.2.1 Computer Vision	22
2.2.2 CNN	23
2.2.2.1 Classification Models	23
2.2.2.2 Metrics	24
2.2.2.3 Hyperparameters	25
2.3 Computer Vision and Fractography	26
2.3.1 Fractographic Classification	26
Chapter 3	28
Problem Formulation and Objectives	28
3.1 Material	28
3.1.1 Ti64	28
3.1.2 B Modified Ti64 alloy	28
3.1.3 Types of Fractures observed in B Modified Ti64	29
3.1.4 Fracture surface analysis	29

3.2 Experimentation	30
3.3 Computer Vision Intervention	30
3.4 Objectives	31
 Chapter 4	
Dataset	32
4.1 Image Cleaning	32
4.2 Dataset Preparation	32
4.3 First Dataset	33
4.4 Second Dataset	33
 Chapter 5	
Method and Results	35
5.1 Image Pre-Processing	35
5.2 Loss function and Hyperparameter Tuning	36
5.3 Network Architectures	37
5.3.1 Introduction	37
5.3.2 Transfer Learning	38
5.3.3 AlexNet	38
5.3.3.1 Overview	38
5.3.3.2 Analysis	40
5.3.4 VGG16	41
5.3.4.1 Overview	41
5.3.4.2 First Dataset	42
5.3.4.2.1 Analysis	43
5.3.4.2 Second Dataset	44
5.3.4.2.1 Analysis	44
5.3.5 VGG19	46
5.3.5.1 Overview	46
5.3.5.2 Analysis	47
5.3.6 ResNet50	49
5.3.6.1 Overview	49
5.3.6.2 Analysis	50
5.3.7 InceptionV3	51
5.3.7.1 Overview	51
5.3.7.2 Analysis	52
5.4 Final Results and Analysis	54
5.4.1 Final Result	54

5.4.2 Analysis	55
5.4.2.1 VGG16	55
5.4.2.2 ResNet50	57
Chapter 6	60
Discussions and Future scope	60
6.1 Key Findings	60
6.1.1 Key Features	60
6.1.2 Model Drawbacks	60
6.1.3 Precautions	61
6.2 Challenges Involved	61
6.2.1 Dataset Size	61
6.2.2 Class Identification	61
6.2.2.1 Interclass and Intraclass variation	61
6.2.2.2 Skewed Dataset	62
6.3 Future Scope	62
References	63

List of Figures

- Figure 1 Classification Dataset Visualisation
Figure 2.1, 2.2, 2.3 Original and transformed images after using Image Data Generator Class
Figure 3 AlexNet architecture
Figure 4.1, 4.2 Images for AlexNet Analysis
Figure 5.1, 5.2 Images for AlexNet Analysis
Figure 6.1, 6.2 Images for AlexNet Analysis
Figure 7 SEM image
Figure 8 VGG16 Architecture
Figure 9.1, 9.2
Figure 10.1, 10.2 Images for VGG16 Analysis
Figure 11.1, 11.2 Images for VGG16 Analysis
Figure 12.1, 12.2 Images for VGG16 Analysis
Figure 13.1, 13.2 Images for VGG16 Analysis
Figure 14.1, 14.2 Images for VGG16 Analysis
Figure 15.1, 15.2 Images for VGG16 Analysis
Figure 16 VGG19 Architecture
Figure 17.1, 17.2 Images for VGG19 Analysis
Figure 18.1, 18.2 Images for VGG19 Analysis
Figure 19.1, 19.2 Images for VGG19 Analysis
Figure 20.1, 20.2 Images for VGG19 Analysis
Figure 21 ResNet50 Architecture
Figure 22.1, 22.2 Images for ResNet50 Analysis
Figure 23.1, 23.2 Images for ResNet50 Analysis
Figure 24.1, 24.2 Images for ResNet50 Analysis
Figure 25.1, 25.2 Images for ResNet50 Analysis
Figure 26 InceptionV3 Architecture
Figure 27.1, 27.2 Images for InceptionV3Analysis
Figure 28.1, 28.2 Images for InceptionV3Analysis
Figure 29.1, 29.2 Images for InceptionV3Analysis
Figure 30.1, 30.2 Images for InceptionV3Analysis
Figure 31.1, 31.2 Images for VGG16 Final Analysis
Figure 32.1, 32.2 Images for VGG16 Final Analysis
Figure 33.1, 33.2 Images for VGG16 Final Analysis
Figure 34.1, 34.2 Images for VGG16 Final Analysis
Figure 35.1, 35.2 Images for VGG16 Final Analysis
Figure 36.1, 36.2 Images for ResNet50 Final Analysis
Figure 37.1, 37.2 Images for ResNet50 Final Analysis
Figure 38.1, 38.2 Images for ResNet50 Final Analysis

Figure 39.1, 39.2 Images for ResNet50 Final Analysis

Figure 40.1, 40.2 Images for ResNet50 Final Analysis

List of Tables

Table 1 Confusion Matrix

Table 2.1,2.2 Information for first dataset, Information for second Dataset

Table 3 AlexNet Confusion Matrix

Table 4 VGG16 Confusion Matrix on first dataset

Table 5 VGG16 Confusion Matrix on second dataset

Table 6 VGG19 Confusion Matrix

Table 7 Resnet50 Confusion Matrix

Table 8 InceptionV3 Confusion matrix

Table 9 F1 Score list for all models

Table 10 New classification criteria

Chapter 1

Introduction

1.1 Motivation

Fractographic Classification is an essential subset of Fractographic Analysis. It is often required to assess the characteristics of the material and serves as a starting point for enhancing the desired features. Generally, the process is time-consuming and inconsistent, performed by manually analyzing the SEM images of fracture surfaces.

One of the ways to enhance this process is to use the advancements made in Deep Learning, especially in the field of computer vision. In this work, computer vision is not used to substitute but assists humans in the process of fractographic classification.

This work uses the SEM images of fracture surfaces of Boron modified Ti64 as a dataset. Small additions of Boron can enhance the properties of Ti64 alloy and offer affordable processing paths for Ti alloys [6]. Boron modified Ti64 also exhibits complex fracture features, which fall between ductile and brittle classes. This work aims to assist humans by providing the classification result and highlighting the areas considered while making this decision.

1.2 Outline of the Report

This report has been divided into 6 main sections. The First section, gives the basic motivation behind this project and the outline of the report. The second section gives the theory regarding the relevant fields to the project and also discusses previous efforts made in this field. The third section gives information considering the characteristics of the material and the fractures, along with a need for a better solution.

The fourth section talks about the preparation of the datasets along with the modifications made to the data to get better results. The fifth section talks about the different architectures used and is focused on the analysis of the results of each architecture. The last section focuses on drawbacks, challenges involved, and areas of improvement in the current project.

Chapter 2

Literature review

2.1 Fractography

Fractography is the study of fracture surfaces to find the relation between material microstructure and fracture mechanism and, eventually finding out the root cause of the fracture. Such studies help in designing novel microstructures with improved resistance to failure.

Records show that ‘fractography’ can be dated back to the 16th Century [7]. Though the study of fracture surfaces has been an old one, the advancements in microscopy in the 20th Century increased its importance significantly. In 1943, the first attempts to view fracture surfaces at high resolution (1000 X) using Optical Microscopy were made. In 1956, Crussard et al. used high-resolution TEM, and their technique led to much better results. Though, high-resolution TEM revealed fine details, use of SEM led to significant advancements in the study of fracture surfaces.

SEM is widely used today as it provided great resolution, no sample preparation is required, has very large depth of focus and it can accommodate large samples[19]. Understanding provided by has been invaluable in failure analysis and in developing improved materials.

2.2 Machine learning

2.2.1 Computer Vision

Machine Learning aims to create algorithms that can do a task without being specifically taught. Deep Learning is a branch of machine learning based on an Artificial Neural Network. Computer vision comes under a broad Deep learning umbrella, intended to gain insights and *learn* with the help of past data. It specifically deals with how computers can gain high-level understanding from digital images or other visual mediums. From the engineering perspective, it seeks to understand and automate tasks that the human optical system can do.

2.2.2 CNN

Convolutional Neural Networks (CNN) is an artificial neural network used in image recognition and specifically designed to process pixel data. The architecture of a CNN is analogous to the connectivity pattern of Neurons in the Human Brain. A CNN involves a Convolutional layer which, in the case of images, helps in understanding the relation of one pixel to its nearby pixels. This relation to nearby pixels is one of the reasons why a fully connected layer is not as effective as a convolutional layer because it does not consider the spatial connectedness of pixels.

A CNN works in separate but progressing parts. The earlier layers understand more basic features like large image patterns and contrasts. As the layers progress, so do the nuances and intricacies that the layers pick out. In other words, the network learns the high-level features deeper into a CNN.

A crucial part of CNN is the convolutional layer. A convolutional layer consists of filters. A filter is a rectangular grid of numbers that traverses through the image and performs matrix multiplication to get the output of the convolutional layer. Since colored images have three channels, Red, Green, and Blue, these operations take place three times at the start. Below the first layers, these channels become feature maps. These define the depth of the convolution layer. The more things the model wants to learn, the more channels will be included in that layer. Each channel has its filter. One can design these filters manually to extract a specific feature, but in the case of complex images, this task becomes very challenging. Filters are 2D weights, and these weights have a spatial relationship. Just as in a regular ANN, the model learns weights via backpropagation. The same case is valid here too. The model itself learns the filters.

During training, the model learns these features via a process called backpropagation. The model compares the predicted output and the actual output in this process. Using a loss function based on this difference, it adjusts the weights of Kernels in convolutional layers and neurons in fully connected layers to reduce the ‘loss.’

Image Classification is one of the most fundamental challenges in Computer vision. In the Classification problem, we give labeled image data to the model. The role of the model here is to learn what kind of image features correspond to which image labels. After training, the model then ‘predicts’ / classifies an image it has never seen before, just by looking at its features.

2.2.2.1 Classification Models

A few models that have significantly impacted Image classification are

1. AlexNet

AlexNet caused a revolution in the study and implementation of CNNs due to its use of GPU to accelerate deep learning. It also introduced important concepts like Dropout layers, the importance of having deep layers to extract more advanced features, image augmentation, and non-linearity in the network.

2. VGGNet

VGG16 expanded on one of AlexNet's features. It was more 'deep,' as it had many more layers to extract more high-level features. It also used small filter sizes, which increased non-linearity in the system.

3. GoogleLeNet

GoogleLeNet introduced an important concept called inception modules. Here, convolutions with different filter sizes are processed on the same input and then concatenated. Inception modules allow the model to take advantage of multilevel feature extraction at every step. InceptionV3 got its start as a module for GoogleLeNet.

4. ResNet

After a point, stacking more layers does not work due to diminishing gradient. ResNet found a way around it by using residual learning. The model consists of skip connections, which simplify the network and solve the problem of vanishing gradients.

2.2.2.2 Metrics

Metrics help us in analyzing the usefulness of a model. Selecting an appropriate metric for the given task is of immense importance. For the classification task, we consider the following metrics:

1. Accuracy

Accuracy is one of the most common metrics used in Machine learning. It is calculated as the percentage of correct predictions.

