Industry Project Report

On

Real Time Defect Detection and Classification Using AI Techniques

Developed By: -
Jainam Shah (18162121033)
Het Mehta (18162121008)
Krutik Shah (18162121035)
Jaspreet Singh Pal (18162121014)

Guided By: -
Prof. Ravindra Patel (Internal)
Punit Lalwani (External Guide)
Mr. Yagnesh Vyas (External Co-Guide)

Submitted to

Department of Computer Science & Engineering

Institute of Computer Technology

Year: 2022

CERTIFICATE

This is to certify that the **Industry** Project work entitled **"Real Time Defect Detection And Classification Using AI Techniques"** by Jainam Shah(Enrolment No.18162121033) of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CBA/BDA/CS) Department at Ganpat University Institute of Computer Technology. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of Internal Guide

Name & Signature of Head

Place: ICT - GUNI

Date:

ACKNOWLEDGEMENT

Industry Internship project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank Dr. Hemal Shah, Principal, ICT, and Prof. Dharmesh Darji, Head, ICT who gave us an opportunity to undertake this project. My grateful thanks to Prof. Ravindra Patel & Mr. Yagnesh Vyas (Internal & External Guides) for their guidance in project work “Real Time Defect Detection and Classification Using AI Techniques”, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without their help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

JAINAM SHAH (Enrollment No:18162121033)

ABSTRACT

In most cases visual inspection of an object when examined by a person considering proper dimensions of an object is a difficult task. This project aims to study data modeling, Machine Learning (ML) model - neural networks (NN) modeling and reliability of such models for automatic detection and classification of defects of objects. In order to classify the defects, present in an object during examination the concept of neural networks comes in handy as it is able to identify defects and classify them based on their types. After providing a perfectly balanced image to the neural network the proposed model will examine whether the testing images are normal or they are filled with defects and will also classify the types of defects the rest of the objects are possessing.

INDEX

CHAPTER: 1 INTRODUCTION

CHAPTER 1 INTRODUCTION

Visual inspection of an object by a person, in most cases not possible because of the error in eyesight and precision to recognize an object properly. In recent times, only video monitors and video recorders have been used where a person can check on-line or taped video sequences for defects. In this way, only small parts of an object i.e. top or bottom side are viewed.

Additionally, this visual inspection is subjective and dependent on a large number of human factors such as the problems of working longer hours, attention being drawn to other events, subjectivity in terms of defect severeness assessment. Developing automatic detection and classification of surface defects of objects has been really a challenging problem. An automatic detection and classification system require knowledge of data concerning the current state of the objects, and a methodology to integrate various types of information into decision-making process of evaluating the quality of the product. The need for surface detection technologies for surface defect classification has long been recognized. Real time image processing typically involves the application of high-speed camera, which may give the defect images in real time. The large amount of information is gathered during this process in the form of images. Human experts evaluate this information and give the level of defects, types of defects of the objects.

The main focus of this project is to utilize the perfect image of an object with proper dimensions and make effective use of this data in the development of successful application using various new technologies.

CHAPTER: 2 PROJECT SCOPE

CHAPTER 2 PROJECT SCOPE

The scope of the created project thus extends to the amount of data provided by the user in the form of image to the model in order to train the model and provide necessary results and is limited to desktop/web application respectively.

CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS

CHAPTER 3 SOFTWARE AND HARDWARE REQUIREMENTS

Minimum Hardware Requirements

Processor	2.0 GHz
RAM	4GB
HDD	40GB

Table 3.1 Minimum Hardware Requirements

Minimum Software Requirements

Operating System	Any operating system which can support an internet browser.
Programming language	Python
Other tools & tech	Visual Studio Code, Jupyter Notebook, Web Camera, HTML. CSS. JavaScript, Flask, MongoDB Compass

Table 3.2 Minimum Software Requirements

CHAPTER: 4 PROCESS MODEL

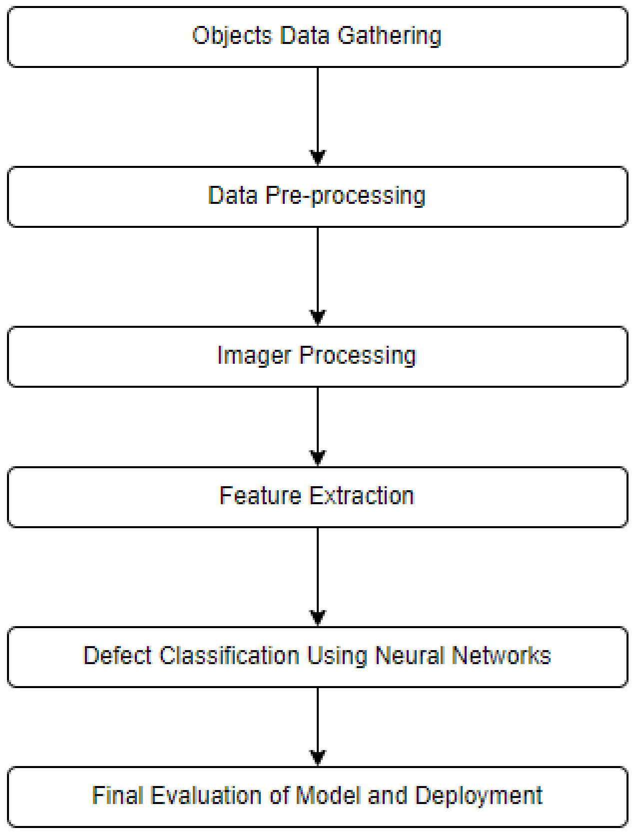

Figure 4.1 Process Model of Project

CHAPTER: 5 PROJECT PLAN

CHAPTER 5 PROJECT PLAN

5.1 List of Major Activities

5.1.1 Tasks for Building Prototype Model in First Phase

Task 1: - Data Gathering
Task 2: - Data Exploration and Pre-processing
Task 3: - Learning About Neural Networks
Task 4: - Implementation of Neural Network

