My understanding of INTP23-ML-9

Piston Defect Detection Using Computer Vision

Piston Defect Detection is one of the most important aspects of production. With the rise of piston demands for safety , they demand for higher quality components is rising.This is where Computer Vision can provide an efficient alternative for commercial automated optical inspection to assist small scale manufacturers to do an automated inspection.

20bic039@nirmauni.ac.in
Priyal Salecha

Week 5:

Whats happening this week:

- LazyPredict Learning
- Pre Processing
- Visualisation

Understanding: The goal of the research is to develop a computerised system that can automatically check for and find piston manufacturing problems. This system will use computer vision techniques to look for any indications of flaws in photographs of pistons by analysing and interpreting digital images or videos.Images of pistons will be recorded using cameras or other imaging equipment during the production process. The automated system will then process these photographs and analyse the visual data using cutting-edge algorithms and methods.The automated system for flaw detection in pistons that uses computer vision techniques will boost quality control procedures, reduce the output of defective pistons, and increase manufacturing efficiency all around.

Approach:

After using multiple images we can classify images as defect images and without it this can be done by image processing and classifies by computer vision.

Weekly Progress:

7th July

Work done:

- Regression Using DataSet

6th July

Work done:

- Exploring LazyPredict
- PreProcessing and Visualisation done

On 9/07

Work done:

- Working on the code with different approach
- Pre processing and heat map

+ Code + Text

```
[ ] from google.colab import drive #importing the drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ] import cv2
    import numpy as np
    import os

    # Pre-processing parameters
    image_size = (256, 256)  # Desired image size for processing
```

```
def preprocess_dataset(content/drive/MyDrive/Normal):
    # Create a list to store preprocessed images
    preprocessed_images = []

    # Iterate over each image in the dataset
    for filename in os.listdir(/content/drive/MyDrive/Normal):
        if filename.endswith(".jpg") or filename.endswith(".png"):
            # Load the image
            image_path = os.path.join(/content/drive/MyDrive/Normal, Normal)
            image = cv2.imread(/kumda_component1.png)
```

NORMAL DATASET