2. F1 Score

To analyze the performance of a classification task, we use a confusion matrix. A confusion matrix is a table that gives us concise information regarding the classification task. It gives us the following numbers:

- a. True Positive: A true positive is one which is predicted positive and is actually positive.
- b. True Negative: A true negative is one that is predicted negative and is actually negative.
- c. False Positive: A false positive is one that is actually negative but has been classified as positive.
- d. False Negative: A false negative is one that is actually positive and has been classified as negative.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive* (TP)	False Negative* (FN)
Actual Negative	False Positive* (FP)	True Negative* (TN)

Table 1 Confusion Matrix

- a. Precision:

Precision is the ratio of true positive to predicted positive

$$Precision = TP / TP + FP$$

- b. Recall:

Recall is the ratio of true positive to all actual positive

$$Recall = TP / TP + FN$$

The F1 score is the harmonic mean of Precision and recall. It gives us a good metric that considers both False positives and False negatives [14].

$$F1 = 2 * Precision * Recall / (Precision + Recall)$$

2.2.2.3 Hyperparameters

Hyperparameters are the variable that we provide to the models. These variables govern how the model learns the features and characteristics. Some of these include Batch size, Image size, Loss function, and optimizers.

2.3 Computer Vision and Fractography

2.3.1 Fractographic Classification

There is substantial existing literature on fractographic classification using ANN.

One of the earliest and most prolific works include Bastidas-Rodriguez et.al's work on classifying fracture samples into three categories as sudden ductile, sudden brittle, and fatigue, using ANN and SVMs [1]. One of the key differences between this work and others (including ours) is that it does not use SEM images for testing. From full-scale images' focused areas, average contrast, Average correlation, energy masks, and fractal dimensions are obtained via Gray Level Co-occurrence Matrix, texture energy laws, and fractal analysis. These characteristics were used for training. First, the characteristics were extracted and fed to the model for prediction during testing. Two kinds of classifiers were used - ANN and SVMs, and the following metrics are considered - accuracy, sensitivity, specificity, and precision of the confusion matrix. Though SVM showed better results for a kind of fracture with a specific combination of characteristics, the ANN-based model was more consistent across fractures and combination of characteristics, with better accuracy. The final algorithm showed 77.4% accuracy, which is within the range of experts.

The other work from the same author, "*Deep Learning for fractographic classification in metallic materials*" is also an insightful resource [2] . In this project, they used two types of datasets. One real scale fracture dataset and the second one being SEM images at various magnification. Two kinds of models were employed: traditional CNNs and modified Deep Adaptive Wavelet Network (DAWN). DAWN is adept at highlighting textural features with the help of using adaptive wavelet transforms. Among traditional pre-trained CNNs, the best accuracy and F1 score were obtained via ResNet18 architecture (74.73% accuracy on Real Scale images and 85.48% accuracy on SEM images). On the real scale fracture dataset, DAWN performs on par (75.27% accuracy) with ResNet(trained from scratch) with considerably lower parameters, M-DAWN not being far behind(74.73%). On SEM images, M-DAWN outperforms others with an accuracy of 63.71%.

Wang et al. also worked on a fractographic classification based on the recognition of Fatigue crack initiation sites [4]. In this case, rather than going the way of pre-trained neural networks, a Deeply Supervised Object Detector (DSOD) was used due to its tendency to achieve state-of-the-art results from the limited dataset. Though the study results were not sufficient for practical levels, they set a precedent for the usage of object detectors in the material domain.

Tsopanidis et al. used UNet (one of the state-of-the-art CNNs for Image Segmentation), for the segmentation of Magnesium Aluminate Spinel (MgAl_2O_4), along with Alumina (Al_2O_3) [3]. The dataset consisted of SEM images of fractured MgAl_2O_4 and three classes: Transgranular, Intergranular, and Background. The model achieved a mean IoU value of 91.1% on the SEM test dataset, even on Alumina samples, which it had not encountered before. The model achieved a total mean IoU value of 94%.

Chapter 3

Problem Formulation and Objectives

3.1 Material

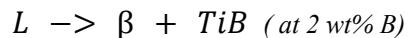
Boron Modified Titanium alloy, consisting of 6 wt% Aluminium and 4 wt% Vanadium, was used at four different boron weight percentages.

3.1.1 Ti64

Titanium typically has an HCP (Hexagonal Close Packing) structure, α phase. It transforms to BCC (Body-Centered Cubic) β phase when heated above 882°C. There is a strong dependence of the transformation temperature on the alloy composition. The relative amounts of alpha and beta phases in any particular alloy have a significant effect on the properties of that material. Titanium alloys are known for their high tensile strength and toughness. They have generally been famous for their lightweight, high strength-to-weight ratio, and corrosion resistance. They are used in biomedical implants and aircraft bodies, among others. Their lack of general use can be attributed to their expensive raw material cost. Some alloying elements stabilize the alpha phase, while others stabilize the beta phase (this is what makes the existence of beta-rich Al alloy possible at room temperature), as they affect the solubility of the alpha or beta phase. These are alpha (e.g., Al, Ga, O, N, C) and beta (e.g., V, Mo, Fe, Cr, Ni) stabilizers. Ti alloys are classified into six types: alpha, Near Alpha, Alpha + Beta, Metastable Beta, Beta, and Titanium Aluminides.

3.1.2 B Modified Ti64 alloy

Boron has a very limited solubility (<0.02%) in the Solid Ti phase. The TiB intermetallic phase forms via the eutectic reaction:



TiB has an orthorhombic crystal structure and is present in phase in the form of short whiskers. It is a stable phase and has good compatibility with the rest of Ti64. Past the eutectic limit, the addition of Boron stops enhancing the material properties, and it, in turn, starts making it more brittle.

4 different amounts of wt% of Boron added to the Ti64 alloy were considered, 0.0, 0.0.9, 0.3, and 0.55. It was found that small Boron additions significantly affect the microstructure and related properties. This can also be confirmed by Yu et al., where the addition of 0.1 wt% Boron was found to have a remarkable effect on the dynamic compression properties of Ti64 [4]. The addition of boron was also found to reduce the grain size significantly.

In Tamirisakandala and Miracle's research on B modified Ti64, they found that the presence of TiB precipitates restricts grain growth at elevated temperatures [6]. The addition of boron was also found to increase the strength and stiffness of the alloy while still preserving good fracture critical properties.

3.1.3 Types of Fractures observed in B Modified Ti64

The three prominent modes of fracture encountered in the three-point bend fracture test of B modified Ti64 were the following

1. Dimpled

Dimple rupture occurs due to a process known as microvoid coalescence. These microvoids first nucleate at grain boundaries, phase interfaces, and inclusions. As the load increases, the microvoids grow, coalesce and eventually fracture. This coalescence results in 'dimple' like shapes on fracture surfaces. Due to the presence of shear force during fracture, the dimples become sheared (greater aspect ratio). The more the material's ductility, the more growth and coalescence will occur, leading to deeper and bigger dimples.

2. Cleavage

This kind of fracture occurs across cleavage planes. The cleavage plane will be the one that will possess the least amount of resistance to the fracture. It is generally observed in BCC and HCP structures. A cleavage surface shows Cleavage steps, river markings, herringbone structure, and feather markings.

3. Quasi-Cleavage/ Quasi-Dimpled

This mechanism involves both microvoid coalescence and cleavage fracture. On the fracture surfaces, characteristics of both types of fractures are not observed exclusively [13]. We generally find dimpled regions merging into cleavage steps or trans-crystalline cracks that form when fracture surfaces pass through lath structures. This project considers Quasi Cleavage and Quasi dimpled as two kinds of fractures, having different amounts of dimpled and cleavage features on their fracture surfaces.

3.1.4 Fracture surface analysis

The fractographic analysis of fracture surfaces using SEM images provides some key insights about the nature of the fracture and its relations with the B content. As the amount of B increased, there is an apparent decrease in grain size and the tendency of intergranular fracture increases.

At 0 and low B%, a significant area of high magnification($> 150X$) images could be classified as ductile. Equiaxed dimples were observed.

As the B% increases, we see more complicated patterns on fracture surfaces. The images could not just be classified as pure Cleavage or Dimpled. Fracture surfaces begin to show Quasi Cleavage and Quasi Dimpled behavior. There was no clear distinction between dimpled area and river markings, which were the characteristics of Cleavage fracture. Also, due to the increase in B%, we see more parallel, straight lines on the fracture surfaces. In high Magnification images, these can be hypothesized as the result of a cleavage plane passing across Lath colonies in Ti64. Cleavage steps are also observed at higher B%.

At 0.55 B%, we begin to observe flat cleavage facets, and the fracture surface behavior shifts from Quasi Cleavage to pure Cleavage. Observing intergranular fracture with flat grain surfaces with some tear edges was not uncommon at this Boron concentration.

3.2 Experimentation

Samples of dimensions 50 mm, 10 mm, and 5 mm with compositions of 0.0 wt%, 0.09 wt%, 0.3 wt%, and 0.55 wt% of Boron were used. The conventional 12 mm SHPB was converted into a Modified SHPB by attaching a jig to hold the sample at the end of the input bar. The instrument consists of An Input bar (material - Al 2062 T6) with a length (Li) of 2.5 m and a diameter (Di) of 12 mm. The mounting of the Strain gauges is done diametrically opposite at the middle of the input bar to measure their respective strain history experienced by the bars during the impact. A striker bar, also of the same material to match the impedance of the input bar, of length (Ls) 650 mm and diameter (Ds) 12 mm. The 650 mm striker bar generates a 250 μ s loading pulse. The striker was released with the help of a gas gun to achieve velocities of around 12 m/s for the impact.

3.3 Computer Vision Intervention

Fractography is used to understand the root cause of fracture via studying the fracture surface. Fractographic classification is a subset of the study of fracture surfaces, which deals with identifying/ assigning the type of fracture surface to meaningful class/ divisions.

Manually Identifying the fracture surfaces requires a lot of time and effort. This process also requires a highly skilled person to identify the characteristics of the fracture. In many cases, the availability of time and skilled personnel is a significant issue, which leads to incorrect classification, resulting in wrong diagnoses and amendments that do not solve the existing problem.

Thus, there is a need for a quick method that accounts for less time and better, consistent accuracy. Here is where we can use the advances made in Machine Learning to solve our problems. In the field, we can use models (already trained) to give us a quick result (a matter of seconds) of the class of this fracture.

We can also understand why it gives this model a particular label, i.e., which features/ area this model is considering.

Such a process will help speed up the process in most cases. In case of an ambiguous image/ sample, experts will be able to understand the reason behind the classification and prepare the model further.

3.4 Objectives

- Characterizing the fracture surfaces using SEM images and classifying them based on the fracture features
- Understand various Machine Learning models to classify the images features
- Provide assistance in identifying fracture surfaces by using image classification and Heat maps on Boron Modified Ti64 Dataset.

Chapter 4

Dataset

The dataset consists of SEM images of Boron modified Ti64 alloys at 0.0, 0.09, 0.3, and 0.55 weight% of boron, obtained via a three-point fracture test.

4.1 Image Cleaning

A total of 104 images were obtained from SEM of dimensions 1280 X 1024. The images were in BitMap format. These had to be converted manually to JPG to make accessibility and interaction much more effortless. These images were cleaned, and the index at the bottom of SEM images to denote scale and magnification was removed, which resized them to 1280 X 960. A few images captured in poor lighting were brightened, and blurry images were discarded.

4.2 Dataset Preparation

Due to such a low number of images, we divided each image into 12 parts of dimensions 320 X 320. During Dataset preparation, two kinds of datasets were created. The first dataset consisted of 3 classes: Dimpled, Cleavage, and Quasi-Cleavage. The rest of the images showed mixed characteristics and hence were irrelevant to the training of a classification model.

The second kind of dataset was created to take advantage of the two classes which had a clear difference in images, Dimpled and Cleavage. The rest of the images, which could not be classified entirely into these 2, were kept in the unsure ‘ class.

The images in the unsure class weren’t considered for training purposes. The models trained on the 2nd dataset were then used to predict the class of these unsure images, to understand the probabilities of each class, and what the model was considering to assign them these probabilities using heat maps.

In all the classes, especially in the first dataset’s Quasi Cleavage class and to some extent in both datasets’ Cleavage classes, some low magnification images needed to be discarded. Discarding

was done as it was increasingly difficult to assign a low magnification image to Cleavage or Quasi-Cleavage due to the indefiniteness of the observed features.