5.1.2 Time Duration to Complete First Phase

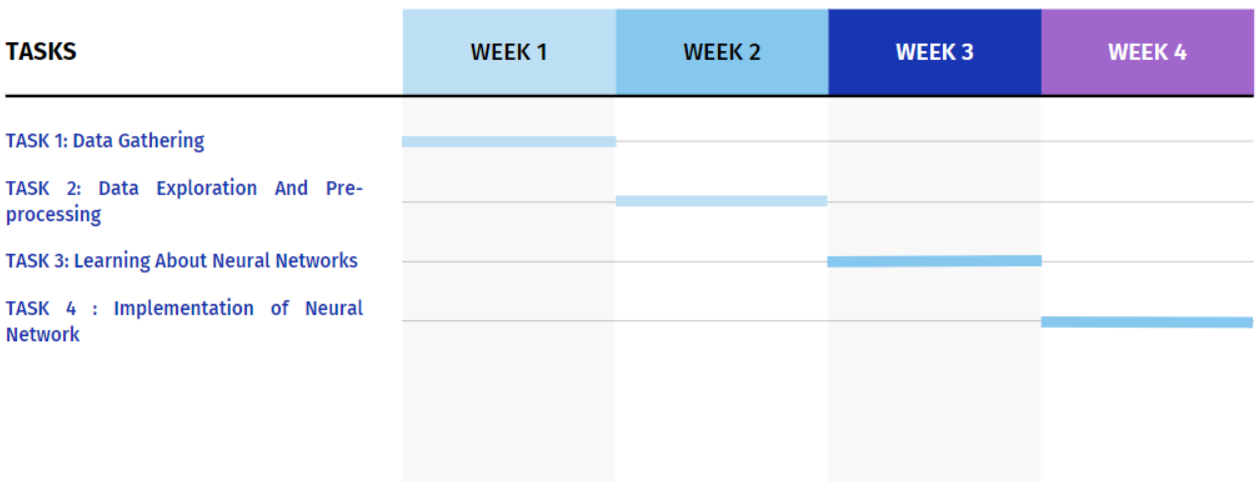

Figure 5.1 Task Completion Time Duration in First Phase

5.1.3 Tasks for Implementing Defect Detection in Second Phase

Task 1: - Data Creation
Task 2: - Data Pre-processing
Task 3: - Integrating the prototype Neural Network with new data
Task 4: - Testing it with video feed

5.1.4 Time Duration to Complete First Phase

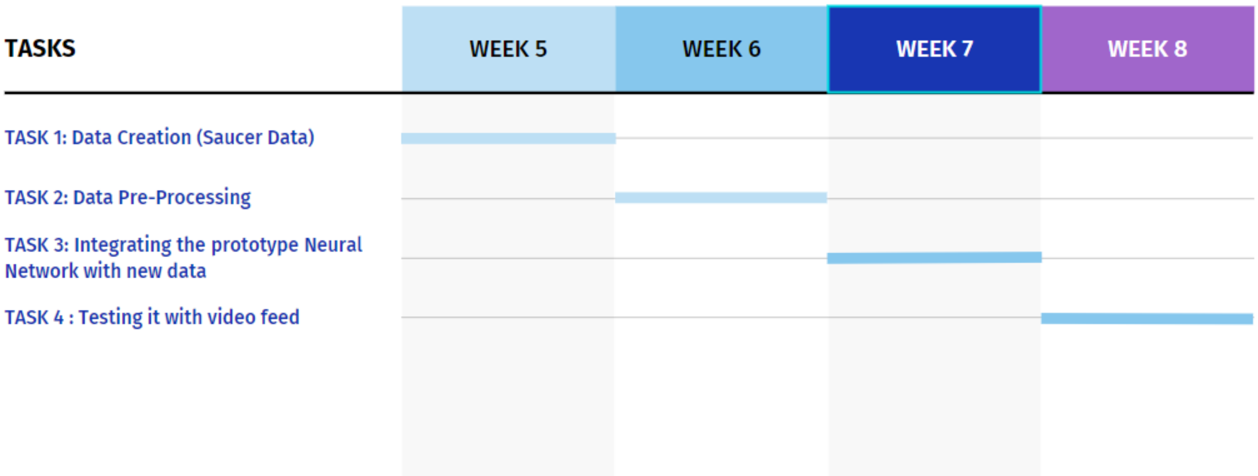

Figure 5.2 Task Completion Time Duration in Second Phase

5.1.5 Tasks for Implementing Front-End in Third Phase

Task 1: - MongoDB Database Exploration
Task 2: - Front-End Development in Flask Framework
Task 3: - Integrating MongoDB database with the framework
Task 4: - Adding miscellaneous functionalities in analytical part of model

5.1.6 Time Duration to Complete Third Phase

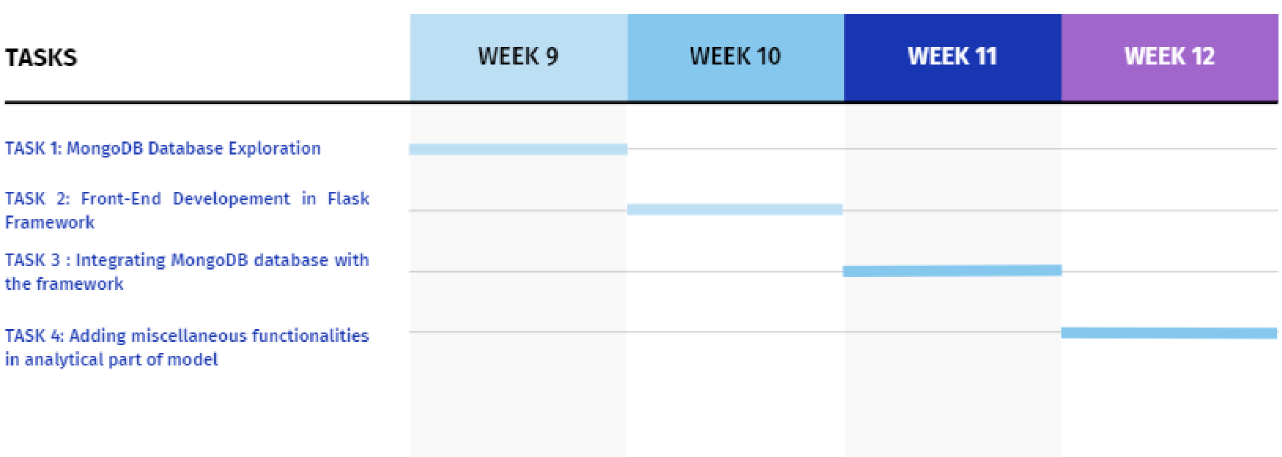

Figure 5.3 Tasks Completion Duration in Third Phase

CHAPTER: 6 IMPLEMENTATION DETAILS

CHAPTER 6 IMPLEMENTATION DETAILS

6.1 Data Gathering

6.1.1 Data Gathering for Prototype Model

In order to gather data for the proposed project multiple websites were explored to find data related to images of objects like pistons, hot strips, fabrics, metal plates etc. For the purpose of data gathering the images related to the above-mentioned examples have been utilized to build a neural network and to explore how to identify defects and furthermore classify them whether they are normal, having defect type 1 or having defect type 2.

