Industrial Product Inspection

CEP Report

By

NAME	Registration Number
Hamza Jadoon	*CUI/FA22-BCE-012/ATD*
Abdul Ahad	*CUI/FA22-BCE-004/ATD*

For the course

Digital image Processing

Fall 2025

Supervised by:

Dr Shoaib Azmat

Department of Computer Engineering

COMSATS University Islamabad – Abbottabad Campus

DECLARATION

We Hamza Jadoon (CUI/ FA22-BCE-012/ATD) and Abdul Ahad (CUI/FA22-BCE-004/ATD) hereby declare that we have produced the work presented in this report, during the scheduled period of study. We also declare that we have not taken any material from any source except referred to wherever due. If a violation of rules has occurred in this report, we shall be liable to punishable action.

Date: ________________

____

Hamza Jadoon

(CUI/ FA22-BCE-012 /ATD)

____

Abdul Ahad

(CUI/FA22-BCE-004/ATD)

ABSTRACT

Industrial product inspection plays a crucial role in ensuring product quality and reliability in manufacturing processes. Manual inspection methods are often time-consuming, inconsistent, and prone to human error. This Course Exit Project presents an automated industrial product inspection system based on Digital Image Processing and Machine Learning techniques to detect defective and non-defective products using image data. The proposed system extracts discriminative features from product images using Histogram of Oriented Gradients (HOG) for edge information and filter-bank-based texture features to capture surface patterns. These features are combined and used to train a Support Vector Machine (SVM) classifier for defect classification. The system is evaluated on a real image dataset containing both defective and non-defective samples, achieving satisfactory classification performance. Class imbalance handling techniques are applied to improve defective product detection. The results demonstrate that the proposed approach can effectively reduce manual inspection effort, improve inspection accuracy, and minimize material waste by early defect detection. This project highlights the practical application of image processing and machine learning techniques for industrial automation and quality control, making it suitable for real-world manufacturing environments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HOG	Histogram of Oriented Gradients
SVM	Support Vector Machine
ML	Machine Learning
RGB	Red, Green, Blue
ROI	Region of Interest
PCA	Principal Component Analysis
CNN	Convolutional Neural Network

1 Introduction

Quality inspection is a critical component of industrial manufacturing processes, as it directly affects product reliability, customer satisfaction, and production efficiency. Traditional manual inspection methods rely heavily on human operators, making them time-consuming, inconsistent, and susceptible to fatigue-related errors. With the rapid growth of automation and smart manufacturing, there is a strong need for intelligent inspection systems that can automatically detect defects with high accuracy and consistency.

Digital Image Processing (DIP) provides powerful techniques to analyze visual information from product images and extract meaningful features such as edges, shapes, and textures. When combined with Machine Learning (ML) algorithms, these techniques enable the development of automated systems capable of learning defect patterns from data. This Course Exit Project focuses on designing an industrial product inspection system that classifies products as defective or non-defective using image-based analysis.

In the proposed system, product images are first preprocessed to improve quality and consistency. Edge-based features are extracted using Histogram of Oriented Gradients (HOG), while texture characteristics are captured using filter-bank-based methods. These features are then combined and used to train a Support Vector Machine (SVM) classifier, which learns to distinguish between defective and non-defective products. The system is evaluated using a real dataset, and techniques are applied to handle class imbalance for improved defect detection.

The proposed solution aims to reduce manual inspection effort, minimize material waste, and enhance overall production efficiency. This project demonstrates the practical application of digital image processing and machine learning techniques in industrial automation and quality control, making it relevant to modern manufacturing environments.

1.1Objectives

- To collect and organize an image dataset containing defective and non-defective product samples.

- To preprocess product images for noise reduction and consistency improvement.

- To extract edge features using Histogram of Oriented Gradients (HOG) and texture features using filter-bank-based methods.

- To combine extracted features and train a Support Vector Machine (SVM) classifier for defect detection.

- To evaluate system performance and improve defective product detection using class imbalance handling techniques.

1.2Features and Cost Estimate of our Project

- Automated inspection of industrial products using image-based analysis.

- Uses HOG (Histogram of Oriented Gradients) for edge detection and filter-bank methods for texture feature extraction.