Partial and grainy images were also discarded, though plenty of efforts were made to keep the intraclass variation. For example, Clear low magnification images were kept, which could be assigned to a specific class by experts. Images were tilted, moved, rotated, and flipped (also mentioned below)) to make the model more robust.

4.3 First Dataset

In the first dataset, three classes were created.

1. Dimpled: Including dimpled regions of low and high density.
2. Cleavage: Including features such as lath markings, river markings, and steps.
3. Quasi-Cleavage: Including features of both the classes but no clear distinction or boundary between them.

The final number of images in the first dataset, in each class, were:

Dimpled	92
Cleavage	382
Quasi-Cleavage	218

Table 2.1

4.4 Second Dataset

In the second dataset, two classes were created.

1. Dimpled: Including dimpled regions of low and high density.
2. Cleavage: Including features such as lath markings, river markings, and steps.

The final number of images in the first dataset, in each class, were:

Dimpled	92
Cleavage	382

Table 2.2



(i)Dimpled

(ii)Cleavage

(iii)Quasi-Cleavage

Fig 1. Classification Dataset Visualisation

Chapter 5

Method and Results

5.1 Image Pre-Processing

As mentioned above, images obtained from SEM were cropped and divided into 12 parts of 320 X 320 dimensions. Unfit images (images that were blurry, dusky, or incomplete) were filtered manually. After we divided images into classes, they were then partitioned into Train (65%), Validation (15%), and Test (20%) folders. Each of these folders had subfolders of classes within them. This partition was done via a python script, which traversed through existing images, created new folders, and randomly pasted the image into its new folders in the ratio:

train:validation: test .65 : .15: .2.

Because the Dimpled class consisted of only 13% of the total images in the first dataset and 20% of total images in the second dataset, Image augmentation was necessary. A way to augment data is to transform the images in a small way and save this transformed image as a new image. This process is one of the older techniques present, demanding time and extra effort. In this project, we have used Keras' Image Data Generator class[18]. This class augments the images in real-time when the model is still training. Random transformations are applied to each image as it is passed to the model. Using the Image Data Generator class allows us to save memory and make the model more robust as the model considers each transformed image as a new one.

A few of the augmentation/processing techniques used were

1. Random rotations: This rotates the image randomly from 0 to the number of degrees provided. This feature helps the model become invariant to the orientation of the image.
2. Width / Height Shift: It shifts the pixels by the amount provided. This feature ensures that the model does not only focus on one particular part of the image.
3. Flips: Flips the image horizontally or vertically. Since our data does not exist in a particular orientation (unlike, e.g. cars and buildings), we can use the flipped images just as well.
4. Zoom: It either randomly zooms in or out of the image within the margin provided. This feature helps make the model more robust as all test images will not have the same magnification.

Examples of images after coming out of Image Data Generator class:

Original Image:

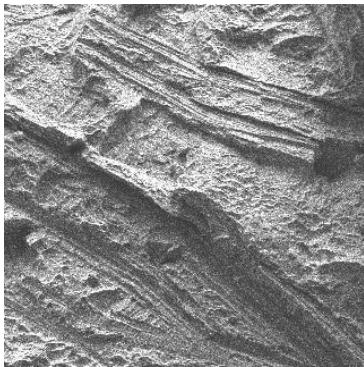


Fig no. 2.1

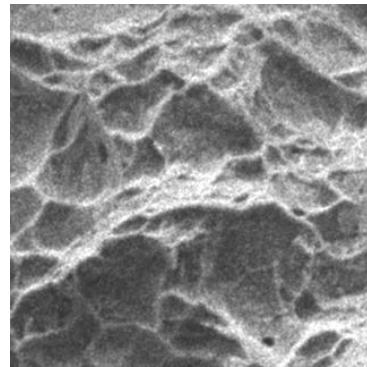


Fig no. 2.2

Transformed images:

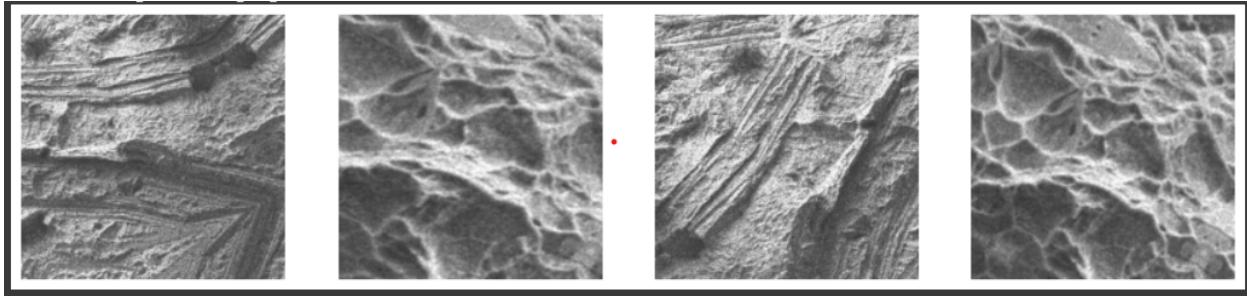


Fig no. 2.3

The Image Data Generator class also consists of a method called `flow_from_directory()`, which directly reads images from the directory, and expects images to be in the folders named after their respective classes, something we have achieved earlier via the use of the python script.

5.2 Loss function and Hyperparameter Tuning

Here is information regarding the loss function and a few of the hyperparameters used, and how the ideal values were found.

1. Loss function:

The loss function is a method to calculate the difference between obtained output and actual output. The ultimate goal of any machine learning model is to minimize the loss. For different problems, different kinds of loss functions are employed.

Since we are dealing with classification problems, we use **categorical cross-entropy loss**. Here is the equation of categorical cross-entropy (y_i : actual output of ith class in classification, y'_i : predicted output for ith class) :

$$Loss = - \sum y_i (\log(y'_i))$$

2. Hyperparameters

Since we are using standard architectures, the only hyperparameters which affect us are the ones that influence the training of the models[17].

a. Batch size

Various values of batch sizes were taken into consideration. It was found that lowering batch size worked better as the model's F1 scores began to improve, as well as the heatmaps began to show the areas we were expecting the model to focus on. Finally, a batch size of 5 was kept for all the models.

b. Optimizers

Adam optimizer was used in all our models. It is an extension of Stochastic gradient descent (SGD), but unlike SGD, this one updates the learning rate for each weight. Adam optimizer also requires minor optimization, has faster runtime, and low memory requirements.

c. Learning Rate

Adam optimizer has an adaptive learning rate, but we can provide one for the model to start from as a parameter. In all the models, the learning rate of 0.001 was employed as any higher rate resulted in low accuracy, and any lower increased the model's training time considerably.

d. Epochs

Various numbers were tried for the number of epochs, from 20 to 200, and via binary search, 100 was found to be a good fit.

5.3 Network Architectures

5.3.1 Introduction

5 Convolutional Neural Network architectures were used in the Classification. They were [15]

1. AlexNet
2. VGG16
3. VGG19
4. ResNet
5. InceptionV3

A short introduction regarding all of the above was given earlier. Same image processing was applied for all the models.

The Classification was decided on whichever class had the majority in the last layer predictions. No threshold was assigned.

In callback parameters, two callback functions were passed. One was for saving the best model based on loss, and the other for stopping the training if the loss does not improve for 20 consecutive epochs.

5.3.2 Transfer Learning

It is a method to use a pre-trained model (trained first on an existing dataset) on your dataset. The weights obtained from earlier training are saved. We can choose to fine-tune them again on the dataset or add more layers and train the model only on the newly added layers, keeping earlier layers frozen. Transfer Learning is an efficient method if we have less data and yields satisfactory results. The models for which transfer learning was employed had been trained on the Imagenet dataset.

5.3.3 AlexNet

5.3.3.1 Overview

We implemented AlexNet architecture from scratch, the only change being that, in the last layer, we put a number of neurons as two as we had only two classes to predict.

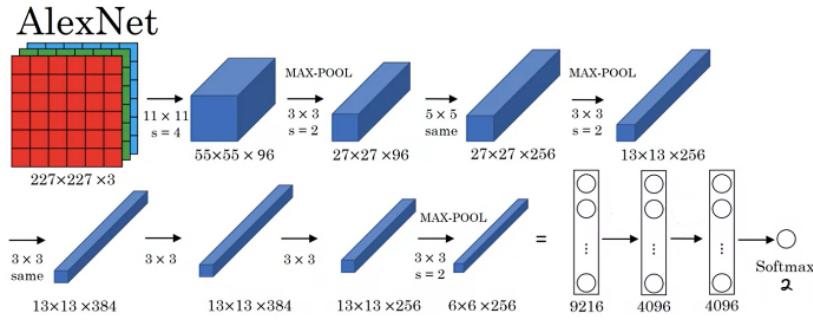


Fig 3. AlexNet Architecture. Inspired by [8]

We trained AlexNet on the 2nd dataset, in which it had to predict between 2 classes: Dimpled and Cleavage. Since we were implementing AlexNet from scratch, and due to the low amount of data (< 500 images), the model exhibited high loss and didn't exhibit good results.

The model's overall accuracy was 70.2%, but a more relevant metric, the f1 score, was 0.82 for Cleavage and 0.0 for dimpled. The loss was also very high ~.68. Here is the confusion matrix:

	Predicted Cleavage	Predicted Dimpled
True Cleavage	66	10
True Dimpled	18	0

Table no. 3

This shows that our model could not classify any dimpled image as dimpled.

Here is the heatmap, when the model was fed an image depicting a Cleavage fracture (left), and what areas of the image influenced the decision-making the most. (Red: most influence, Blue / Black: most minor influence) (The warmer the area is represented, the more impact it has on the prediction)

5.3.3.2 Analysis

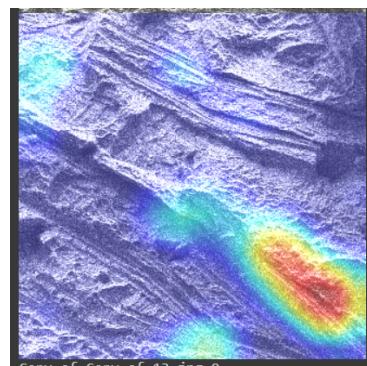
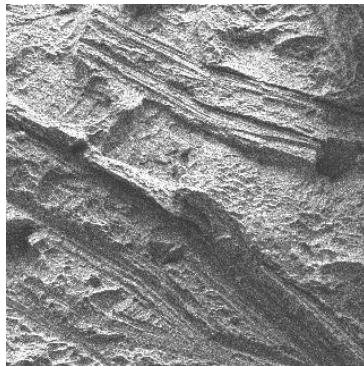


Fig 4.1 Actual: Cleavage Predicted: Cleavage Fig 4.2

Here, we can see that the model focuses on the parallel straight lines, which are hereafter called lath marks (more accurately, lath-like marks). These are not necessarily the surface marks left after the fracture occurs through lath structure, as the lath structures are around 10 microns in size, and here the lath marks go well beyond that range, as shown below (Fig 7).