6.1.2 Data Creation for Implementing Defect Detection Model

For implementing defect detection and also to classify what type of defects are present in an object while examining it or passing it through a live feed/camera a saucer is considered to for building data to check whether the model which will be implemented in the first phase of prototype is feasible or not. The Data comprises of images of saucer which are taken after making a video of saucer and converting that video into number of frames of images in order to train it through convolution neural network.

6.2 Data Understanding

6.2.1 Data for prototype model building (Pistons Dataset)

In order to build a neural network for the proposed project we have utilized the data consisting of piston images for classification.

The data contains images of piston which are classified into 3 categories:

1. Normal pistons
2. Pistons having Defect Type 1
3. Pistons having Defect Type 2

The images available in the dataset are of dimensions 80x80 pixels each.

In Pistons Dataset there are total 285 images. They have been split into 3 categories training, testing and validation dataset. Training dataset contains 240 images, validation dataset contains 30 images and Testing dataset contains 15 images.

Figure 6.1 Pistons Dataset

6.2.2 Data for building defect detection model (Saucer Dataset)

For building saucer dataset, a video has been taken of three different saucers and then the video has been converted into image frames so that neural network can train and test the model easily based on image processing.

The saucer dataset has been classified as follows into 3 different categories:
1. defect1 (having minor defects like design, dents etc.)
2. defect2 (broken or chipped saucer)
3. Normal (perfect saucer)

The images available in the dataset are of dimensions 200x200 pixels each.

In Saucer Dataset there are total 3086 images. They have been split into 3 categories training, testing and validation dataset. Training dataset contains 2981 images, validation dataset contains 79 images and Testing dataset contains 26 images.

Figure 6.2 Saucer Dataset

6.3 Building Neural Network and Implementing Defect Detection

6.3.1 Model Building for Pistons Dataset

For the proposed project Sequential Neural Network has been built which consists of following layers: -

1st Layer: - Convo2D Layer and MaxPool 2D Layer
2nd Layer: - Convo2D Layer and MaxPool 2D Layer
3rd Layer: - Convo2D Layer and MaxPool 2D Layer
4th Layer: - Dense Layer
Output Layer: - Dense Layer consisting of 3 inputs

To optimize the output Adam optimizer has been utilized and the Accuracy metrics has been employed to obtain proper accuracy of the Sequential Neural Network.

Up to 10 Epochs were executed to obtain the accuracy

```
Epoch 6/10
10/10 [==============================] - 2s 150ms/step - loss: 0.2324 - accuracy: 0.9333 - val_loss: 0.3488 -
val_accuracy: 0.8000
Epoch 7/10
10/10 [==============================] - 2s 164ms/step - loss: 0.1597 - accuracy: 0.9500 - val_loss: 0.2732 -
val_accuracy: 0.8333
Epoch 8/10
10/10 [==============================] - 1s 138ms/step - loss: 0.0875 - accuracy: 0.9875 - val_loss: 0.1140 -
val_accuracy: 0.9333
Epoch 9/10
10/10 [==============================] - 1s 140ms/step - loss: 0.0492 - accuracy: 0.9958 - val_loss: 0.0665 -
val_accuracy: 1.0000
Epoch 10/10
10/10 [==============================] - 1s 124ms/step - loss: 0.0306 - accuracy: 0.9958 - val_loss: 0.0419 -
val_accuracy: 1.0000
```

Figure 6.3 Accuracy of neural network model of Pistons dataset

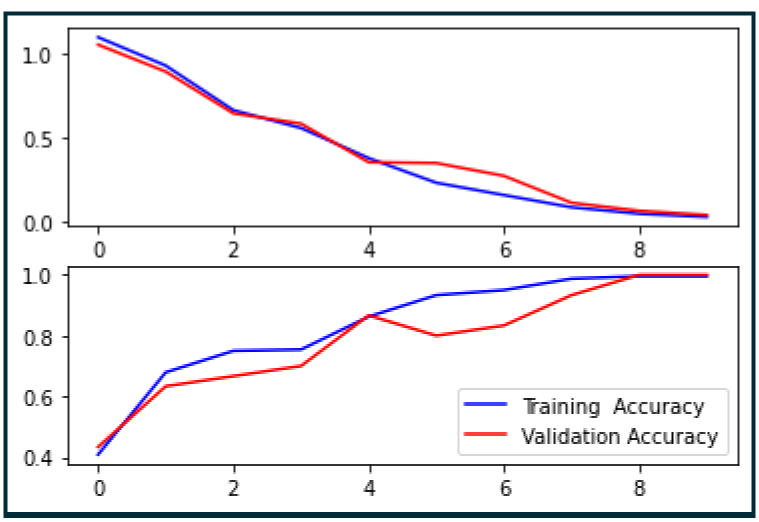

Figure 6.4 Accuracy Visualization (Pistons Datset)