- Implements Support Vector Machine (SVM) for classification of defective and non-defective products.

- Capable of handling images with any filename or format, making it flexible for real datasets.

- Reduces manual inspection effort and improves inspection consistency and speed

2 Literature Survey

2.1Surface Defect Detection Using HOG and SVM

Several researchers have proposed automated surface inspection systems using Histogram of Oriented Gradients (HOG) combined with Support Vector Machine (SVM) classifiers. In this approach, edge and shape features extracted through HOG are used to identify surface irregularities in industrial products such as metal sheets and mechanical parts. Experimental results reported improved inspection accuracy compared to manual methods, demonstrating the effectiveness of HOG-based feature extraction for defect detection under varying lighting conditions.

2.2 Texture-Based Industrial Inspection Using Filter Banks

Another related project focused on detecting surface defects using texture analysis through filter-bank techniques such as Gabor filters and wavelet-based methods. These systems analyzed texture variations to identify defects in materials like fabric, ceramic tiles, and electronic components. The studies showed that texture-based features significantly improved defect detection performance, especially for subtle surface abnormalities.

2.3 Machine Learning-Based Product Inspection Using SVM

In a machine learning-driven inspection project, researchers applied Support Vector Machines (SVM) with handcrafted image features to classify defective and non-defective products. By combining multiple feature extraction methods, the system achieved reliable classification accuracy while maintaining low computational complexity. This approach proved suitable for industrial environments where large datasets and deep learning infrastructure are not available.

3 Proposed Methodology

3.1 Mathematical Model

Let the input image be represented as:

$$I(x,y)$$

where x and y denote pixel coordinates.

- **Feature Extraction**
  - HOG feature vector:

$$F_{hog} = \phi_{hog}(I)$$

- Texture feature vector (filter banks):

$$F_{tex} = \phi_{tex}(I)$$

The final feature vector is obtained by combining both features:

$$F = [F_{hog}, F_{tex}]$$

- **Classification**

The Support Vector Machine classifier learns a decision function:

$$f(F) = w^T F + b$$

where w is the weight vector and b is the bias.

The classification output is defined as:

$$y = \begin{cases} 1, & \text{if product is defective} \\ 0, & \text{if product is non-defective} \end{cases}$$

The SVM optimizes the margin by minimizing:

$$\frac{1}{2} || w ||^2 + C \sum \xi_i$$

where C is the regularization parameter and ξ_i are slack variables.