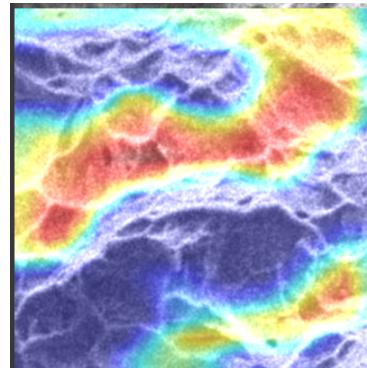
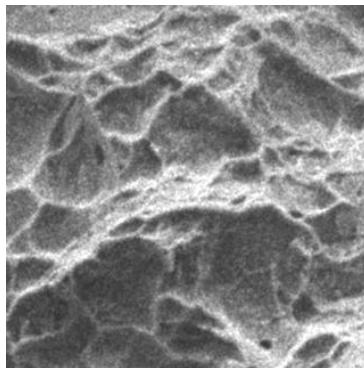


Fig 5.1 Actual: Dimpled Predicted: Dimpled Fig 5.2

In Figures 5.1 and 5.2, we can see that the model is not sure at all, with the probability of dimpled being .48. Here the model is skipping smaller dimple areas, and we hypothesize that it considers large dimpled areas as some kind of plane.

Figure 6.1 shows the image of the cleavage fracture surface that the model considers dimpled and its heatmap. This Heatmap (Fig 6.2) summarizes the problem with the current model. The model has not learned the key features of dimpled fracture through its supplied images.

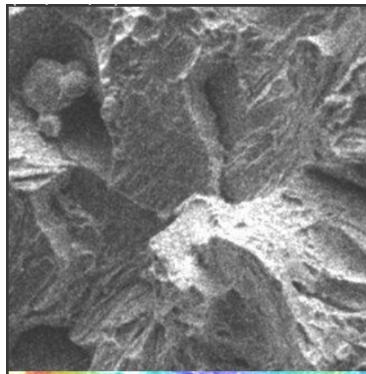


Fig 6.1

Actual: Cleavage Predicted: Dimpled Fig 6.2

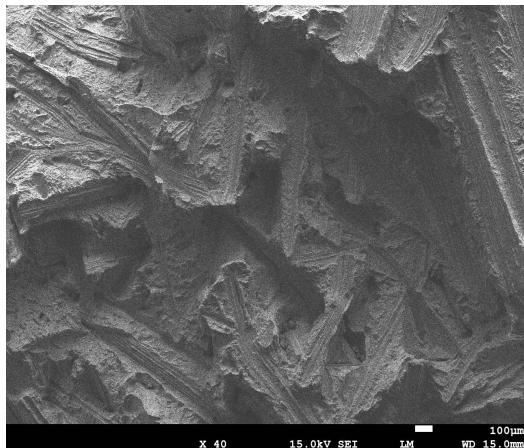
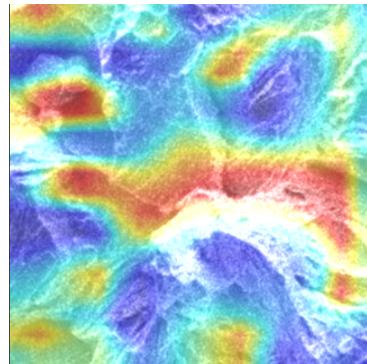


Fig 7. The cleavage image (Fig 5.1) is one of 12 images obtained from the following full-size image.

5.3.4 VGG16

5.3.4.1 Overview

We used a pre-trained VGG16 model for classification.

The model was trained on the Imagenet dataset. Only the convolutional base of VGG16 was taken, and these layers were frozen. Later a Global Average Pooling layer was added, along with a dense layer having 256 neurons. Lastly, each layer had a dense layer having a softmax classification function. VGG16 was the only model trained on both datasets. The two different models trained on different datasets only had the difference in the number of neurons in the last layer.

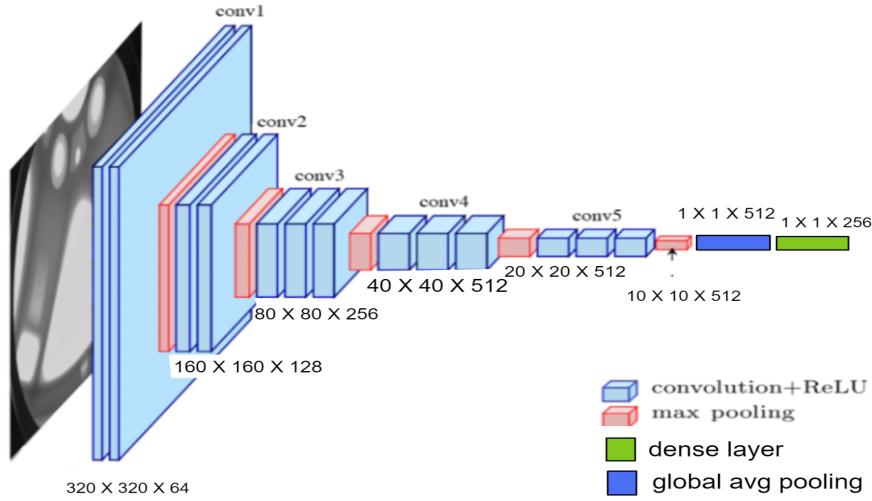


Fig 8 VGG 16 architecture. Inspired from [9]

5.3.4.2 First Dataset

The first dataset consisted of 3 Classes: Cleavage, Dimpled, and Quasi-Cleavage (QC). The final model exhibited a test loss of 0.38. The F1 score for the cleavage class was 0.87, for the dimpled class was 0.88 and for the QC class was 0.77. The overall accuracy of the model was 84%. Here is the confusion matrix

	Predicted Cleavage	Predicted Dimpled	Predicted QC
True Cleavage	66	0	10
True Dimpled	1	14	3
True QC	8	0	36

Table no. 4

Few of the highlights here are that the precision of the dimpled class is 1, i.e., it never predicted non-dimple image as dimpled. Another highlight is that many Cleavage images were predicted as QC and vice versa. This inaccuracy can be explained as there is less room for distinction between Cleavage and Quasi-Cleavage images. It was an arduous task even to create a Quasi-Cleavage class as the experts were also unsure of most of these images. Hence, it was decided to keep using the 2nd dataset to move forward for the following models. The model then used the QC and other unsure images to predict and understand its prediction based on HeatMaps.

5.3.4.2.1 Analysis

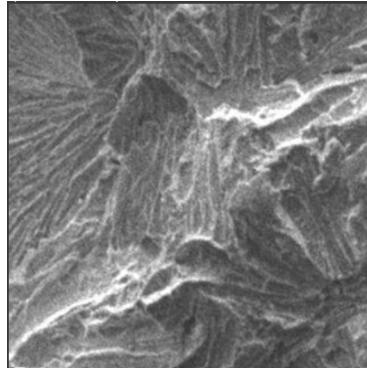
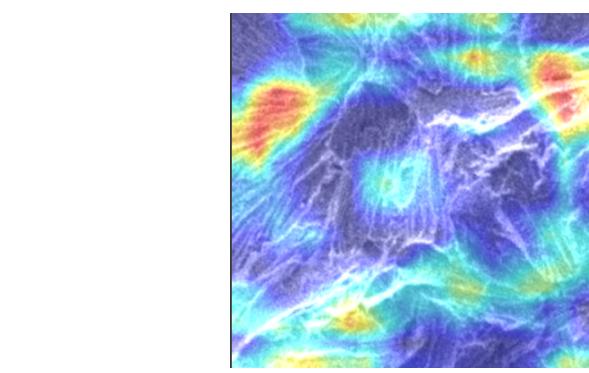


Fig 9.1



Actual: QC Predicted: QC

Fig 9.2.

As we can see, there are many river-marking steps highlighted here. We can also see some dimpled regions below being highlighted. These features must have led it to classify this image as Quasi-Cleavage. Here is where we run into a problem. The feature highlighted here can be a small part of any cleavage or dimpled image. Such scenarios would make the model understand the differences less due to low inter-class variations. The effects of these can be seen below, where the model incorrectly predicted the following images as QC

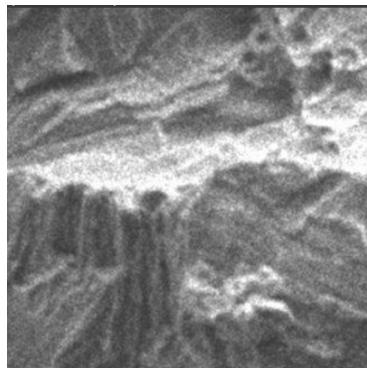


Fig 10.1

Actual: Cleavage Predicted: QC

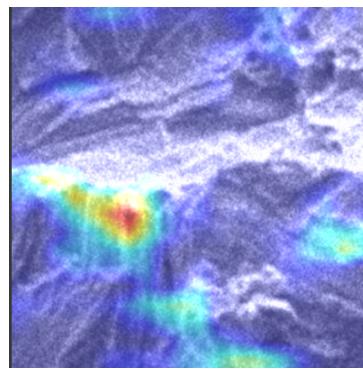


Fig 10.2

Here, we can see that the model has observed the same features as it observed above, a cluster of straight, parallel white lines, which can be called lath marks or just regular striations. Similar features make the overlap between cleavage and QC even larger.

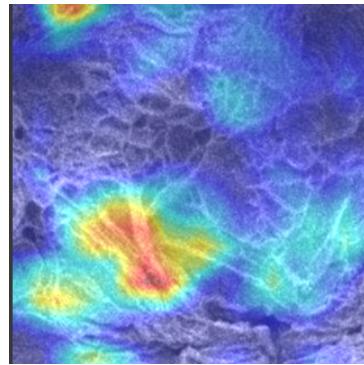
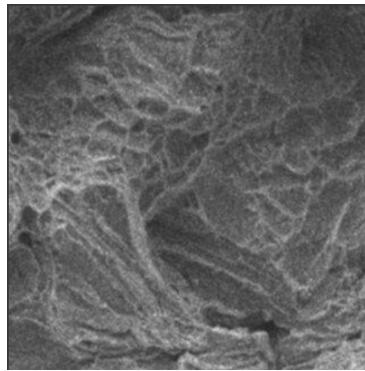


Fig 11.1 Actual: Dimpled Predicted: QC Fig 11.2.

Though this image is ambiguous, it has both dimpled ad cleavage features, the model focuses more on the cleavage features and has classified this image as QC.

5.3.4.2 Second Dataset

The Second dataset consisted of 2 Classes: Cleavage and Dimpled.

The final model exhibited a test loss of 0.07. The F1 score for the cleavage class was 0.99 and for the dimpled class was 0.94. The overall accuracy of the model was 98%. Here is the confusion matrix

	Predicted Cleavage	Predicted Dimpled
True Cleavage	75	1
True Dimpled	1	17

Table no. 5

The confusion matrix seems to show that our models are behaving ideally on the test set.

5.3.4.2.1 Analysis

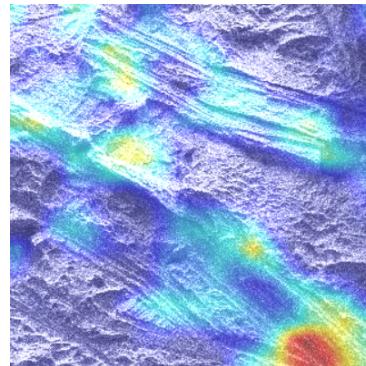
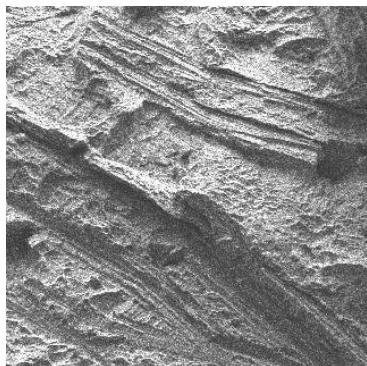


Fig 12.1 Actual: Cleavage Predicted: Cleavage Fig 12.2

Here, we can see that the model has predicted this cleavage fracture. The lath marks have been highlighted precisely.

