Automated Object Tracker

Ms A. Santhiya[1], K. Susindharan[2], D. Surya[3], D. Seshadri[4]

[1]*M.Tech., Assistant Professor, Department of Electronics and Communication, Engineering, Jeppiaar Engineering College, Chennai*

[2,3,4] *Student, Department of Electronics and Communication, Engineering, Jeppiaar Engineering College, Chennai*

***Abstract*—In recent years, drones are used for security purposes, monitoring purposes, and Aerial photography. But capturing pictures and recording videos using drones requires precise manual control and also needs manpower. Hence we proposing this "Automated Object Tracker" project. The main objective of this project is to detect, recognize and track objects precisely using a drone in an automated manner. The following techniques are important in Automatic Object Tracking: Detection, Recognition, and Tracking. Hence we use the Open CV method for the Detection and Face Recognition module for recognition purposes using Python. By programming motion controls, the drone can stabilize its movements using a flight controller according to the detected object. The detection and tracking of the object can be monitored through mobile or laptop.**

***Index Terms*—Automated object tracking drone, Object detection using OpenCV, Object detection, Face recognition.**

I. INTRODUCTION

Drones are capable to revolutionize several areas of our life. The automation process helps us in performing certain tasks quite easier than doing them manually. Few tasks are in development to make it efficient and simple with more accuracy and error-free. Automating the drone operations reduces the damage and makes it more efficient. In this project detection and recognition are performed by Open CV, Face Recognition and Mediapipe. Based on these data, automation can be done by python.

The face recognition method is used to locate features in the image that are uniquely specified. The facial picture has already been removed, cropped, scaled, and converted to grayscale in most cases. Face recognition involves 3 steps: Face Detection, Feature Extraction and Face Recognition. OpenCV is an open-source library written in a programming language like Python. It contains the implementation of various algorithms and deep neural networks used for computer vision tasks.

Mediapipe is a framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nao. MediaPipe Toolkit comprises the framework and the solutions. The following diagram shows the components of the MediaPipe Toolkit. MediaPipe is Google's open-source framework used for media processing. It is cross-platform or we can say it is platform friendly. It is run on Android, iOS, Web, and YouTUbe servers that's what cross-platform means to run everywhere. Pose estimation is a key feature of Mediapipe which is used over a vast field of machine learning, Pose estimation means finding a person's or an object's key points. A person's key points are the elbow, knee, wrist, etc. so Mediapipe can be used for training the ML model to learn the key points and further use the knowledge for specific tasks, this actually can be useful for action recognition.

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage and then Itsezz (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License., OpenCV features GPU acceleration for real-time operations. The OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The main contributors to the project included several optimization experts in Intel Russia, as well as Intel's Performance Library Team. Advance vision-based commercial applications by making portable, performance-optimized code available for free with a

license that did not require code to be open or free itself.
The components used in this project are Raspberry Pi 4, BLDC Motors 1000KV, 30A ESC, 1045 Propeller, Li-ion Battery, APM 2.8 flight controller, MPU 6050, CT6B 6Ch Transmitter and Receiver.

II. LITERATURE SURVEY

1. Study on Face Recognition Technique
Authors: Madhusmita Sahu, Rasmita Dash
Year: 2020
An accurate and efficient face recognition system is a more interesting topic in most industries and research areas. It is a type of biometric information process that is easily adaptable as compared to the traditional card recognition system. Generally, a face recognition system is preceded by a face detection technique. The face detection technique is the preliminary stage to detecting a face in live images. In this paper, some face detection techniques are discussed such as finding skin likelihood images, skin segmentation, the morphological operation for extracting boundary regions, Haar-like features, and the Ada-boost algorithm. This Haar-like feature algorithm continually searches its pattern from the particular face and has better advantages over other techniques. After the face detection technique, the face recognition technology is applied to the detected face for further identification by using some classifiers. Real-time visual object tracking requires learning a robust end-to-end trainable model online during the inference stage. A good tracker needs to exceed the real-time processing level while ensuring accuracy and robustness. Benefit from the booming deep learning technology and increasing maturity of the efficient object detection algorithm, real-time object tracking is moving towards the real application level.