Output of Defected Images

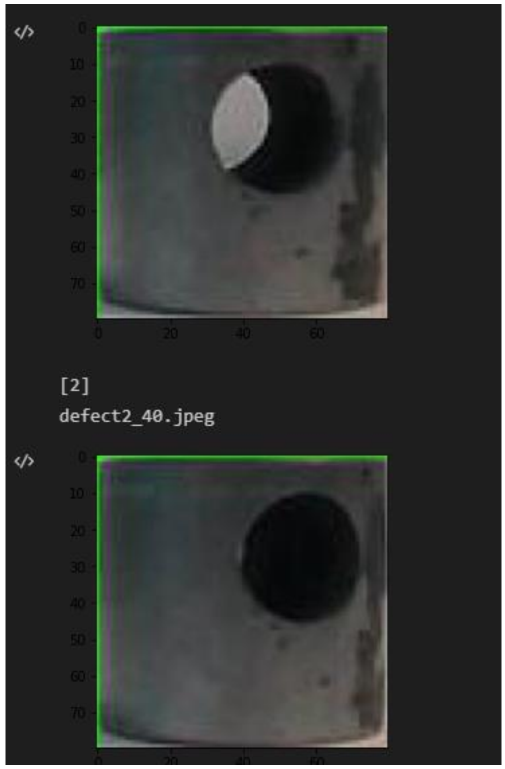

Figure 6.5 Defective Piston Output

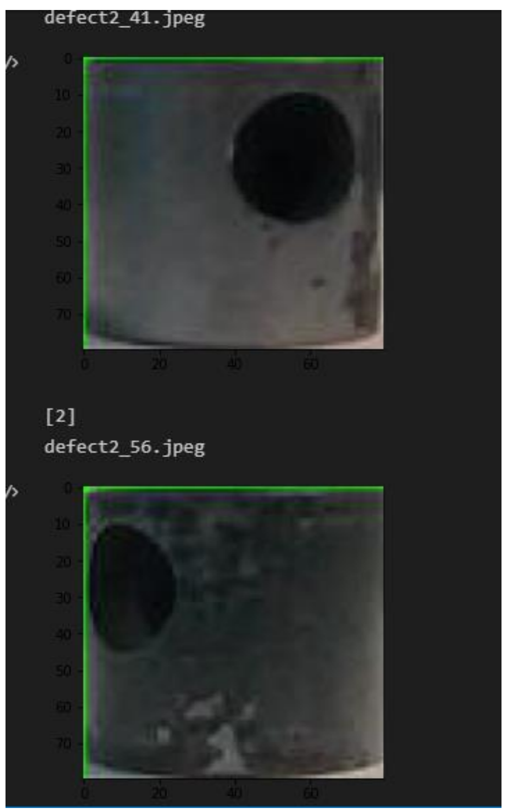

Figure 6.6 Defective Piston Output

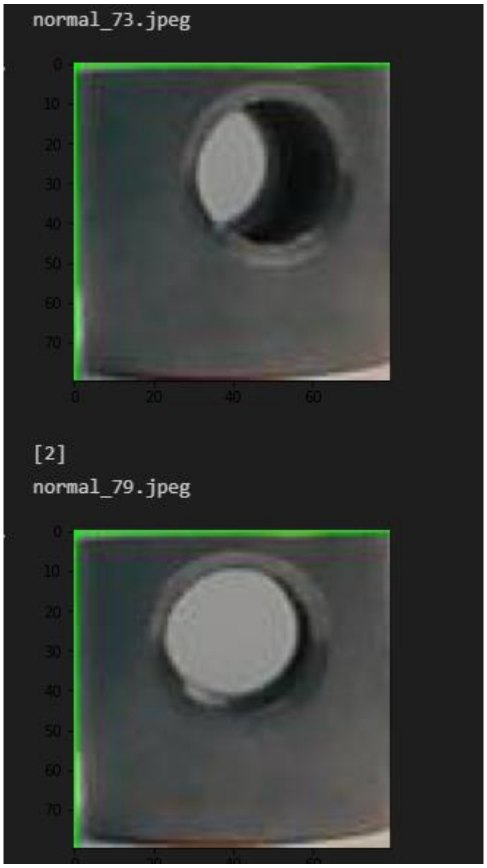

Figure 6.7 Defective Piston Output

6.3.2 Data Preparation and Pre-processing on Saucer Data

As mentioned in Data Understanding section three different types of saucers are considered for data creation and checking and classifying defects present within them. Initially according to the mentioned 3 categories of the saucer video has been taken of each saucer as follows: -

Figure 6.8 Normal Saucer Video

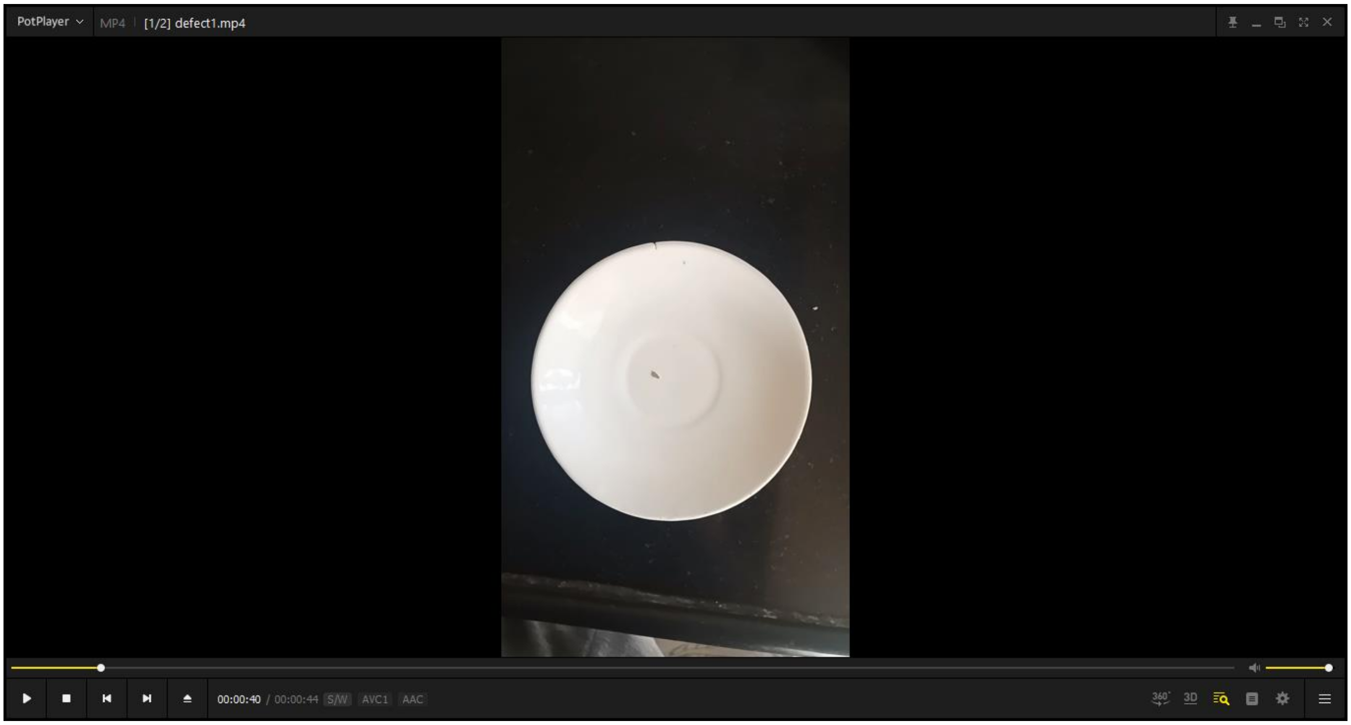

Figure 6.9 Defect1 Saucer Video

Figure 6.10 Defect2 Saucer Video

After taking videos of the above-mentioned 3 different types of saucers all the videos are converted into dataset of images frame by frame. All the images obtained from the above videos are stored in a training dataset initially. After that some images are segregated into other two categories of testing and validation in order to build the neural network and train them accordingly. The frames obtained in each category leads up to 1000 approximately.