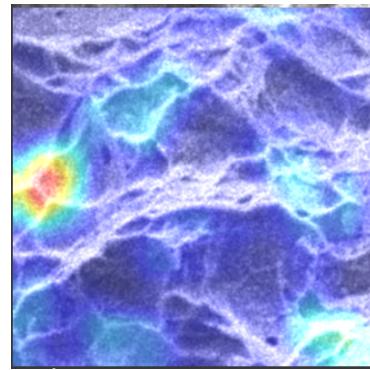
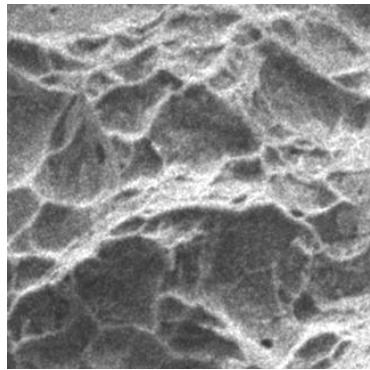


Fig 13.1 Actual: Dimpled Predicted: Dimpled Fig 13.2

Here, we can see that the model has correctly predicted the dimpled image. We can see that the model highlights the correct areas, but not confidently. The red highlighted area is of particular concern here as that is arguably the edge of 2 dimples and something very few experts would focus on. This might be the reason for the lower F1 score for dimpled images.

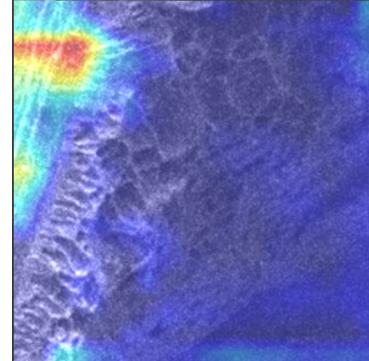
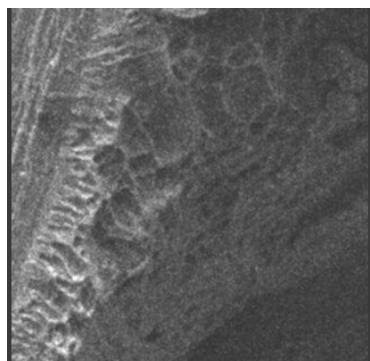


Fig 14.1 Actual: Dimpled Predicted: Cleavage Fig 14.2

The original image consists of both features, dimpled and cleavage. This image has been kept on purpose to understand what the model focuses upon when both features are present. Here, the original image was classified as dimpled, but we can see that the model focussed on the lath markings, classifying this as cleavage. It is interesting to note that the model did this with a 0.92 probability of cleavage.

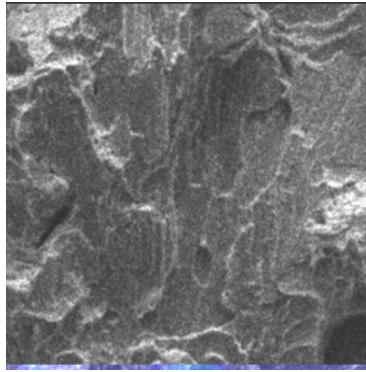
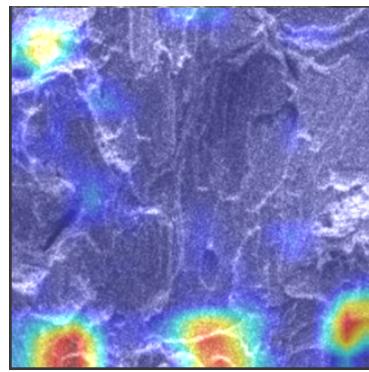


Fig 15.1



Actual: Cleavage Predicted: Dimpled Fig 15.2

This is where the model incorrectly predicted that this image belonged to the dimpled class. The more concerning fact here is that this is one of the more detailed images. The mode also focussed on the regions that seem to be cleavage steps. This case leads us to hypothesize that the model has learned only a few features of cleavage images and is considering the structure formed by edges in the bottom (red highlighted area) as dimples. The model also predicted this as dimpled with 0.87 probability.

Granted that the wrongly predicted images are only a minute percentage, it leads us to believe that even though the model's results are excellent, it might be focussing on a few wrong areas and might not have an overview of what is considered dimpled or cleavage.

5.3.5 VGG19

5.3.5.1 Overview

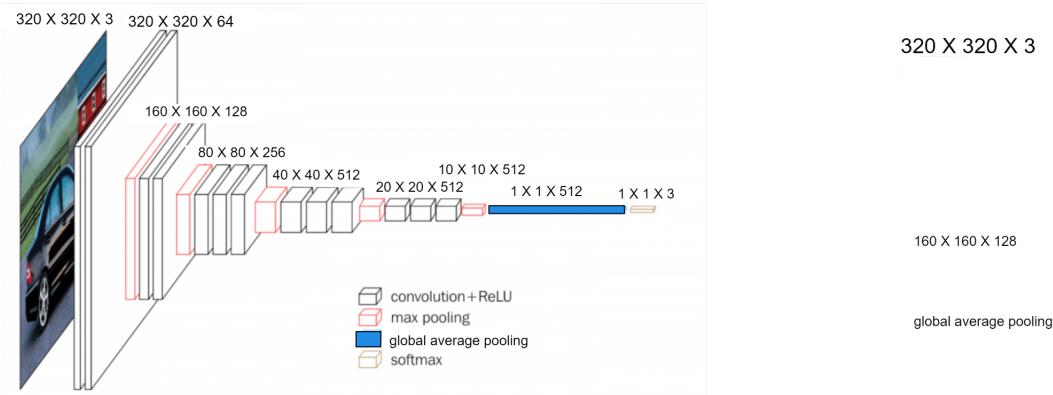


Fig 16 VGG19 Architecture. Inspired from [10]

VGG19, pre-trained on the Imagenet dataset, was fine-tuned on the 2nd Dataset. The final model exhibited a test loss of 0.12. The F1 score for the cleavage class was 0.98, and for the dimpled class was 0.91. The overall accuracy of the model was 97%. Here is the confusion matrix:

	Predicted Cleavage	Predicted Dimpled
True Cleavage	75	1
True Dimpled	2	16

Table 6

5.3.5.2 Analysis

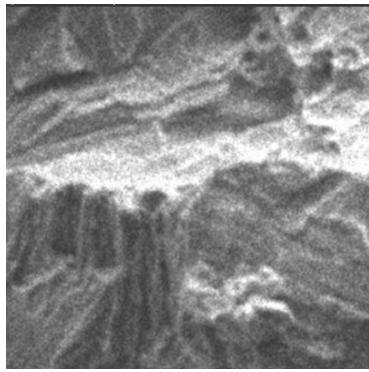
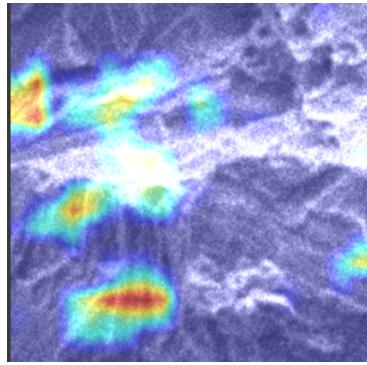


Fig 17.1



Actual: Cleavage Predicted: Cleavage

Fig 17.2

We can see that the model focuses on the right areas, such as the flat-surfaced, lath marks/striations. These highlighted characteristics are also common in the rest of the cleavage images, showing that the model has trained well.

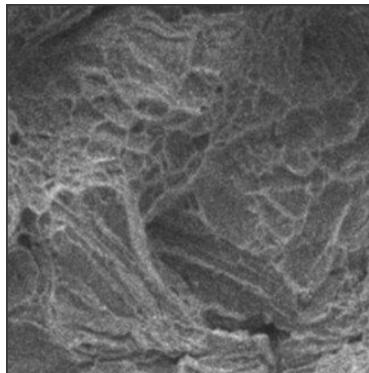
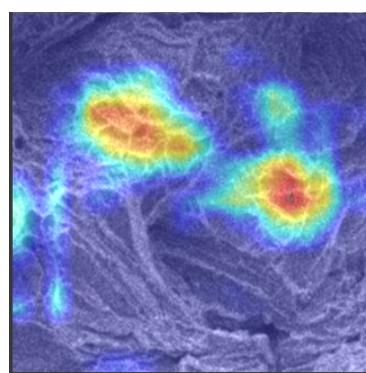


Fig 18.1



Actual: Dimpled Predicted: Dimpled

Fig 18.2

Here, we can see that the model has predicted this image as dimpled. Even though the image consists of a few cleavage-like features, dimpled areas are only highlighted.

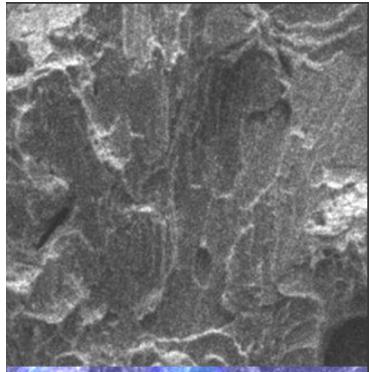
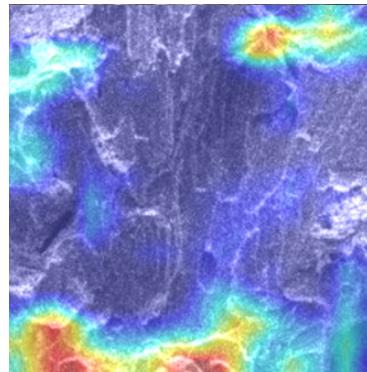


Fig 19.1

Actual: Cleavage



Predicted: Dimpled

Fig 19.2

Here, the model incorrectly predicts this image as dimpled. It focuses on the same areas as the VGG16 model. We hypothesize that it considers the highlighted area at the bottom to be some kind of boundary of a dimpled region. It is also interesting to note that the probability of dimpled here was close to 0.97.

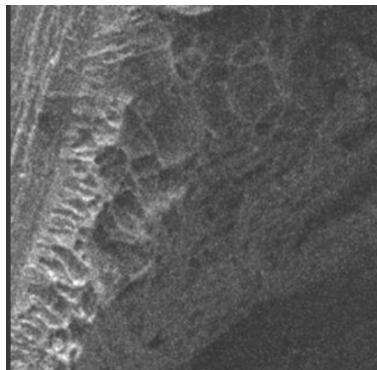
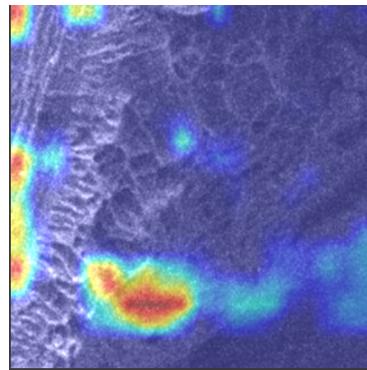


Fig 20.1

Actual: Dimpled



Predicted: Cleavage

Fig 20.2

Here, we have again fed the model an image of mixed features. We can see that the model has highlighted both the areas, dimpled and cleavage lath marks. It's interesting to note that the probability of cleavage (0.94) was way more significant than that of dimpled, causing the model to classify this image as cleavage.

