# Instructions for labeling defect images

NextGen Summer 2018 Fellowship

Primary Editors

Leah Krudy : leah.krudy@hope.edu
Jacob Greaves : jrgreaves@wisc.edu

## 1.Introduction

This document provides detailed instructions for labeling the TEM images of irradiated FeCrAl alloy used for the defect detection machine learning project of the 2018 NextGen fellowship. The software and organizational methods, however, could easily translate to micrograph labeling of another class of defects or cubic materials. You may need to read this document and watch the videos multiple times to understand the process. In general, labeling involves a researcher making educated selections of defects using an image processing package and organizing the selections and defect classes in a reputable manner. We strongly recommend you have some calibration labeling practice before launching the final labelling process.

### Work Flow

Labeling defects takes time and is prone to human error. The following steps were taken to ensure accurately labeled micrographs crucial within the context of machine learning.

- A team of nine researchers viewed and discussed Kevin Field's labeling videos (linked in document) and decided on consistent labeling parameters
- Each team member was given 3 images to label individually using Fiji, an image processing software
- The team reconvened to discuss discrepancies in labeling choices and formed one unified version of labeled images
- The team was then split into teams of two to three people and given 8 images to label together
- Image labeling choices were cross-checked between teams as well as with more experienced labelers
- Note: the most effective labeling took place when 3 individuals worked together using an extra monitor and a mouse for added efficiency

Recording Fiji Measurements

- Store the numbered Fiji measurements in a text editor such as Sublime Text with the following naming convention
- Each number should be assigned a continuous range of the marked defects that correspond to that number ( see example below in yellow) these numbers should be tabulated in a .txt file that is saved to a name that follows the naming conventions that are listed below.

0 = 111 loops
1 = Black Dots
2 = 100 loops (all types)
3 = Possible pre-existing or maybe a defect
4 = Possible black dot or maybe 100 loop
5 = Defect of some type, but not sure what type (this ambiguity category does not include black dot/100 uncertainty such as #4

**Example of Correct Input of Fiji Measurements into text editor**

**note continuous number ranges, It is critical to label any given micrograph, one defect type at a time and record them on continuous non overlapping domains so that the .txt file is machine readable.*

```
0
1 38

1
72 154

2
39 71

3
155 158

4
159 173

5
174 176
```

## Output naming convention for file storage

While labelling an image it is up to the operator to adopt organizational methods for labeling that ensure the consistency and agreement of each defect label across all four of the documents that pertain to each micrograph image (see below). Each micrograph (when finished labelling all defects) should have:

Save **original image** (without overlay or numbering) as **photoName.jpg**

Save **“results”** output produced by ImageJ as **photoName_results.csv**

Save **labeled photo** with ellipse markings and numbers as **photoName_overlap.tiff**

Save **text file** with the number range for each type of defect as **photoName_log.txt**

# 2.Steps

## Step 1: Preliminary education on defects

Defects are small regions where there are no atoms. They can be thought of as clusters or vacancies and appear as shapes within the larger TEM image. Although some lines and marks are pre-existing defects, this document will explain correct identification for three major classes of defects appearing within irradiated FeCrAl alloy.

- *Labeled Image by Kevin Field:*

https://drive.google.com/drive/folders/1T9xZzDe9wnGKRFL5r3IS979jPkWygeoT

- Labeling Tutorial by Kevin Field

https://drive.google.com/open?id=1xbYssLUe7p6sYCa51Lzm8LYuYLCTw-At

- Express Labeling Tutorial by Kevin Field

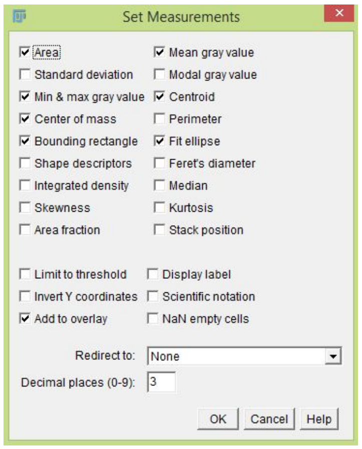
https://www.youtube.com/watch?v=OGHBY2XIjVs&amp=&feature=youtu.be

## Step 2 : Establish software for defect labeling

- Fiji(ImageJ) : Fiji is an image processing package—a "batteries-included" distribution of ImageJ, bundling a lot of plugins which facilitate scientific image analysis. https://fiji.sc/
- Sublime Text (or another text editor)

## Step 3: Manual Labeling of Defects

- Fiji Presets for Defect Labeling:
  Go to tool bar > Analyze > Set Measurements and make options match photo below



Set Measurements

| | |
|---|---|
| ☑ Area | ☑ Mean gray value |
| ☐ Standard deviation | ☐ Modal gray value |
| ☑ Min & max gray value | ☑ Centroid |
| ☑ Center of mass | ☐ Perimeter |
| ☑ Bounding rectangle | ☑ Fit ellipse |
| ☐ Shape descriptors | ☐ Feret's diameter |
| ☐ Integrated density | ☐ Median |
| ☐ Skewness | ☐ Kurtosis |
| ☐ Area fraction | ☐ Stack position |
| ☐ Limit to threshold | ☐ Display label |
| ☐ Invert Y coordinates | ☐ Scientific notation |
| ☑ Add to overlay | ☐ NaN empty cells |