6.3.3 Model Building for Saucer Dataset

As mentioned in the model building section of pistons dataset same parameters have been utilized for model building in saucer dataset.

To optimize the output Adam optimizer has been utilized and the Accuracy metrics has been employed to obtain proper accuracy of the Sequential Neural Network. Up to 10 Epochs were executed to obtain the accuracy.

```
curacy: 1.0000
Epoch 6/10
125/125 [==============================] - 57s 454ms/step - loss: 5.0739e-06 - accuracy: 1.0000 - val_loss: 9.5367e-07 - val_ac
curacy: 1.0000
Epoch 7/10
125/125 [==============================] - 56s 451ms/step - loss: 4.0501e-06 - accuracy: 1.0000 - val_loss: 7.7561e-07 - val_ac
curacy: 1.0000
Epoch 8/10
125/125 [==============================] - 56s 448ms/step - loss: 3.2021e-06 - accuracy: 1.0000 - val_loss: 6.4433e-07 - val_ac
curacy: 1.0000
Epoch 9/10
125/125 [==============================] - 56s 450ms/step - loss: 2.6458e-06 - accuracy: 1.0000 - val_loss: 5.2663e-07 - val_ac
curacy: 1.0000
Epoch 10/10
125/125 [==============================] - 56s 451ms/step - loss: 2.3707e-06 - accuracy: 1.0000 - val_loss: 4.4213e-07 - val_ac
curacy: 1.0000
```

Figure 6.11 Accuracy of neural network model of Saucer dataset

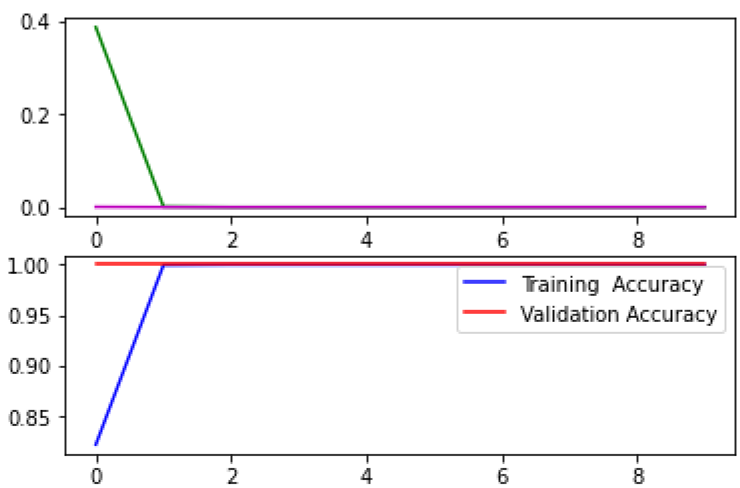

Figure 6.12 Accuracy Visualization (Saucer Dataset)

Now testing it on random images and comparing the model recognizes the types of defects correctly when an image is provided to it

Figure 6.13 Defect1 detection

Figure 6.14 Defect2 detection

Figure 6.15 Normal Saucer Detection

After testing it on random images the neural network model is saved in the form of a json file in order to run it anytime and avoid waiting for epoch to complete processing.

6.4 Testing on Video Feed

In order to check whether the implemented model for defect detection and classification is working properly or not when a video or image of the saucers is displayed a test video is created to test the model.

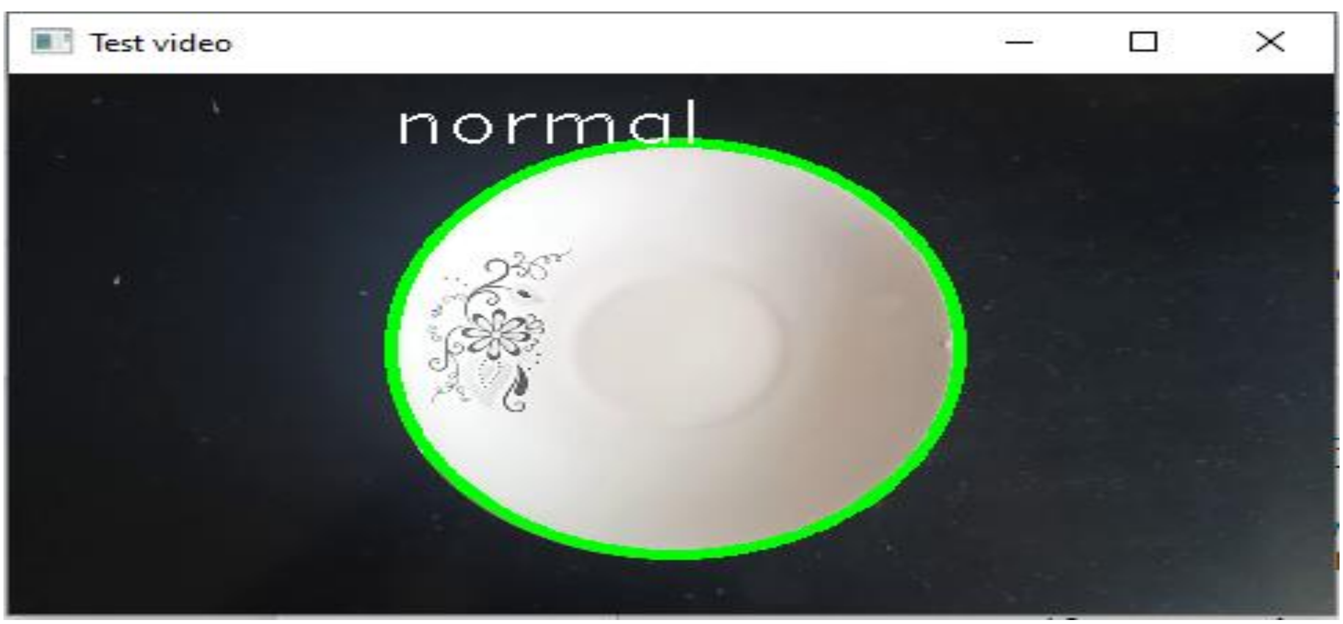

Figure 6.16 Normal Saucer detection in video

Figure 6.17 defect_1 Saucer detection in video

Figure 6.18 defect_@ Saucer detection in video

The built Convolution Neural Network model detects the defects and classifies them in a proper manner when used to detect objects in a video.