5.3.6 ResNet50

5.3.6.1 Overview

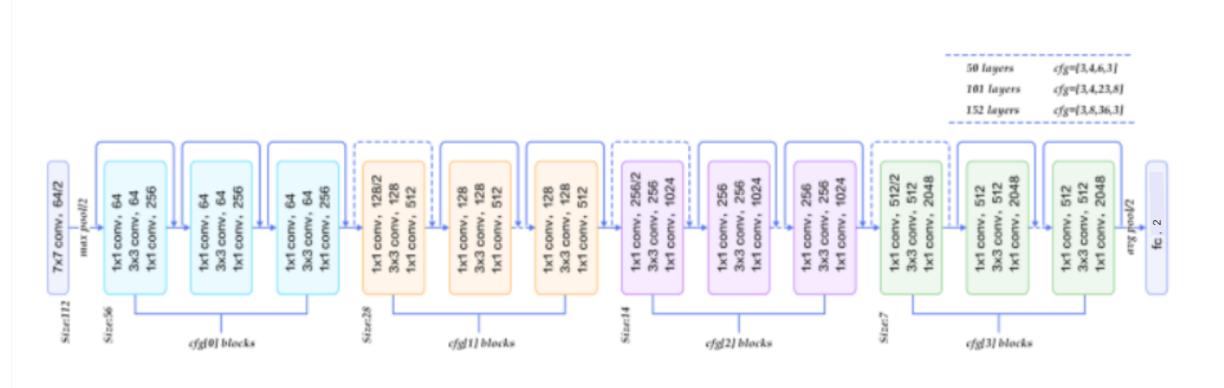


Fig 21. Resnet50 architecture. Inspired by [11]

The Resnet model was also pre-trained on the Imagenet dataset. The model was imported without the top part, and a global average pooling layer and a dense layer with softmax activation were added. Then the model was fine-tuned on the 2nd dataset. The fine-tuned model had a test loss of 0.117 and an accuracy of 98%. The F1 score for the cleavage class was 0.99, and for the dimpled class was 0.94.

Here is the confusion matrix:

	Predicted Cleavage	Predicted Dimpled
True Cleavage	75	1
True Dimpled	1	17

Table 7

The confusion matrix indicates that the model has trained well. An analysis of a few images was done using a heat map.

5.3.6.2 Analysis

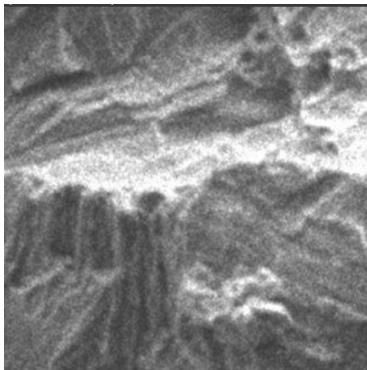
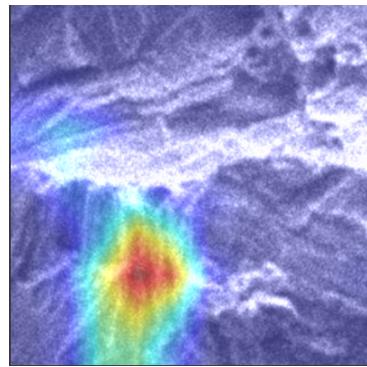


Fig 22.1

Actual: Cleavage



Predicted: Cleavage

Fig 22.2

Here, the model correctly predicts the image as Cleavage. The model focuses on the right areas, more prominently on the lath marks/striations.

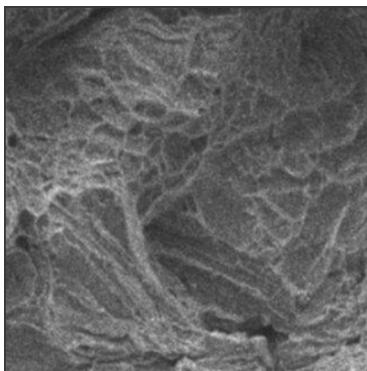
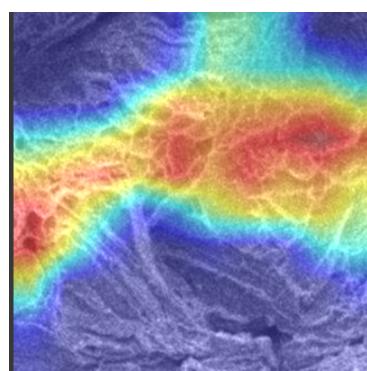


Fig 23.1

Actual: Dimpled



Predicted: Dimpled

Fig 23.2

Here, the model correctly predicts the image as Dimpled. We can see that the model is exclusively focused on the dimpled regions. The model does not consider all the steps/ striations. It may be because the cleavage features observed in this image are unlike the ones the model encountered while training, especially in the images depicting cleavage fracture.

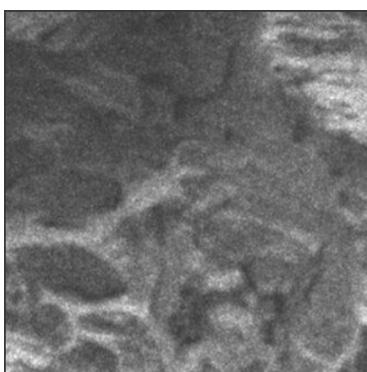
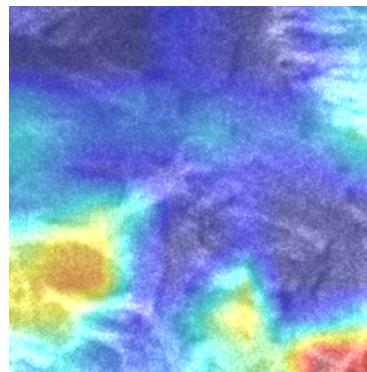


Fig 24.1

Actual: Cleavage



Predicted: Dimpled

Fig 24.2

Here, the model has incorrectly classified this image as dimpled. As we can see from the highlighted areas on the heat map, the model considers the cleavage facets as the dimpled regions. It can be attributed to the fact that while training, the model may not have encountered such images in cleavage class.

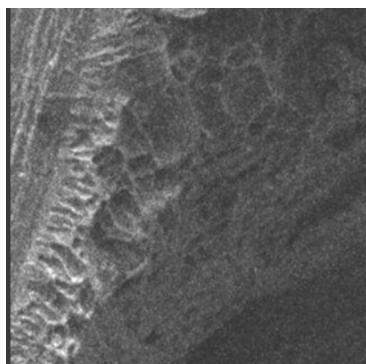
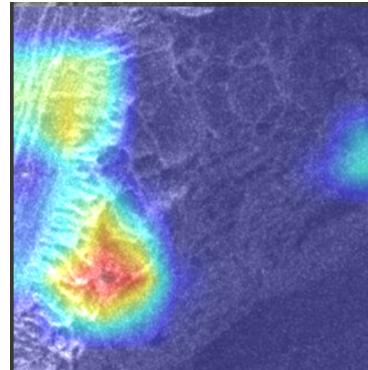


Fig 25.1

Actual: Dimpled



Predicted: Cleavage

Fig 25.2

Here, we can again see that both kinds of features have been highlighted as in the previous model. Interestingly, the model has predicted the image to be cleavage with 0.94 probability. It may be because the dimpled region is not properly illuminated at an angle.

5.3.7 InceptionV3

5.3.7.1 Overview

InceptionV3 is one of the more complex models. Here is a brief overview of the general architecture in the form of an image.

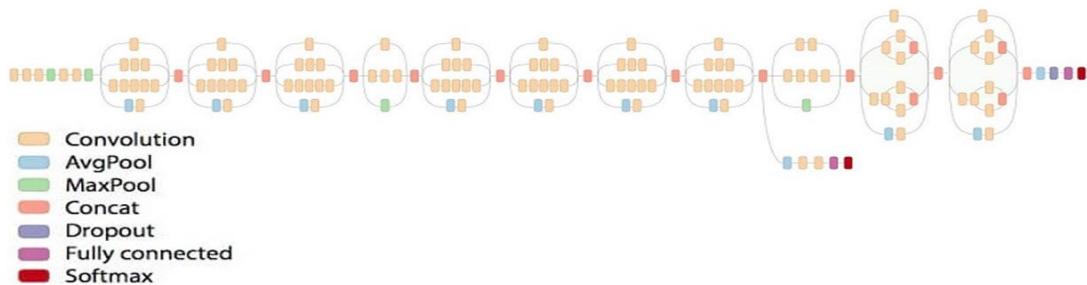


Fig 26 InceptionV3 architecture. Inspired by [12]

We imported the pre-trained (Imagenet) model and excluded the top. Global Average Pooling Layer and later dense layer with softmax activation function were added. The model was fine-tuned on 2nd dataset. The test loss achieved was 1.2, and the F1 scores for Cleavage and Dimpled were 0.89 and 0.56, respectively. Here is the confusion matrix

	Predicted Cleavage	Predicted Dimpled
True Cleavage	68	8
True Dimpled	8	10

Table 8

The model is performing above average in the Cleavage class but performs underwhelmingly in the Dimpled class. It can be understood by analyzing the results of some of the images classified.

5.3.7.2 Analysis

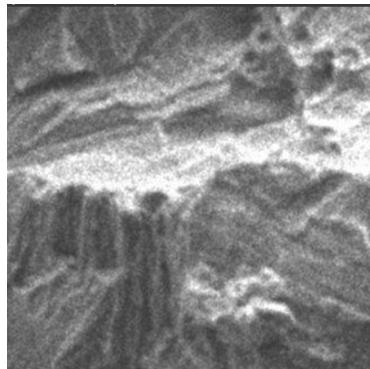


Fig 27.1

Actual: Cleavage Predicted: Cleavage

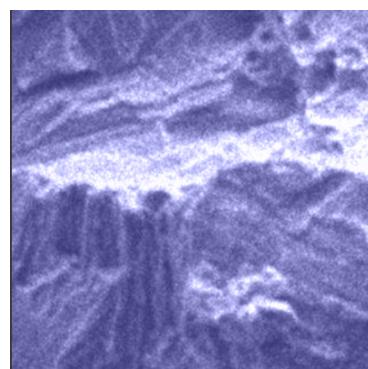


Fig 27.2

Here, the model has accurately predicted the image as Cleavage. However, the heatmap obtained here does not show any particular area that is being highlighted. It promotes less confidence in the model even when the results are accurate.

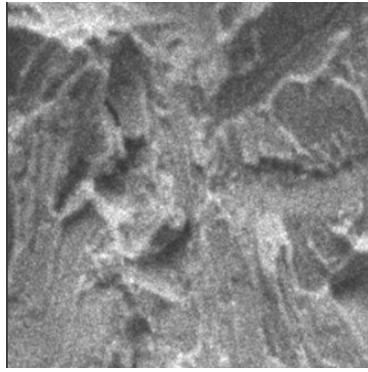


Fig 28.1

Actual: Cleavage

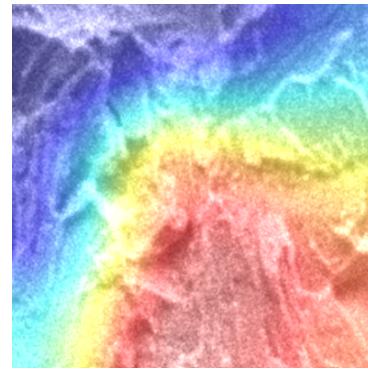


Fig 28.2

Here, the model has incorrectly predicted the image of the cleavage surface as dimpled. Even looking at the heatmap, it cannot be easily figured out the basis on which our model made this decision. It seems like the model has picked up different things from these classes than were expected.

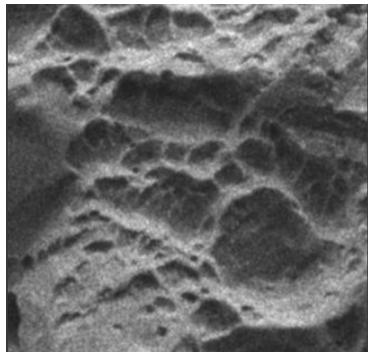


Fig 29.1

Actual: Dimpled

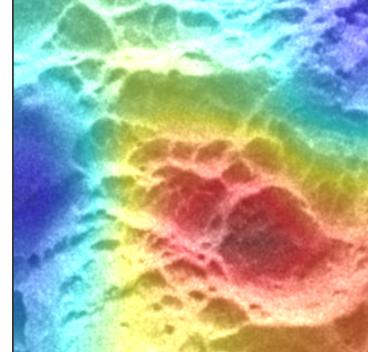


Fig 29.2

This seems like one of the model's accurate attempts. The model has correctly predicted this image as dimpled. We can see highlighted dimpled regions and flat surfaces, which represent cleavage-like features, ignored.