2. Evaluation of Object tracking system using OpenCV in Python
Authors: Hemalatha Vadlamudi
Year: 2020
Object Tracking System is used to track the motion trajectory of an object in a video. First, I use the OpenCV function, select ROI, to select an object on a frame and track its motion using a built-in-tracker. Next, Instead of using select ROI, he uses YOLO to detect an object in each frame and track them by object centroid and size comparison. Then he combines YOLO detection with the OpenCV built-in tracker by detecting the object in the first frame using YOLO and tracking them using select ROI. Video tracking is widely used for multiple purposes such as human-computer interaction, security and surveillance, traffic control, medical imaging, and so on. Object tracking is a very challenging task in the presence of varied Illumination conditions, background motion, complex object shapes partial and full object occlusions. Object detection and location in digital images have become one of the most important applications for industries to ease users, save time and achieve parallelism. This is not a new technique but improvement in object detection is still required in order to achieve the targeted objective more efficiently and accurately. The main aim of studying and researching computer vision is to simulate the behaviour and manner of human eyes directly by using a computer and later on develop a system that reduces human efforts and shows the basic block diagram of detection and tracking.

III. PROPOSED SYSTEM

Existing System
Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results. Object detection combines classification and localization to determine what objects are in the image or video and specify where they are in the image[17]. It applies classification to distinct objects and uses bounding boxes, as shown in fig 3.1.

Fig 3.1 Object Detection

Object detection is one of the fundamental problems of computer vision. It forms the basis of many other downstream computer vision tasks, for example, instance segmentation, image captioning, object tracking, and more. Specific object detection applications include pedestrian detection, people counting, face detection, text detection, pose detection, and number-plate recognition. Person detection is a variant of object detection used to detect a primary class "person" in images or video frames. Detecting people in video streams is an important task in modern video surveillance systems. Object detection is completely interlinked with other similar computer vision techniques such as image segmentation and image recognition that assist us to understand and analyze the scenes in videos and images.

Block Diagram

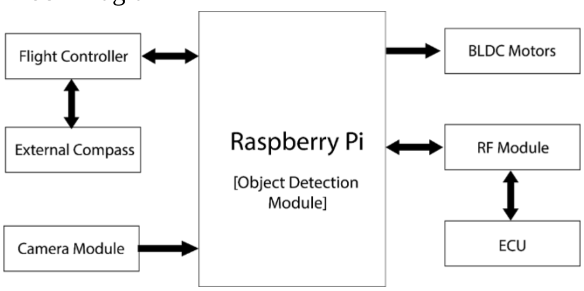

Fig 3.2 Existing System Block Diagram

In fig 3.2 it is shown that the camera module captures the image /video and then the captured footage is sent to Raspberry Pi where the object detection algorithm detects the object. Here flight controller uses an external compass for proper flight movement and stability. RF Module and ECU are used for manual drone control.

Drawbacks

This existing system has the capability to detect the object in the capturing image or video. The system will detect the object or person in a manual method. The drone must be operated manually using a remote control so that the movement control of the drone would be difficult and needs experience and manpower. Sometimes this may lead to crashing or improper flight of the drone or one need extra manpower to operate the drone. The object detection of the drone will not be precise. These are some of the drawbacks of the existing system. In order to overcome these issues, an automated object tracking method can be used. By including the object tracking feature the drone can automatically track a particular object or person. By implementing object detection and tracking, the drone can detect and track the person or object automatically.

Proposed System

To overcome the drawback in the existing system we made this Automated Object Tracking Drone. With this idea, we can automatically track the object or person with very less effort.

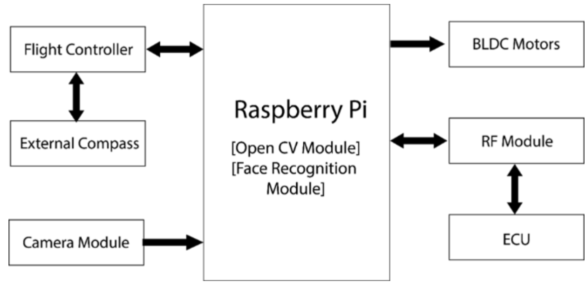

Fig 3.3 Proposed System Block Diagram

In fig 3.3. it is shown the block diagram of the proposed system. In this project, the camera module captures the image/video and the Raspberry Pi which contains an Open CV and Face recognition module process the footage and then detects the particular object by face recognition module and tracks the object by Open CV module.

Configurations

The configuration of the flight controller is one of the important parts of a flying drone. Flight Controller can be configured using Mission Planner Software.