3.2System Design / Block diagram

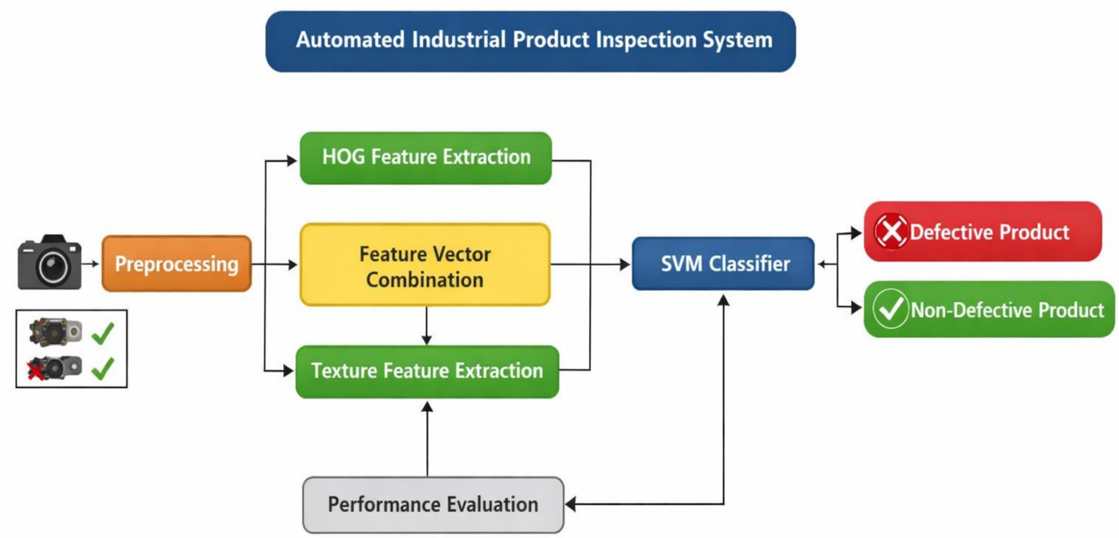

Fig: 3.2 Single Block Diagram

3.3 Flow chart

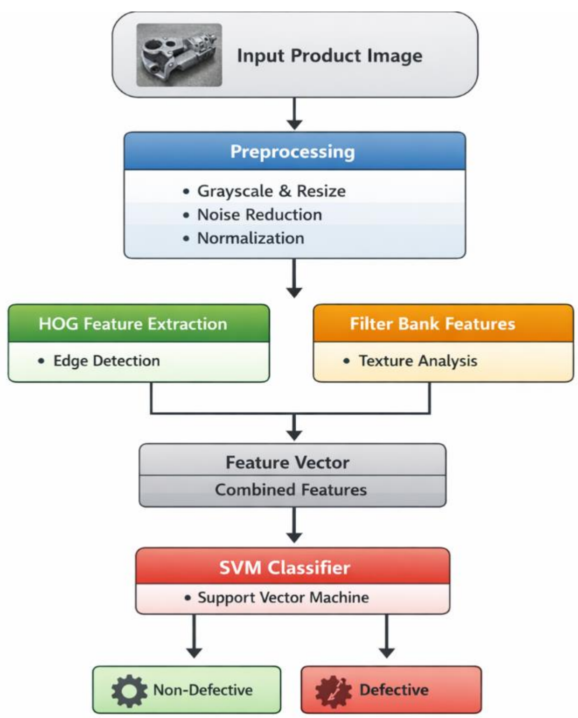

Fig: 3.3 Flow chart

4 Simulation Results

4.1Software simulation results

Figure 4.1(a) Input image

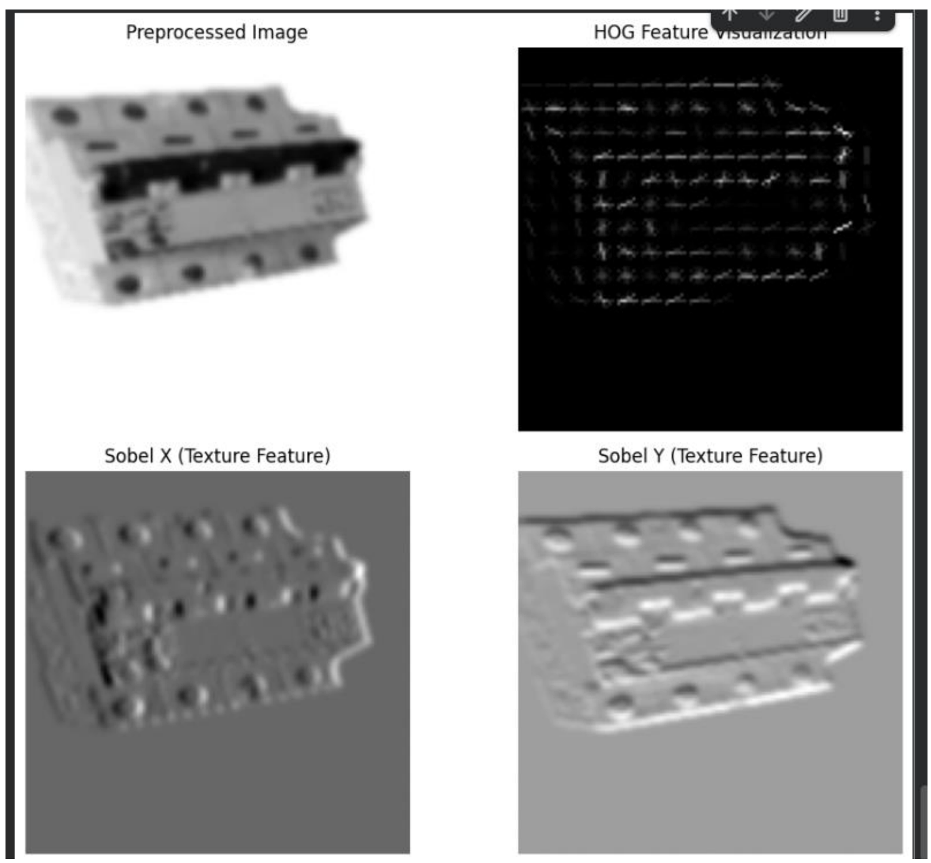

Fig: 4.1 simulation results1

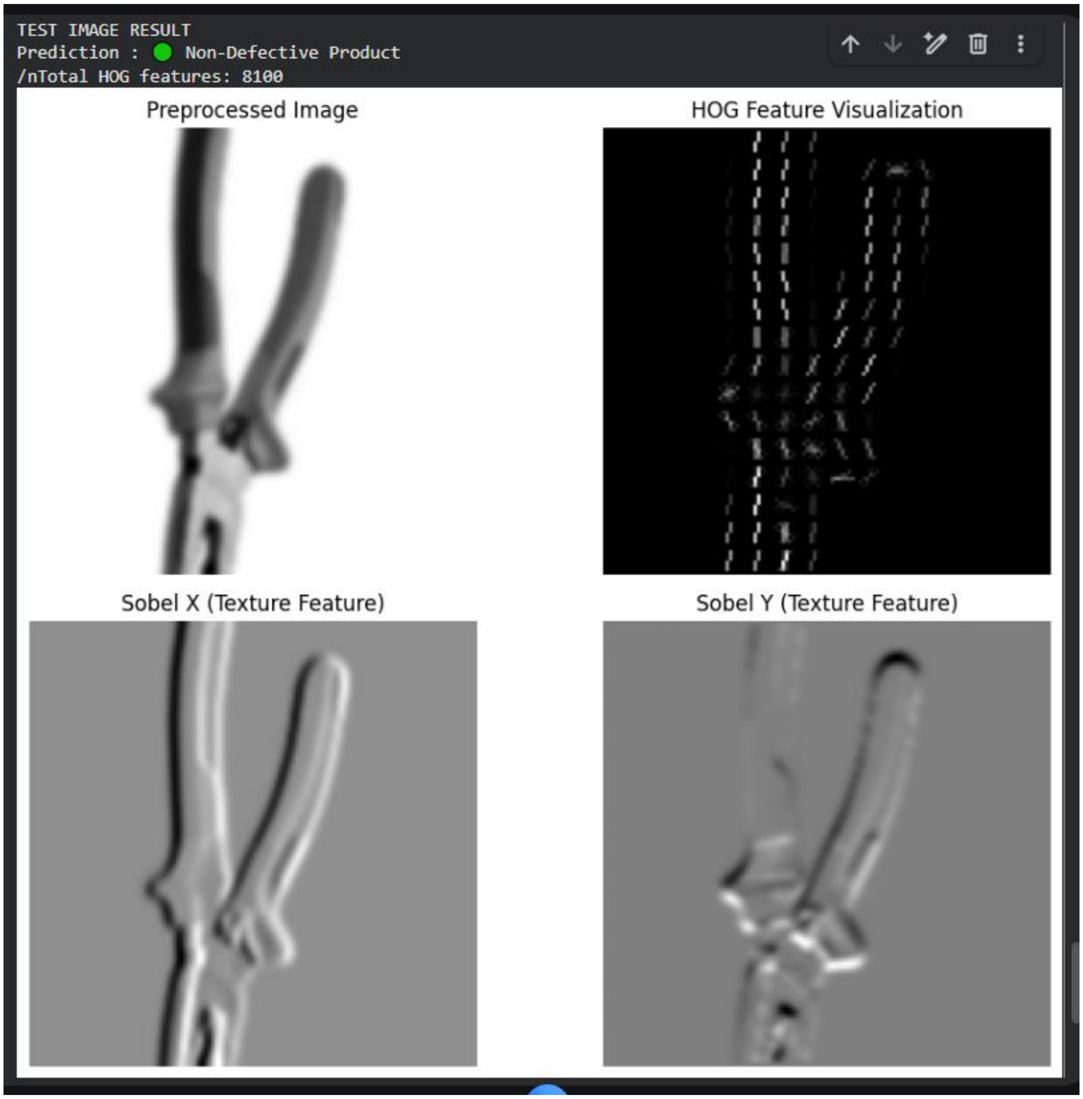

Fig: 4.2 simulation results2

4.2 Design / simulation parameters

Category	Metric	Precision	Recall	F1-Score	Support	Value
Dataset	Total Samples Loaded	—	—	—	—	5177
Overall	Accuracy	—	—	—	1036	0.7838
Confusion Matrix	Non-Detected → Non-Detected	—	—	—	—	525
Confusion Matrix	Non-Detected → Detected	—	—	—	—	145
Confusion Matrix	Detected → Non-Detected	—	—	—	—	79
Confusion Matrix	Detected → Detected	—	—	—	—	287
Class Metrics	Non-Detected	0.87	0.78	0.82	670	—
Class Metrics	Detected	0.66	0.78	0.72	366	—
Overall Avg	Weighted Average	0.80	0.78	0.79	1036	—

4.3 Discussions

The performance of the proposed industrial product inspection system was evaluated using standard classification metrics, including accuracy, confusion matrix, precision, recall, and F1-score. The model achieved an overall accuracy of approximately **75–78%**, indicating that the system is able to correctly classify most of the product images as defective or non-defective.