6.5 Front End Development

Front-end development recognizes that area of development where the users come into the picture. It represents how a user sees the model and recognizes it front of his/her own eyes. The model should be presented to the user in the most aesthetic way so that it is easy to understand and interpret.

For front-end development of above built model HTML, CSS and JavaScript has been utilized which is then integrated with Flask Framework to provide a slick view to the user.Defect detection of saucer or any other objects is also possible through webcam, the above built model has been designed to detect defects through webcam as shown below.

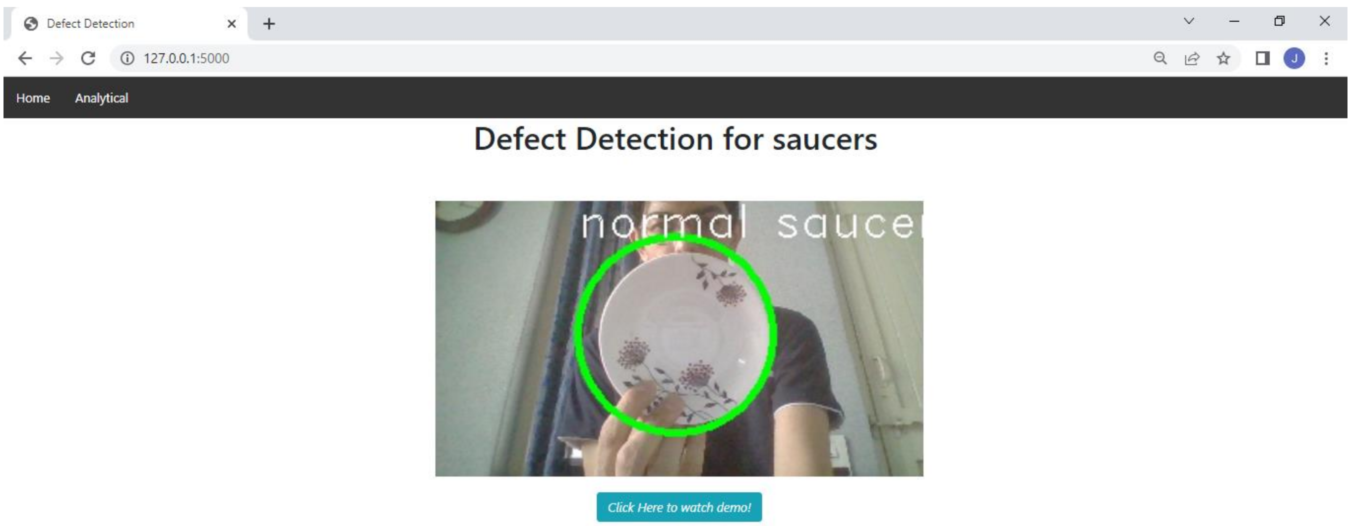

Figure 6.19 Defect Detection through webcam

Also, another functionality that has been assigned to the web app is that one can see a demo video or any video which is provided by the user and based on that defect detection can also take place by clicking on "Click Here to watch demo!".

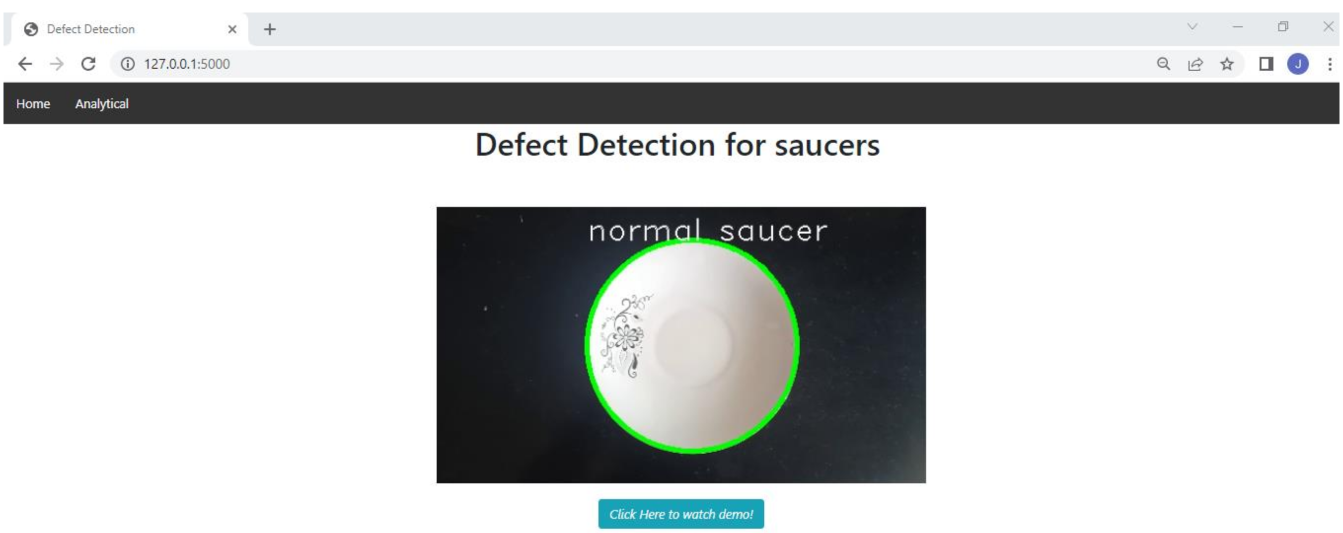

Figure 6.20 Defect Detection through a video

6.6 Analytics and Insights

In order to provide insights and information about the detected defects an ID of the object is extracted each time it is scanned along with the displayed category of normal saucer, plain saucer or chipped saucer and then it is stored in the document database using MongoDB. MongoDB is later integrated with flask framework to provide insights of the data obtained.