Redirect to: None

Decimal places (0-9): 3

*Set Measurement" parameters for labelling in Fiji*

- Drag and drop jpg file into Fiji bar
- Select "**elliptical selections**"
- Enable **brush strokes**
- Begin selecting defects in the image using the cursor
- Select and fit overlay to each shape as closely as possible

**note*** ellipses/numbers will not appear on a tiff file unless the file is opened in Fiji

## Step 4 : Defect Type Recognition.

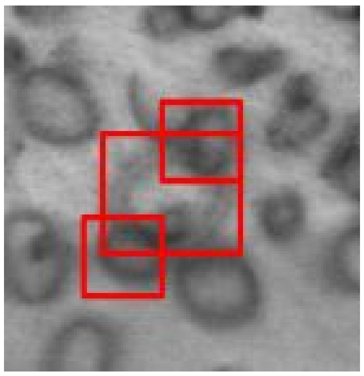
There are a number of different parameters to keep close in mind when deciding on classifying various types of defects in any given micrograph.

- Distance = measurement from end to end of longest side of shape
- Ratio = ratio of shorter diameter of ellipse to longer diameter of ellipse
- Contrast = difference in light intensity between two or more elements of a shape
- Orientation (for 100) = The directionality of the defect with respect to others of a similar type (This category only really applies to identifying 100 defects.)

Each of these conditions are important to for classifying any given defect. For any given defect on an image, one or more of these conditions may describe its features; however each condition may have more or less weight in determining defect type than others depending upon which defect type it may be. In general it is better to leave out a defect from the labelling set or utilize the ambiguity classes for defects if a consensus cannot be reached as to a defect's type. In this section the parameters listed are ranked in order of importance to characterization. Below are a number of scenarios that illustrate the difficulty in classifying defects.

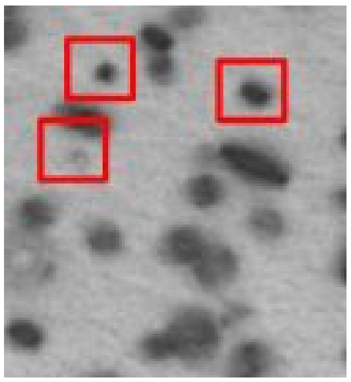
**Challenging Cases**

*Overlap*



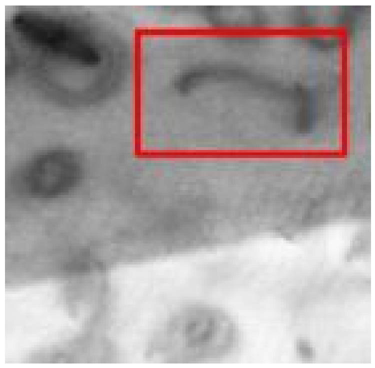
When defects overlap, if their different characteristics can be parsed they should be labeled. If not, confusing clusters of defects should not be included in the dataset.

*Small Defects*



Black dot and ambiguity classes should be used to characterize defects that are below the resolution of the micrograph.

*Partial Dislocations*

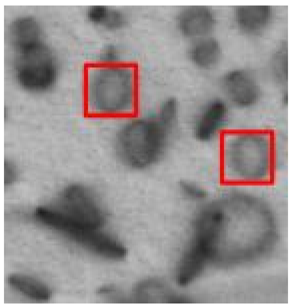


If more than 50 percent of a defect is present in a micrograph it can be included in the dataset.

---

## <111> (label FIRST in Fiji)

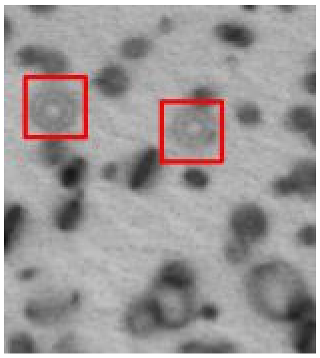
- Shape: Strictly Elliptical



- Outline: Sharp Black
- Contrast: light center
- Distance: approximately 60-120 nm
- Typical Ratio: 1.6

<111> defects are in general ought to be the largest and easiest defects to classify. They are characterized by a strictly elliptical structure outlined by a solid black or gray line. These defects may fall on any orientation with respect to one another.

