**ASSIGNMENT 9**

**NAME: CHAITANYA KRISHNA**

**REG NO: 18BIS0013**

**Build the face detection application using OpenCV.**

**#!/usr/bin/env python**

**"""**

**A ROS node which reads images from a given directory,**

**detects faces, and publishes the images with the detected**

**faces encircled over the sensor\_msgs/Image topic which**

**can be subscribed by rviz for visualisation.**

**"""**

**import sys**

**import os**

**import rospy**

**import math**

**import cv2**

**import glob**

**import numpy as np**

**from std\_msgs.msg import String**

**from sensor\_msgs.msg import Image**

**from cv\_bridge import CvBridge, CvBridgeError**

**## Command-line function.**

**#**

**# Reads and checks the number of command-line arguments**

**# and calls the imag\_pub function**

**#**

**#path\_name\_1 variable is the OpenCV directory**

**#**

**#path\_name\_2 variable is the database directory**

**def main(argv):**

**no\_arg = 2**

**if not len(sys.argv) > no\_arg:**

**print("Please enter only the file paths.")**

**sys.exit()**

**path\_name\_1 = str(sys.argv[1])**

**path\_name\_2 = str(sys.argv[2])**

**imag\_pub(path\_name\_1, path\_name\_2)**

**##Publisher function.**

**#**

**#Detects faces in an image and publishes it.**

**def imag\_pub(path\_name\_1,path\_name\_2):**

**if not os.path.exists(path\_name\_1):**

**print("Path to the OpenCV directory does not exist.")**

**sys.exit()**

**if not os.path.exists(path\_name\_2):**

**print("Path to the image directory does not exist.")**

**sys.exit()**

**pub = rospy.Publisher("sensor\_msgs/Image", Image, queue\_size=10)**

**rospy.init\_node('imag\_pub', anonymous=True)**

**rate = rospy.Rate(0.5) # 1 Hz**

**## loads the required XML classifiers**

**file\_path\_1 = path\_name\_1 + '/data/haarcascades/haarcascade\_frontalface\_default.xml'**

**face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')**

**face\_cascade.load(file\_path\_1)**

**## loads the image in grayscale**

**image\_list=[]**

**file\_path\_2 = path\_name\_2 + '/\*jpeg'**

**for image in glob.glob(file\_path\_2):**

**img = cv2.imread(image)**

**image\_list.append(img)**

**gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)**

**## detects faces & returns positions of faces as Rect(x,y,w,h)**

**faces = face\_cascade.detectMultiScale(gray, 1.3, 5)**

**## draws circles around the detected faces**

**for (x,y,w,h) in faces:**

**square =(w/3)\*\*2 +(h/3)\*\*2**

**radius =int(math.sqrt(square))**

**cv2.circle(img,(x+w/2,y+h/2),radius,(0,0,255),2)**

**## converts OpenCV image to ROS image**

**bridge= CvBridge()**

**out\_image = bridge.cv2\_to\_imgmsg(img, "bgr8")**

**## publishes the image with detected faces**

**pub.publish(out\_image)**

**rate.sleep()**

**if name == '\_\_main\_\_':**

**try:**

**main(sys.argv)**

**except rospy.ROSInterruptException:**

**pass**