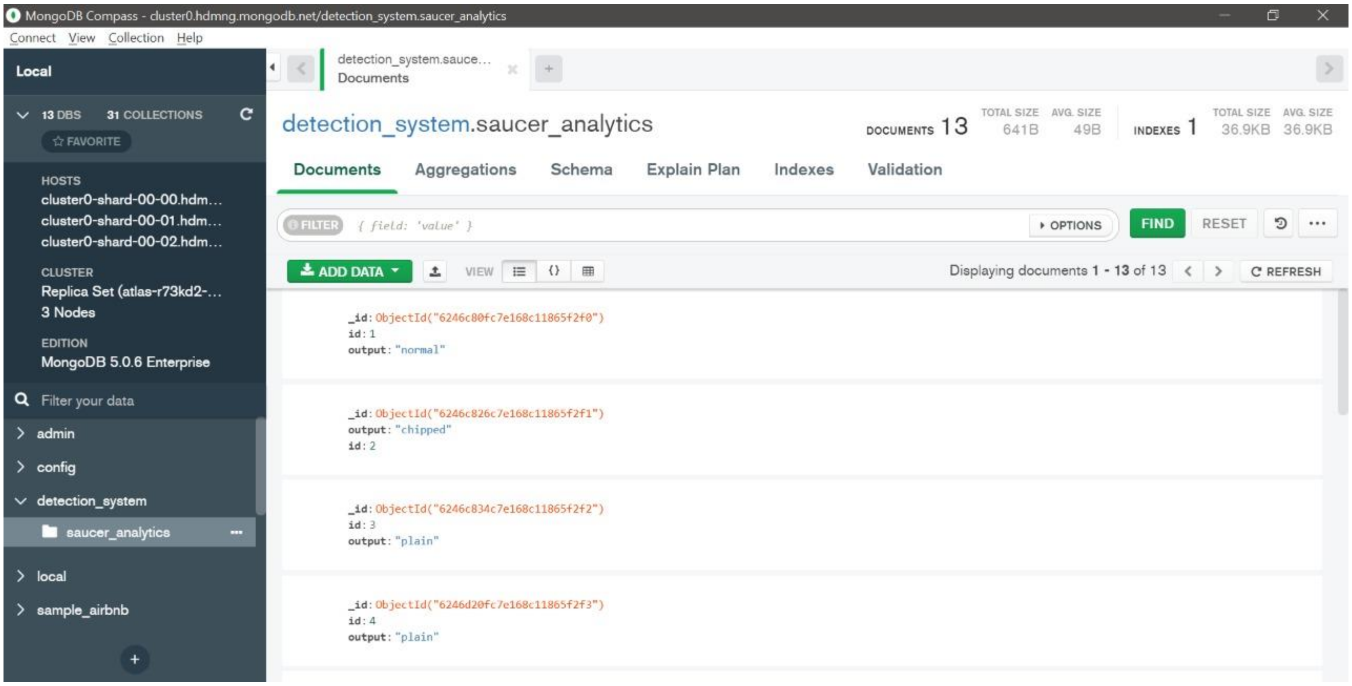

Figure 6.21 Document Database on MongoDB

After integrating the database with the Flask Framework, the following insights are displayed to the user every time a demo video is played or an object is detected using webcam, it categorizes the data and provides valuable information for the same.