In this project, APM 2.8 flight controller has been used. The flight controller helps in stabilizing the flight motion of the drone while flying[7]. After installing firmware to the flight controller, the calibration must be done. The user should choose the frame type and mainly needs to do accel calibration (acceleration calibration) and compass calibration. The compass calibration is used to configure the flight controller's compass for better navigation. Then Radio calibration must be done. It is used to calibrate the radio signal from the remote controller. Other calibration like ESC calibration, Flight modes and Fail-Safe system must be done In this project, Raspberry Pi 4 has been used for processing purposes and a Pi camera has been used for capturing images or videos. OpenCV method using Python programming is used for object detection and tracking[6]. OpenCV is a pre-built, open-source CPU- only library (package) that is widely used for computer vision, machine learning, and image processing applications. OpenCV has a bunch of pre-trained classifiers that can be used to identify objects such as trees, number plates, faces, eyes, etc[11]. We can use any of these classifiers to detect the object as per our needs. This will make a boundary box with (x,y) coordinates over the detected object link in fig 3.4. [13] In the program, the OpenCV module and face recognition module are used. The image needs to track is uploaded into the storage if the coordinates of the image matches with the capturing image or video then the particular object or person could be detected and tracked[15]. Based on the position and motion of the detected object the drone will reposition itself to capture that particular object or person. Raspberry Pi has been used as the processing unit. The code is uploaded to the Raspberry Pi 4. In this project, Raspberry Pi 4 4GB version is used. The drone will capture the images or videos using Pi Camera. The captured image or video will be stored in the storage.

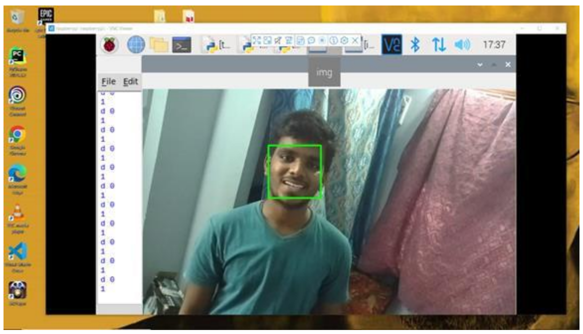

Fig 3.4 Face Detection using Face recognition module

Functioning of the drone

The main function of the drone is to detect and track the object or person automatically using OpenCV and Face Recognition method. The image of the target should be uploaded into the storage. The drone will analyze the screen and detect and track the particular target. It will try to reposition itself in order to capture the target while moving. fig 3.5 shows the flow chart of the Automated object tracking drone functioning.

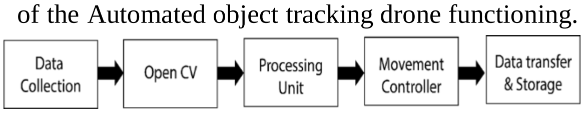

Fig 3.5 Flowchart of the object tracking process

The above mentioned is the flow chart of the project. The data of the image or video has been collected and sent to processing. Here OpenCV tries to analyze the collected data for matching faces or objects in the library using the face recognition module. If the object or person is matched then it will detect and track the particular object or person. Then instruction will be given to the flight controller and based on the object's motion it will reposition itself. The captured video will be stored in the storage. Some of the pictures of the drone are given below. The initial take-off of the drone is shown in fig 3.6. In fig 3.7 and fig 3.8 the top view of the drone is shown.

Fig 3.6 Initial take-off of the drone

Fig 3.7 Top View of the Dron

Fig 3.8 Front View of the Drone

IV. RESULTS AND DISCUSSION

Face Recognition

The face recognition module captures the face encodings of the tracked face. The first captured face's encoding is stored for the further process of recognition. Based on the stored encoding, the real-time captured face encodings are compared and identify the perfect match. In this process, face location is attained with which control signal is generated and passed on to the ESCs. The processed frame is shown in fig 4.1.

