Internship Report

By: Sriram Ponnambalam Gangadharan

To: Ms Kajal

Table of Contents

NDT, short for Non-Destructive Testing, is a fundamental discipline within the realm of engineering that is instrumental in assessing the integrity and quality of materials, components, and structures without causing any harm to them. Its significance spans across a multitude of industries, including manufacturing, construction, aerospace, automotive, and energy, where safety and performance are paramount considerations.

The primary objective of NDT is to meticulously detect and analyze defects, irregularities, or imperfections within materials or structures, thereby safeguarding their operational reliability. Through the application of a diverse array of testing methods and techniques, NDT serves as a vital quality control mechanism, ensuring that products and structures conform to stringent standards and regulations. Professionals specializing in NDT undergo specialized training, equipping them with the expertise needed to handle sophisticated equipment and interpret inspection results accurately, all while preserving the structural integrity of the subject at hand.

In this project, NDT is used to inspect pipes to ensure there are no hidden cracks and defects that could compromise the structural integrity and the stability of the pipes. The sensor used to check for defects is the camera. A tiny robotic vehicle with magnetic wheels is mounted with a camera to perform the test.

Python Code:-

```
import numpy as np
import socket
import cv2

# stream =
cv2.VideoCapture('rtsp://192.168.0.253:554/user=admin_password=tlJwpbo6_ch
annel=1_stream=0.sdp?real_stream')
# Reading video via web Camera /Laptop camera
stream = cv2.VideoCapture(0)

# Connection to Ethernet shield
IP = "192.168.0.182"
port = 22000

addr = (IP, port)

#first =
'tsp://192.168.0.253:554/user=admin_password=tlJwpbo6_channel=1_stream=0.s
dp?real_stream'
second = 'rtsp://192.168.0.201:554/ch01.264?dev=1'

names = [first, second]
window_titles = ['first', 'second']
cap = [cv2.VideoCapture(i) for i in names]

frames = [None] * len(names)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(addr)


def send(msg):
    message = msg.encode("utf_8")
    msg_length = len(message)
    send_length = str(msg_length).encode("utf_8")
    send_length += b' ' * (64 - len(send_length))
    s.send(message)

# Detecting Colorline using camera


# # //black color
# gray_lower = np.array([0, 0, 0], dtype=np.uint8)
# gray_upper = np.array([50, 50, 50], dtype=np.uint8)
# # //black color
gray_lower = np.array([0, 0, 0], dtype=np.uint8)
gray_upper = np.array([30, 30, 30], dtype=np.uint8)


# # //blue color
# gray_lower = np.array([100, 0, 0], dtype=np.uint8)
# gray_upper = np.array([255, 100, 100], dtype=np.uint8)

kernel = np.ones((3, 3), np.uint8)
setpoint = 320
previous_ang = 0

while True:
    ret, frame = stream.read()

    grayline = cv2.bitwise_and(frame, frame, mask=cv2.inRange(frame,
gray_lower, gray_upper))
    grayline = cv2.erode(grayline, kernel, iterations=5)
    grayline = cv2.dilate(grayline, kernel, iterations=9)
    grayline_gray = cv2.cvtColor(grayline, cv2.COLOR_BGR2GRAY)

    _, threshold = cv2.threshold(grayline_gray, 1, 255, cv2.THRESH_BINARY)

    contours_gray, hierarchy_gray = cv2.findContours(threshold,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    if len(contours_gray) > 0:
        graybox = cv2.minAreaRect(contours_gray[0])
        (x_min, y_min), (w_min, h_min), ang = graybox

        if w_min < h_min and ang > 0:
            ang = (90 + ang)

        error = int(x_min - setpoint)
        ang = int(ang)
        box = cv2.boxPoints(graybox)
        box = np.int0(box)

        cv2.drawContours(frame, [box], 0, (0, 0, 255), 3)
        cv2.putText(frame, str(ang), (10, 40), cv2.FONT_HERSHEY_SIMPLEX,
1, (0, 0, 255), 2)
        cv2.putText(frame, str(error), (10, 320),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
        cv2.line(frame, (int(x_min), 250), (int(x_min), 300), (255, 0, 0),
3)

# Angular Correction for proper driving
```

```
        if ang >= 88 and ang < 92:
            # cmd = 'F'
            # cmd = cmd + '\r'
            send('F')
        elif ang < 88 and ang > -1:
            cv2.putText(frame, "Turn Left by  ", (90, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
            cv2.putText(frame, str(90 - ang), (400, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
            # cmd = 'L'
            # cmd = cmd + '\r'
            send('L')
        elif ang > 92 and ang < 180:
            cv2.putText(frame, "Turn Right by  ", (90, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
            cv2.putText(frame, str(ang - 90), (400, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
            # cmd = 'R'
            # cmd = cmd + '\r'
            send('R')

# close if color is not black
        elif len(contours_gray) > 0:
            send('X')
    else:
             # cmd = '0'
            # cmd = cmd + '\r'
            # send('P')
            # send('0')
            send('X')
# close the robot/ Video Stream via Screen press q
    cv2.imshow("original with line", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

for c in cap:
    if c is not None:
        c.release()

cv2.destroyAllWindows()
```

Photos :-

Acknowledgments

I thank Janyu Technologies Pvt Ltd., for giving me the opportunity to intern at their prestigious organization.

Ms Kajal was gracious enough to work with me during my internship and supervise me. His/Her/Their exceptional direction, persistent mentoring, and meticulous monitoring were instrumental in the effective completion of the NDT project.

All the individuals that worked along with me at Janyu Technologies Pvt Ltd.; their patience and openness created an enjoyable working environment for me. I acknowledge the help of all of these individuals with an immense sense of gratitude.