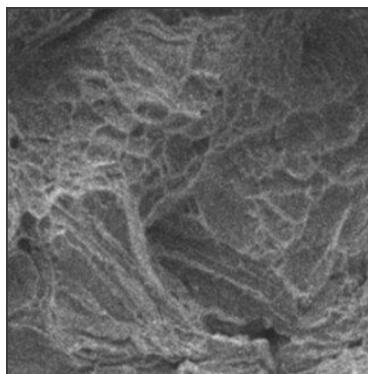


Fig 30.1

Actual: Dimpled

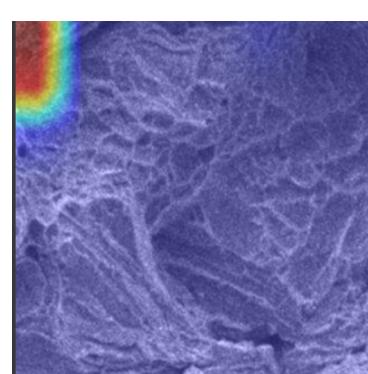


Fig 30.2

Here, the model has predicted this image as a Cleavage fracture with a 0.57 probability. If just observed based on the number, this decision seems to make sense considering the image has

characteristics of both the classes. However, upon inspection of HeatMap, we observe that the model is not focussing on the dimpled or cleavage characteristics. Rather it focuses on the top left area, which does not give us information regarding the type of fracture surface.

5.4 Final Results and Analysis

5.4.1 Final Result

Here is the list of different architectures used and the relevant metrics to describe them. (Note: This is concerning the 2nd dataset) (‘C’ represents Cleavage class and ‘D’ represent Dimpled class.). Another metric, Macro-Average F1, accounted for both F1 scores. It was defined as the Arithmetic mean of both F1 scores [16].

$$Overall\ Score = (F1(c) + F1(d))/2$$

	F1 score (C)	F1 score (D)	Macro-Average F1
AlexNet	0.82	0.00	0.41
VGG16	0.99	0.94	0.96
VGG19	0.98	0.91	0.94
ResNet50	0.99	0.94	0.96
InceptionV3	0.89	0.56	0.72

Table 9

Considering the macro-average F1 score, the best models, on the second dataset were Pre-Trained VGG16 and Pre-Trained ResNet50. Pre-Trained VGG19 was close behind, but Alexnet and Pre-Trained InceptionV3 showed poor results. AlexNet, in particular, shouldn’t even be considered, as it had an F1 score for dimpled as 0.

The VGG16 model was found to have better results than the VGG19 model, due to a small difference in the architecture. In VGG19, upon the convolutional base, a global average pooling layer and later dense layer with softmax activation were added. In VGG16 however, to improve results, another dense layer was added, consisting of 256 neurons between the global average

pooling layer (GAP) and the last dense layer. This is the reason that VGG16 based model defeats the VGG19-based model here.

In ResNet50 no additional dense layer was added (like the one in VGG16). Only the GAP layer and the last dense layer were added to the convolutional base. The success of ResNet can be attributed to the fact that it uses Residual Block. These are skipped connection blocks that allow the flow of information from early layers to later layers. This is what might have helped Resnet50 understand the features more clearly than the InceptionV3 model.

5.4.2 Analysis

VGG16 and ResNet50 were then used for the last part of the project, Classifying Unseen images into 5 classes Dimpled, Quasi-Dimpled, Evenly Mixed, Quasi-Cleavage, and Cleavage.

This was done in a similar way to the earlier method. Images were fed into the fine-tuned model, which classified the images based on their ‘proximity’ to the 2 classes these models had been trained on (Cleavage and Dimpled). The model output the probability of the image being in each class, and based on that probability, the images were divided into the 5 classes. Table no. 10 depicts the division of classes based on predicted probabilities. (Note: P_c represents the probability of the image to be in cleavage class, and P_d represents the probability of the image to be in dimpled class)

Class of Image	
$P_c \geq 0.8$	Cleavage
$0.8 > P_c \geq 0.6$	Quasi-Cleavage
$0.6 > P_c \geq 0.4$	Evenly Mixed
$0.4 > P_c \geq 0.2$	Quasi-Dimpled
$0.2 > P_c \geq 0.0$	Dimpled

Table 10

Here are some of the outputs obtained after unseen images were passed to these model and it had to predict the class (out of the 5 mentioned above).

5.4.2.1 VGG16

1. Cleavage

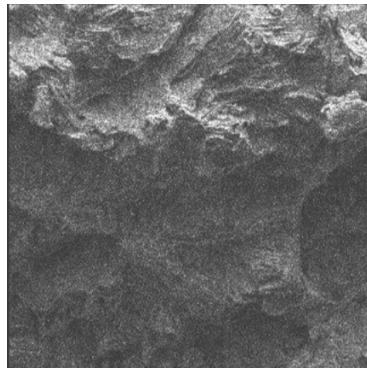


Fig 31.1

Here, the model has classified the image as cleavage due to the various steps / cliffs observed, along with a few lath markings and flat surfaces.

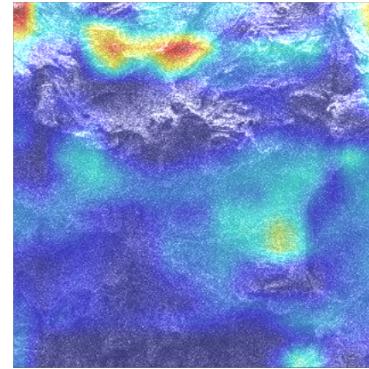


Fig 31.2

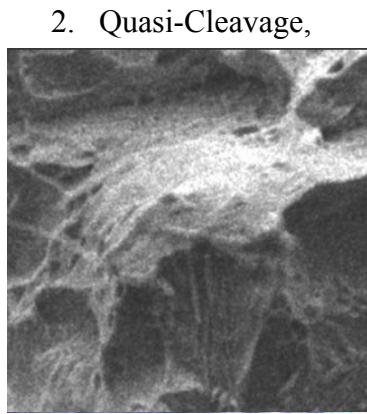


Fig 32.1

The model has classified this image as QC. This can be because of the highlighted step region and lath markings and a few dimpled regions near the middle of the image.

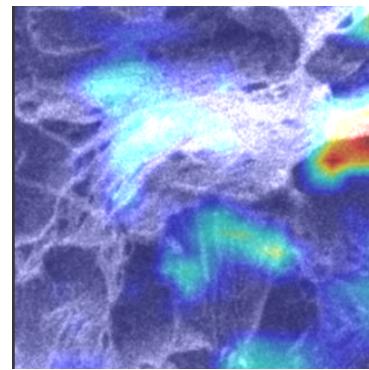


Fig 32.2

2. Quasi-Cleavage,

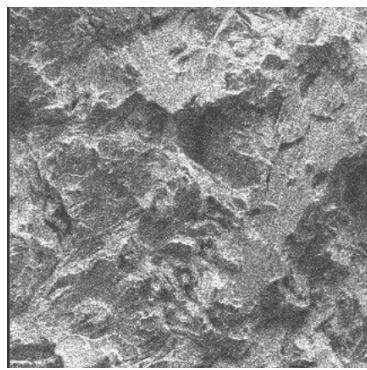


Fig 33.1

This image was classified as Evenly mixed due to the presence of flat surfaces, river markings, and dimpled areas highlighted.

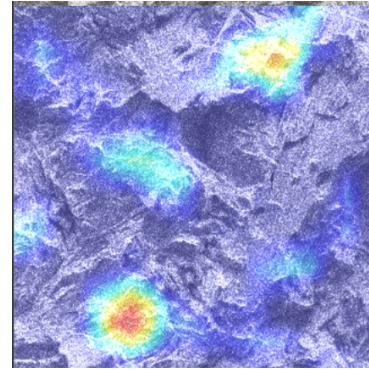


Fig 33.2

4. Quasi-Dimpled

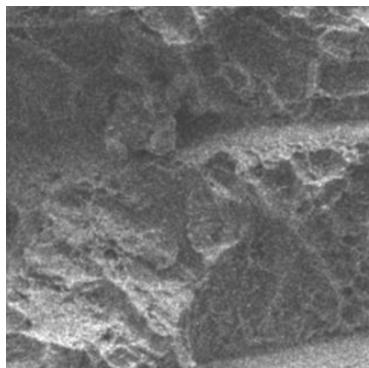


Fig 34.1

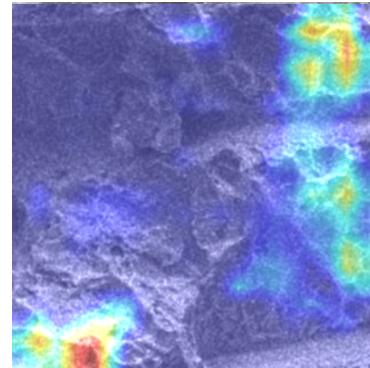


Fig 34.2

This image was classified as Quasi-dimpled. We can observe highlighted dimpled regions and a step surface in the image.

5. Dimpled:

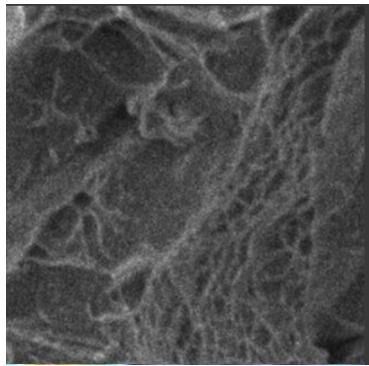


Fig 35.1

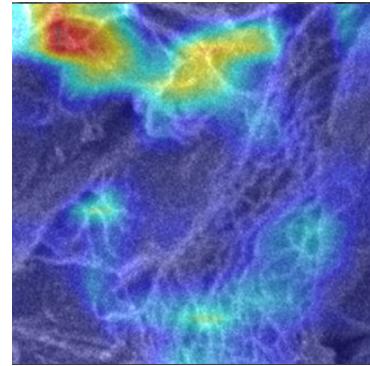


Fig 35.2

The model classified this image as dimpled due to the significant presence of dimpled regions throughout the image.

5.4.2.2 ResNet50

1. Cleavage

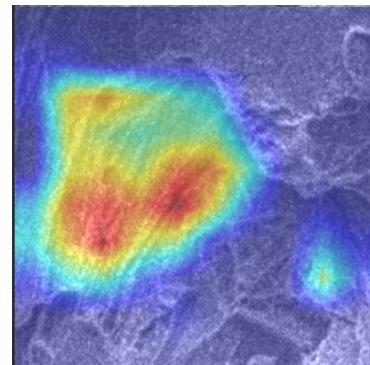
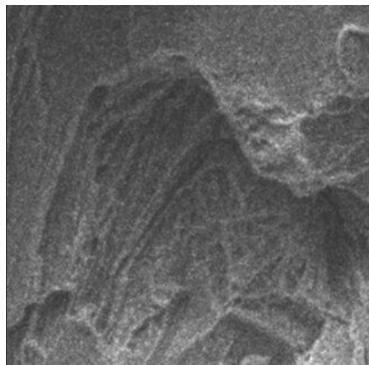


Fig 36.1

Here, the image was classified as cleavage due to the presence of significant lath markings along with flat surfaces and step/cliff region.