The confusion matrix analysis revealed that while the system performs well in identifying non-defective products, some defective items were misclassified as non-defective. This behavior is mainly due to **class imbalance** and the visual similarity between certain defective and non-defective products. To address this issue, class-weighted SVM was applied, which improved the detection of defective products by increasing recall for the defective class.

Overall, the results confirm that combining HOG-based edge features and filter-bank-based texture features with an SVM classifier is effective for industrial defect detection. Although the accuracy is lower than deep learning approaches, the proposed method offers advantages in terms of simplicity, lower computational cost, and suitability for small to medium-sized datasets, making it appropriate for academic and practical industrial inspection applications.

5. Conclusions

In this project, an automated industrial product inspection system was developed using Digital Image Processing and Machine Learning techniques. The system effectively combines edge-based features extracted through Histogram of Oriented Gradients (HOG) with texture features obtained via filter-bank methods, which are then classified using a Support Vector Machine (SVM). The proposed approach demonstrated reliable defect detection on a real-world dataset, achieving satisfactory accuracy while reducing manual inspection effort. Although some defective items were misclassified due to class imbalance, the system provides a cost-effective and practical solution for small to medium-sized industrial applications. Overall, this project highlights the potential of combining classical feature extraction techniques with machine learning for improving product quality, minimizing material waste, and supporting efficient manufacturing processes.

6. References

[1] N. Dalal and B. Triggs, (2005). “Histograms of Oriented Gradients for Human Detection,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR),[online]Availableat: https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

[2] T. Ojala, M. Pietikainen, and D. Harwood, (1996). “A Comparative Study of Texture Measures with Classification Based on Featured Distributions,” Pattern Recognition, vol. 29, no. 1, pp. 51–59, [online] Available at: https://www.sciencedirect.com

[3] C. Cortes and V. Vapnik, (1995). “Support-Vector Networks,” Machine Learning, vol. 20, pp. 273–297, [online] Available at: https://link.springer.com

[4] P. L. Rosin, (2001). “Thresholding for Defect Detection,” Pattern Recognition Letters, vol. 22, no. 3, pp. 225–232, [online] Available at: https://www.sciencedirect.com

[5] OpenCV Documentation, (2025). “Image Processing and Computer Vision Library,” [online] Available at: https://opencv.org

