To,

IITD-AIA Foundation od Smart Manufacturing

Subject: Weekly Progress Report for Week-4

Respected sir,

June 27:

Familiarize my self with libraries and frameworks, such as OpenCV and TensorFlow.
(These libraries provided a set of pre-implemented functions and tools for various computer vision tasks, enabling efficient development and implementation of the defect detection system)

Understood how to make changes in an image and change its colour dimension.

Learned basic image manipulation , enhancement.

Challenges:

Integrating these libraries into the development environment and dealing with potential compatibility issues posed additional challenges.

Gone through reserch paper: https://asianssr.org/index.php/ajct/article/download/783/629

June 28: Familiarize my self with Computer Vision .

Study about Computer Vision though research paper and articles .
Gone Through video lectures.
With this i also explore Computer Vision Algorithms .

Went through this article : https://www.ibm.com/topics/computer-vision#:~:text=Computer%20vision%20is%20a%20field,recommendations%20based%20on%20that%20information.

June 29: Learned about Eda and its implementation on data set and continuing learning YOLOV5 (You Only Look Once) .

The steps of EDA are..
Here are six key steps to conduct EDA:

- **Observe dataset.**
- **Find any missing values.**
- **Categorize values. ...**
- **Find the shape of dataset. ...**
- **Identify relationships in dataset. ...**
- **Locate any outliers in dataset.**

Reference :

https://www.youtube.com/watch?v=ag3DLKsl2vk

https://www.digitalocean.com/community/tutorials/exploratory-data-analysis-pytho

-

June 30: Learned about Eda and its implementation on data set

Learning about Defect Extraction Algorithm

Continue the resize image to 128*128 using PIL library and create a new file for resize images, converting images to grayscale and try to applying specific feature extraction techniques.

```
from PIL import Image

import os

directory = r'C:\Users\Nandan\Desktop\fsm\dataset\perfect'

for file in os.listdir (directory):

    if file.endswith(('jpeg', 'png', 'jpg')):

     filepath = os.path.join(directory, file)

     outfile = os.path.join(directory, 'resized_'+file)

      with Image.open (directory+'/'+file) as im:

       im.thumbnail ((128, 128))

         im.save(outfile)
```

Reference :

https://www.youtube.com/watch?v=ag3DLKsl2vk

https://youtu.be/LB9SklRNDUA

July2:

Extracting the feature and Apply defect cropping on dataset images using OpenCV. OpenCV has implemented a function cv2. goodFeaturesToTrack() which is very useful when we don't need to detect every single corner to extract information from the image

```
import cv2

import numpy as np

def detect_defects(image):
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    binary_image= cv2.threshold(gray_image, 0, 255, cv2.THRESH_BINARY)
```

```
contours, _ = cv2.findContours (binary_image,
cv2.RETR_EXTERNAL, cv2.CHAIN_ APPROX
_SIMPLE )

defect_mask np.zeros_like(gray_image)

cv2.drawContours (defect_mask, contours, -1,
255, thickness= cv2.FILLED)

defect boxes = []

for contour in contours:

x, y, w, h =cv2.boundingRect(contour)
defect_boxes.append((x, y, x + w, y + h))

return defect _mask, defect_ boxes

def crop_ defects(image, defect_boxes):

cropped _images = []
```

```
for box in defect boxes:

x1,y1, x2, y2= box

cropped_ image= image[y1:y2, x1:x2]
cropped_images.append(cropped_image)

return cropped_ images

# Load the image

Image=
cv2.imread(r"C:\Users\Nandan\Desktop\fsm\de
fected\o.jpg")

if image is None:

print("Error: Failed to load image.")
exit()

defect _mask, defect _boxes =detect _defects
(image)
```

```
cropped _images= crop defects (image,
defect_boxes)
 for 1, cropped_ image in
enumerate(cropped_Images):
cv2.imshow(f"Cropped Defect{i+1}",
cropped_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

}
```