## <100 Type 1> (label SECOND in Fiji)

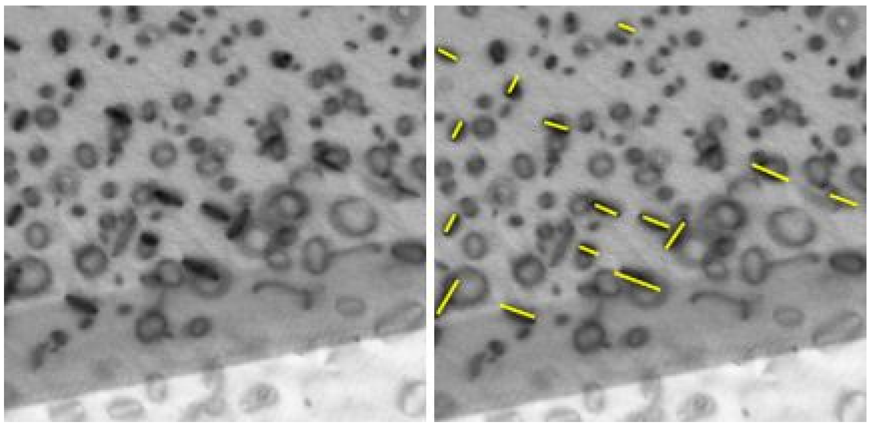


- Shape: Square or Circular, but usually appears as a thick dark line OR a square
- Ratio: (1:1)
- Contrast: Light in comparison to the background
- Positioning: depending on the angle at which a electron microscopic image is taken

<100> Defects can have a number of different characteristics based upon their thickness, orientation and size. <100> defects are defects that occur parallel to cubic crystal faces and as such they must fall 90 degrees to one another. As such there are a number of different "types" of <100> defect that may appear in any given micrograph.

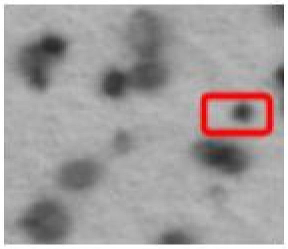
## <100 Type 2> (label SECOND in combination with type 1 unless otherwise specified by PI)

- Dark line/highly distorted ellipse, high contrast.
- 0 or 90 degree Orientation with respect to other *type 2 <100> defects



*Yellow lines indicate Type 2 <100> defects at 0 or 90 degree orientation.*

## <Black Dot>


*Black Dot Defect*

- Shape: varies. often a circle or ellipse
- Contrast: usually the darkest pixel regions in the image
- Positioning: Black dots can appear on their own, within a 111 or 100 loop, or intersecting with the line of a loop.

Black dot defects do not represent any real physical defect, in reality they are some form of <111> <100> or pre existing defect that is too small to be accurately characterized. But due to the resolution limitations of the microscope we cannot always definitively label each defect. Black dots are often the largest defect category

## <Ambiguity Categories>

Although black dots will encompass a large portion of unclear defects, we also used three ambiguity categories to classify defects that do not fall cleanly into the three major categories. These include:

- Possible pre-existing defect or maybe a classifiable defect
- Possible black dot or maybe 100 loop
- Defect of some type, but not sure what type (this ambiguity category does not include black dot/100 uncertainty)

## Appendix

### 1.How to use program to generate bounding box from the labeling log files? （Dataset Generation）

Link: https://drive.google.com/drive/u/1/folders/1udTgqHIPqocWG50VxhaTXQOsD2nClreK

Instructions:

1. Open Bounding-Boxes.py file
2. Set the variables at the top of the file marked as “TODO”:
   a. listofFiles: path to txt file that contains the name of files that need to be parsed
   b. csv_filepath: path to folder containing csv files
   c. log_filepath: path to folder containing log files
   d. output_directory_path: path to folder where bounding box files should be saved
3. In your Terminal, navigate to directory containing Bounding_Boxes.py and run:

```
$python Bounding_Boxes.py
```

4. The format of the bounding box file is **[index][xmin][ymin][xmax][ymax][label]**
**Label can be optional.**

### 2. Bounding Box plotting

The BBPlotter.py script can be used to draw ground truth or generated  bounding boxes on images.

Link: https://drive.google.com/open?id=1yhM46s1DqWsc0BLLtGDSdF1hvAApitL7

Instructions:

1. Open BBPlotter.py file.
2. Set the variables at the top of the file marked as “TODO”:
   a. bboxes_filepath: Path to CSV file containing bounding boxes.
   b. images_directory_path: Path to directory containing unlabeled images.
   c. output_directory_path = Path to directory you wish to save labeled images.'
   d. col_image_filename: Column in CSV file which contains image name. (default=0)
   e. col_defect_type: Column in CSV file which contains defect type. (default=1)
   f. col_xmin: Column in CSV file which contains xmin value. (default=2)
   g. col_ymin: Column in CSV file which contains ymin value. (default=3)
   h. col_xmax: Column in CSV file which contains xmax value. (default=4)
   i. col_ymax: Column in CSV file which contains ymax value. (default=5)

   j. bbox_color: Color in RGB format (example: #FF0000 for red) used to draw bounding box. Useful link: https://www.w3schools.com/colors/colors_rgb.asp
   k. linewidth: Weight (thickness) of bounding box outline (default=20)
3. In your Terminal, navigate to directory containing BBPlotter.py and run the following command:

```
python BBPlotter.py
```

Bounding box (bbox) data should be contained in a CSV file as follows.

1. Each bbox is identified by the image_filename to which it belongs, the defect_type that it identifies and the delimiting values xmin, ymin, xmax and ymax.
2. Each row contains information for one bbox.
3. Row format: [image_filename], [defect_type], [xmin], [ymin], [xmax], [ymax]
4. All bbox rows for a single image should be grouped contiguously otherwise multiple instances of the image will be opened