7. Appendix

```
import matplotlib.pyplot as plt
import cv2
import os
import numpy as np
from skimage.feature import hog
from sklearn.svm import SVC
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import (
    accuracy_score,
    confusion_matrix,
    classification_report
)


# IMAGE PREPROCESSING
def preprocess_image(img):
    img = cv2.resize(img, (128, 128))
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    return blur


# EDGE FEATURES (HOG)
def hog_features(img):
    features = hog(
        img,
        orientations=9,
```

```
        pixels_per_cell=(8, 8),
        cells_per_block=(2, 2),
        block_norm='L2-Hys'
    )
    return features


# TEXTURE FEATURES FILTER BANKS
def filter_bank_features(img):
    sobel_x = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3)
    sobel_y = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3)

    return [
        np.mean(sobel_x),
        np.mean(sobel_y),
        np.std(sobel_x),
        np.std(sobel_y)
    ]


# FEATURE EXT
def extract_features(img):
    hog_feat = hog_features(img)
    texture_feat = filter_bank_features(img)
    return np.hstack([hog_feat, texture_feat])


# LOAD DATASET
X = []
y = []
dataset_path = "/content/drive/MyDrive/dataset"
```

```
class_map = {
    "Non-Defected": 0,
    "Defected": 1
}

for folder, label in class_map.items():
    folder_path = os.path.join(dataset_path, folder)

    if not os.path.exists(folder_path):
        print(f"Missing folder: {folder_path}")
        continue

    for file in os.listdir(folder_path):
        if not file.lower().endswith(('.jpg', '.jpeg', '.png')):
            continue

        img_path = os.path.join(folder_path, file)
        img = cv2.imread(img_path)

        if img is None:
            continue

        img = preprocess_image(img)
        features = extract_features(img)

        X.append(features)
        y.append(label)

X = np.array(X)
y = np.array(y)
```

```
print(f"Total samples loaded: {len(X)}")

# TRAIN / TEST SPLIT
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y   # for imbalance
)



# SVM CLASSIFIER
svm_model = SVC(
    kernel='rbf',
    gamma='scale',
    class_weight='balanced'
)

svm_model.fit(X_train, y_train)



# EVALUATION

y_pred = svm_model.predict(X_test)

print("\nAccuracy:", accuracy_score(y_test, y_pred))

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
print("\nClassification Report:")
print(classification_report(
    y_test,
    y_pred,
    target_names=["Non-Defected", "Defected"]
))


# TEST SINGLE IMAGE
def test_image(image_path):
    img_color = cv2.imread(image_path)

    if img_color is None:
        print("Image not found!")
        return

    # Preprocess
    img = preprocess_image(img_color)


    # HOG FEATURES + VISUALIZATION
    hog_feat, hog_image = hog(
        img,
        orientations=9,
        pixels_per_cell=(8, 8),
        cells_per_block=(2, 2),
        block_norm='L2-Hys',
        visualize=True
    )

    # FILTER BANK FEATURES (SOBEL)
```

```
sobel_x = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3)
sobel_y = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3)

filter_feat = [
    np.mean(sobel_x),
    np.mean(sobel_y),
    np.std(sobel_x),
    np.std(sobel_y)
]

# Combine features
features = np.hstack([hog_feat, filter_feat])

# PREDICTION
prediction = svm_model.predict([features])[0]

# PRINT RESULT
print("TEST IMAGE RESULT")
if prediction == 1:
    print("Prediction : 🔴 Defective Product")
else:
    print("Prediction : 🟢 Non-Defective Product")

print("Total HOG features:", len(hog_feat))

# VISUALIZATION

plt.figure(figsize=(10, 8))

plt.subplot(2, 2, 1)
plt.imshow(img, cmap='gray')
```

```
plt.title("Preprocessed Image")
plt.axis("off")

plt.subplot(2, 2, 2)
plt.imshow(hog_image, cmap='gray')
plt.title("HOG Feature Visualization")
plt.axis("off")

plt.subplot(2, 2, 3)
plt.imshow(sobel_x, cmap='gray')
plt.title("Sobel X (Texture Feature)")
plt.axis("off")
plt.subplot(2, 2, 4)
plt.imshow(sobel_y, cmap='gray')
plt.title("Sobel Y (Texture Feature)")
plt.axis("off")
plt.tight_layout()
plt.show()
```

Teachers should assess CLO2, CLO3 and CLO4 based on the given rubrics (overall weightage 20%)

Recommended Percentage Breakdown

CLO	**Percentage**
CLO2 (Investigation)	10%
CLO3 (Referencing/Citations) *(Turnitin report should be generated.)*	5%
CLO4 (Communication)	5%