2. Quasi-Cleavage

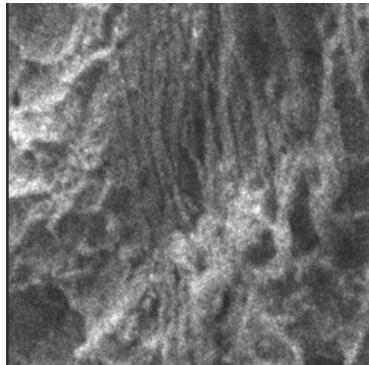


Fig 36.2

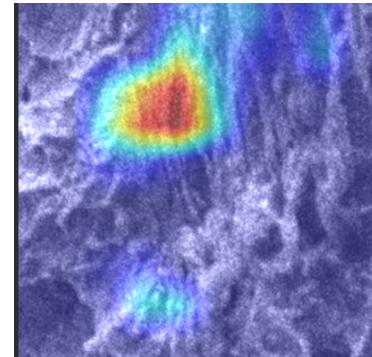


Fig 37.1

In this image, we can clearly see the presence of lath markings/ striations along with a few dimpled areas on the left side of the image.

3. Evenly Mixed

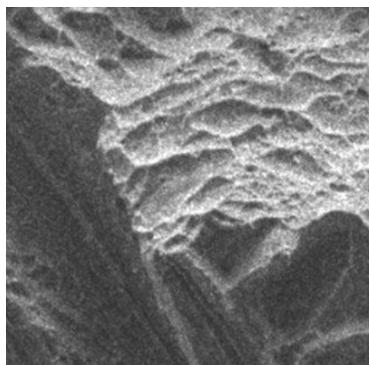


Fig 37.2

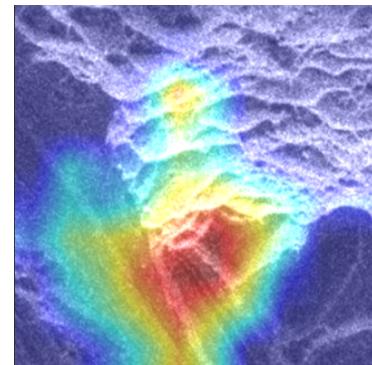


Fig 38.1

Here, we clearly see both dominant factors at play. The dimpled region is highlighted along with the cliff edge and lath markings.

4. Quasi-Dimpled

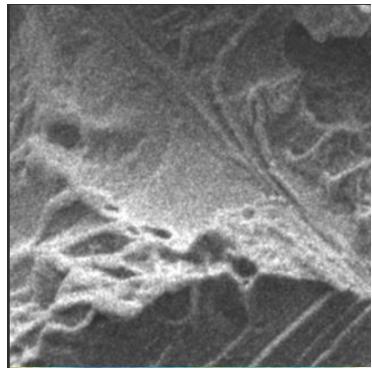


Fig 39.1

In this image, we mostly see dimpled regions with some prominent cliffs and lath markings.

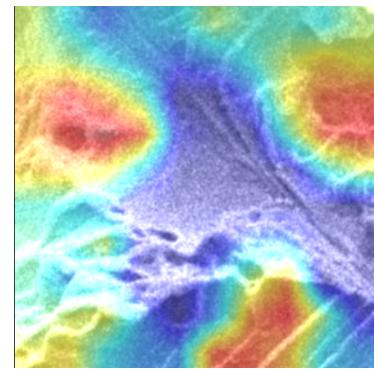


Fig 39.2

5. Dimpled

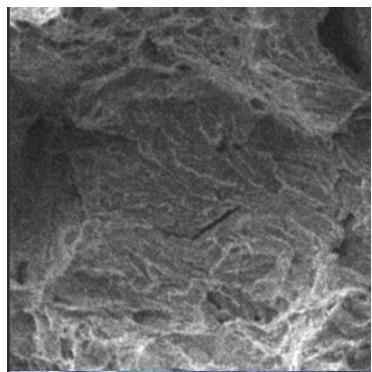


Fig 40.1

Here, the model highlights the dimpled regions but does not consider the middle region as having any resemblance to a feature of the cleavage class.

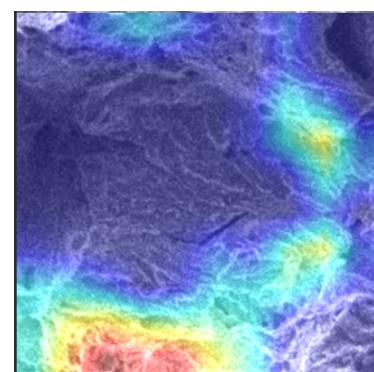


Fig 40.2

Chapter 6

Discussions and Future scope

6.1 Key Findings

6.1.1 Key Features

Through analysis of the results of top-performing models, a few hypotheses were made. The model considers lath markings, steps/cliffs, and river markings as strong indicators when identifying the cleavage class. Plane surfaces and cleavage facets are also considered relevant. When identifying dimpled class, the model looks for bright enclosed regions with dark interiors. The aspect ratio of dimples also matters. If the aspect ratio is closer to 1, the model is more likely to identify this as a dimple.

We hypothesize that the model identifies the white contours, and based on the pattern formed by them, it calculates the confidence score and determines the classes. E.g., if the model identifies straight, equidistant contours, it considers them as lath markings, or a single contour with space around it, it will consider as Step/ cliff edges. If the contour region is closed, enclosing a dark, hollow region, the model will likely consider it a dimple. When these contours form complex patterns, the model cannot exclusively identify features of a particular class. In such cases, the confidence score and the class are subsequently decided based on the quantity and quality of observed features.

6.1.2 Model Drawbacks

The top models can identify the contour patterns and display impressive metrics. Nonetheless, a few drawbacks and imperfections were observed in the model. E.g., the model could not classify a flat surface as a characteristic of cleavage fracture if there were no accompanying edges, steps, or lath marks. Similarly, sometimes a large cavity-like shape formed during intergranular fracture was classified as a dimple if the image was not correctly lighted or was dusky. Also, in the case where characteristics of both classes are visible in the image, i.e., lath marks/steps and dimpled region, different models gave priority to different regions.

6.1.3 Precautions

Some of the precautions that need to be taken to get the best results to include proper lighting, capturing images at a specific angle, and ensuring no edge cases are fed into the model. While training the model, keeping a certain level of interclass and intraclass variation is necessary. It is a fine line to walk, ensuring that the model identifies the difference between the classes and becomes robust. It is also essential to ensure that the model is trained on a large amount of data to learn from more diverse data and ensure credible results.

6.2 Challenges Involved

6.2.1 Dataset Size

After dividing the SEM images into 12 parts, we obtained around 1200 images. Out of 1200 only around 690 images could be classified as belonging to any particular class. These are the 690 images that were used to create the first dataset. Even while using the first dataset, there was an extensive overlap between two of the classes. To avoid this, we later reduced the dataset size to 2 classes consisting of 474 images. This is a very small number of images, considering we are using deep learning. As the amount of data decreases, so does the diversity of data, hence the model does not become as robust as it would otherwise have been.

6.2.2 Class Identification

We had earlier created a dataset consisting of 3 classes, but upon considering the overlap of features between two of the classes, we had to narrow it down to two, to get good results.

6.2.2.1 Interclass and Intraclass variation

During the preparation of the dataset, there was low interclass variation between the cleavage class and the QC class. This resulted in poor results as even the experts were not able to differentiate between some of the images belonging to different classes. In the second dataset,

there were images belonging to different magnifications, which were included to keep the model robust. This sometimes resulted in high intraclass variation and was a problem in some models.

6.2.2.2 Skewed Dataset

In the second dataset, dimpled images accounted for only 20% of images. This meant that the accuracy metric could not be trusted, and hence we had to rely on F1 scores. Class weights were included while training the model to encounter the low amount of dimpled data.

6.3 Future Scope

There are various ways to enhance the model's performance, the obvious one being increasing the size of the dataset. This is a common problem in the metallurgical field as obtaining data is time-consuming and inefficient. Increasing the diversity of data is also an important step. The more diverse the data, the more fractures will be encountered, enabling us to add more classes. This will increase the employability of the model.

The direct derivative of this work would be to use image segmentation. Using segmentation will be more practical as materials fracture via various mechanisms. This will be a challenging task considering creating classes with high interclass variation and low intraclass variation.

In practical work, the availability of data will always be a problem. Hence, the model trained on low amounts of data is meant to assist humans during the classification process. This assistance is meant to increase the efficiency of the process while still factoring in human knowledge as a cover. If the model is inaccurate on some image, that image can be labeled and added to the dataset, enhancing the model and making it more independent step by step. In such a way, as the dataset keeps on increasing and the model keeps on learning, it will slowly reduce the dependence on human intervention.

References

- [1] Bastidas-Rodriguez, M. X., Prieto-Ortiz, F. A., & Espejo, E. (2016). Fractographic classification in metallic materials by using computer vision. *Engineering Failure Analysis*, 59, 237–252. <https://doi.org/10.1016/j.engfailanal.2015.10.008>
- [2] Bastidas-Rodriguez, M. X., Polania, L., Gruson, A., & Prieto-Ortiz, F. (2020). Deep Learning for fractographic classification in metallic materials. *Engineering Failure Analysis*, 113(April), 104532. <https://doi.org/10.1016/j.engfailanal.2020.104532>
- Fractography Fractography. (n.d.).
- [3] Tsopanidis, S., Moreno, R. H., & Osovski, S. (2020). Toward quantitative fractography using convolutional neural networks. *Engineering Fracture Mechanics*, 231(March), 106992. <https://doi.org/10.1016/j.engfracmech.2020.106992>
- [4] Wang, S. Y., Zhang, P. Z., Zhou, S. Y., Wei, D. B., Ding, F., & Li, F. K. (2020). A computer vision based machine learning approach for fatigue crack initiation sites recognition. *Computational Materials Science*, 171(June 2019). <https://doi.org/10.1016/j.commatsci.2019.109259>
- [5] Konovalenko, I., Maruschak, P., Prentkovskis, O., & Junevičius, R. (2018). Investigation of the rupture surface of the titanium alloy using convolutional neural networks. *Materials*, 11(12), 1–13. <https://doi.org/10.3390/ma11122467>
- [6] Tamirisakandala, S., & Miracle, D. B. (2010). Microstructure engineering of titanium alloys via small boron additions. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 2(4), 168–180. <https://doi.org/10.1007/s12572-011-0033-z>
- [7] Lynch, S. P., & Moutsos, S. (2006). A brief history of fractography. *Journal of Failure Analysis and Prevention*, 6(6), 54–69. <https://doi.org/10.1361/154770206X156231>
- [8] <https://coursera.org/share/1fe2c4b8b8d1e3039ca6ae359b8edb30>
- [9] <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>
- [10] <https://towardsdatascience.com/extract-features-visualize-filters-and-feature-maps-in-vgg16-and-vgg19-cnn-models-d2da6333edd0>
- [11] <https://blog.devgenius.io/resnet50-6b42934db431>
- [12] <https://www.intel.com/content/www/us/en/developer/articles/technical/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic.html>
- [13] TITANIUM ALLOYS An Atlas of Structures and Fracture Features by Vydehi Arun Joshi
- [14] <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>
- [15] <https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception7baaaecc96>
- [16] All training codes have been implemented on Google Collaboratory notebooks using GPU.
- [17] <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-thehyperparameters-in-a-deep-neural-network-d0604917584a>

[18]https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

[19]Johari, Om. “Comparison of Transmission Electron Microscopy and Scanning Electron Microscopy of Fracture Surfaces.” *The Journal of The Minerals, Metals & Materials Society (TMS)*, vol. 20, no. 6, 1968, pp. 26-32. *SpringerLink*,
<https://link.springer.com/article/10.1007/BF03378720>.