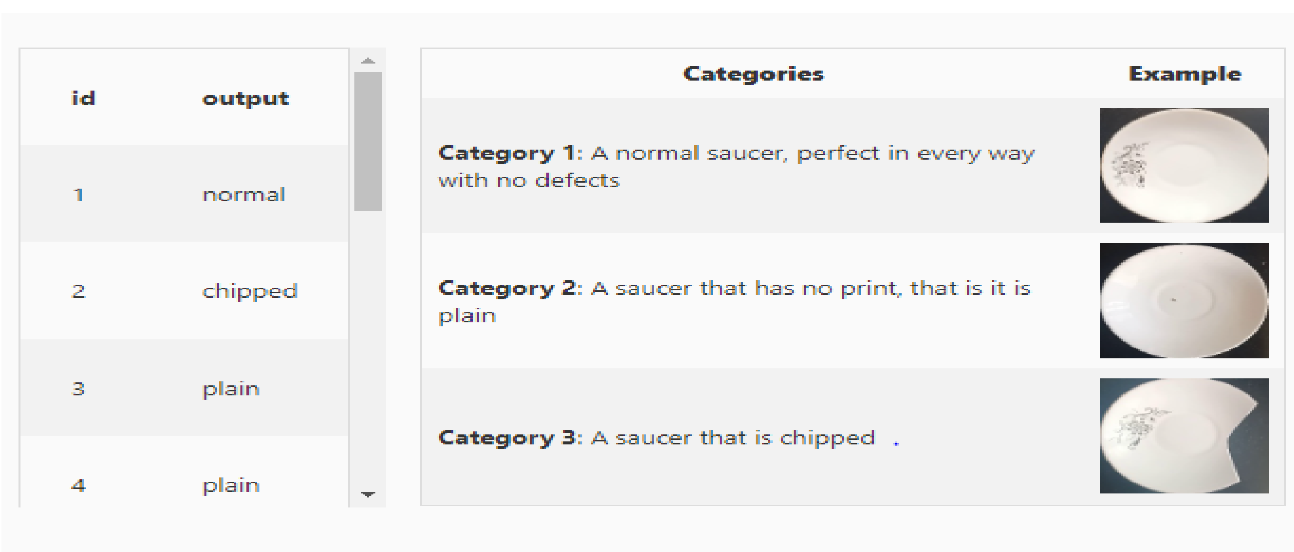

Figure 6.22 Classification of saucers

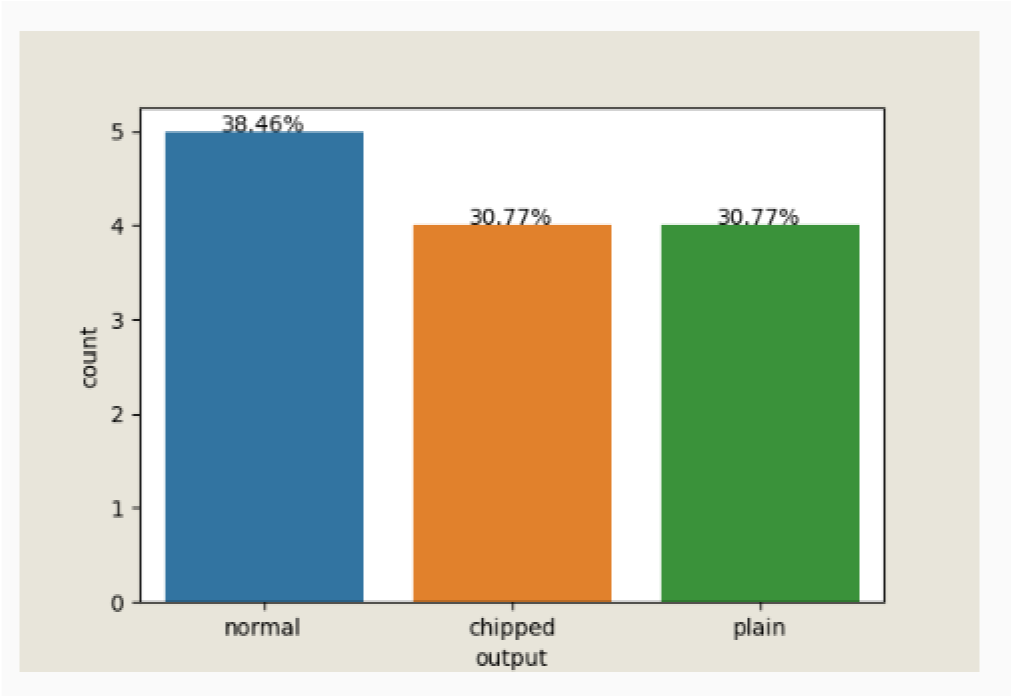

Figure 6.23 Bar Graph of number of saucers detected over time

The final analytical window displayed to the user will be as shown below

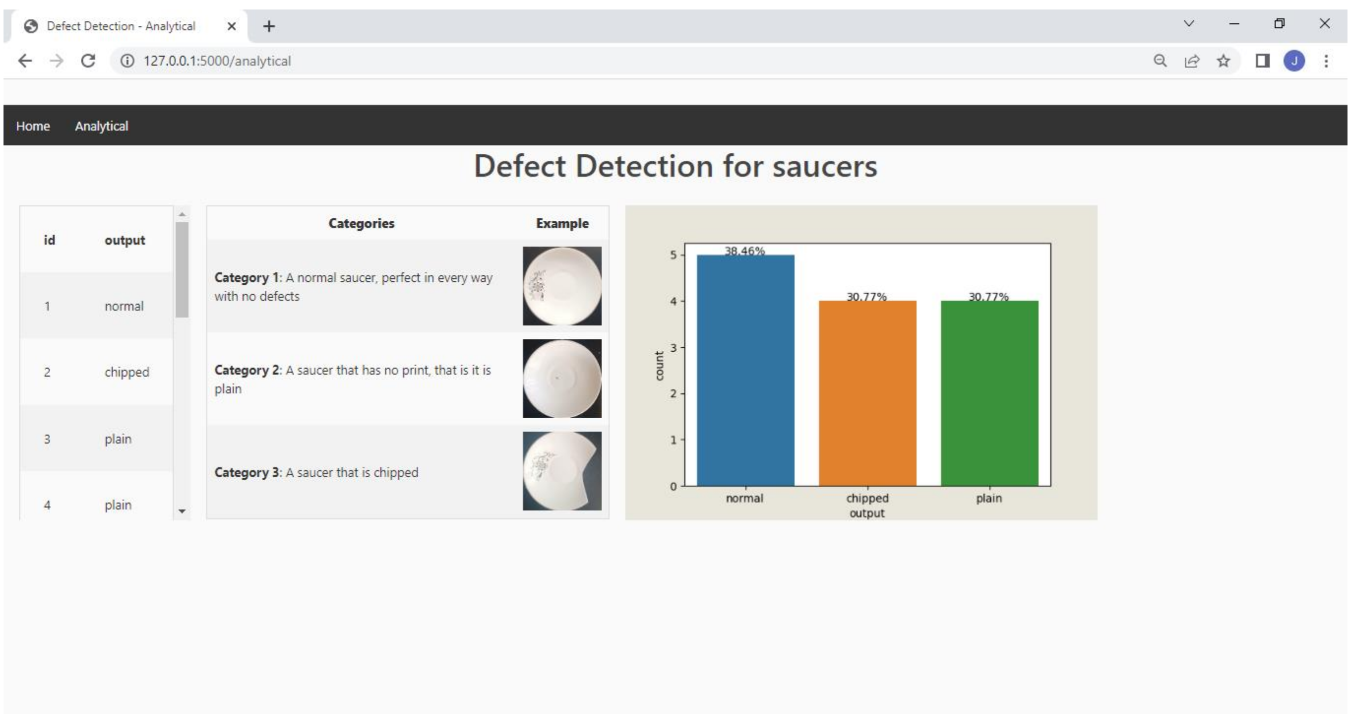

Figure 6.24 Final Analytical Output

CHAPTER: 7 CONCLUSION AND FUTURE WORK

CHAPTER 7 CONCLUSION AND FUTURE WORK

Conclusion

In conclusion a Sequential Neural Network is successfully built for pistons dataset and saucer dataset and it is achieving accuracy nearly to 100% for training, validation and testing data. Furthermore, saucer dataset is also created and tested perfectly for the built model. Defect Detection in saucers is identified correctly and giving promising results. General guidelines for developing machine learning model have been identified based on defect classification paradigm. The above developed model after being deployed on an online web application shows meaningful information about the defects detected and still displays a lot of potential to evolve and provide further insights based on data provided.

Future work

In this project a database has been developed from the scratch and a neural network has been trained on the aforementioned database, then the model is saved so that anyone can use it easily without wasting any time to retrain the model every time we run the program. After that, a front-end application has been developed using flask through workers can look at the output and pick out the defective pieces from the production line. For future work, we can extend the program to automatically select and remove the defective pieces from the production line, integrating the purely software with IOT based production line.

CHAPTER: 8 REFERENCES

CHAPTER 8 REFERENCES

[1] https://www.ndt.net/article/wcndt2004/pdf/in-process_ndt-nde/788_popat.pdf

[2] https://www.analyticsvidhya.com/blog/2021/06/image-classification-using-convolutional-neural-network-with-python/

[3] https://mobidev.biz/blog/ai-visual-inspection-deep-learning-computer-vision-defect-detection

[4] Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges By Jing Yang, Shaobo Li, Zheng Wang ,Hao Dong ,Jun Wang and Shihao Tang

[5] https://matplotlib.org/stable/index.html

[6] https://flask.palletsprojects.com/en/2.1.x/

[7] Defect Detection and Classification Using Machine Learning Classifier By Mitesh Popat and S.V. Barai, Indian Institute of Technology, Kharagpur

[8] https://www.analyticsvidhya.com/blog/2021/06/image-classification-using-convolutional-neural-network-with-python/

[9] https://www.tensorflow.org/

[10] https://www.kaggle.com/satishpaladi11/mechanic-component-images-normal-defected