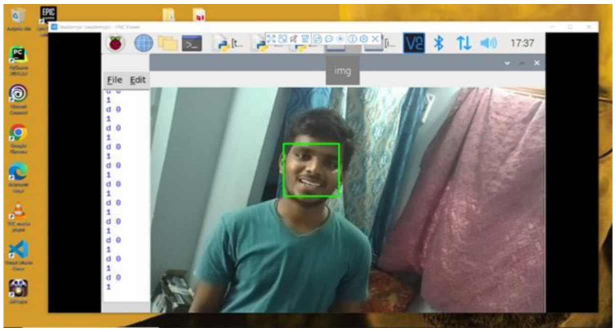

Fig 4.1 Face detection using Face recognition module

The face recognition module identifies the target face even if the person is not facing straight. It identifies the position of the face and captures the face encodings. Even if the encoding got is for a partial face, it compares that portion alone gives the output. The output of the partial face captured and recognized is shown in fig 4.2.

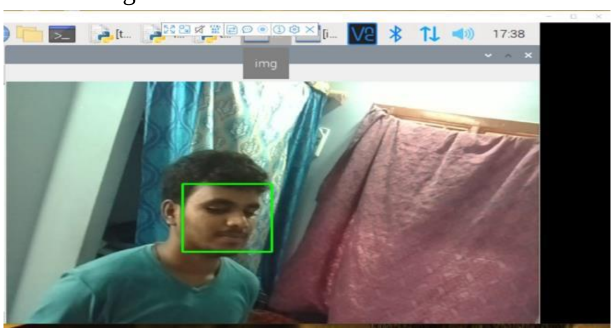

Fig 4.2 Face Detection with partial face captured

Mediapipe

After the match is found the process is then moved on to the next stage. Here using the Mediapipe the facing direction is determined as shown in fig 4.3. With this determined facing direction, the drone is moved to the perfect position in which the camera captures the complete face of the person. Till this process is completed, the code is iterated continuously and the drone is moved in a circular path.

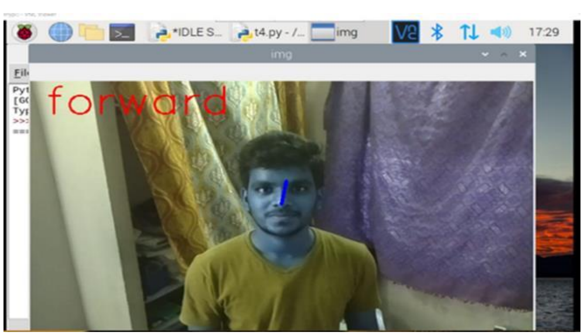

Fig 4.3 Facing position using Mediapipe

The detection of the facing direction is determined by the Mediapipe module and the outputs of this module are shown in fig 4.4 and fig 4.5.

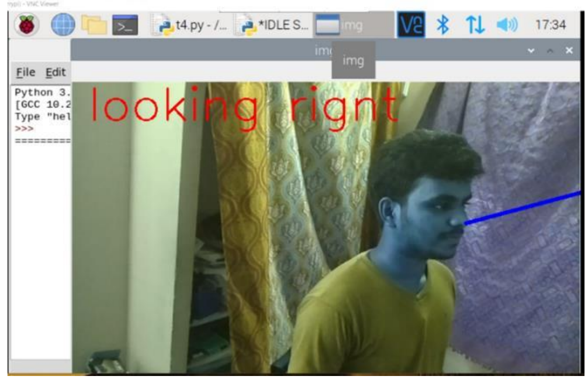

Fig 4.4 Image facing right

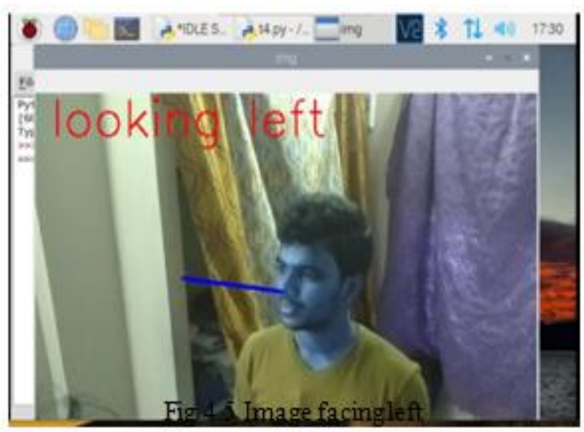

Fig 4.5 Image facing left

OpenCV

Using OpenCV the input image is captured and read, this input is further sent for processing. Based on the data from that process the control signals are generated. After the completion of these processes, every frame is merged and converted as a video and stored with an extension of .avi with a fixed frame rate as specified.

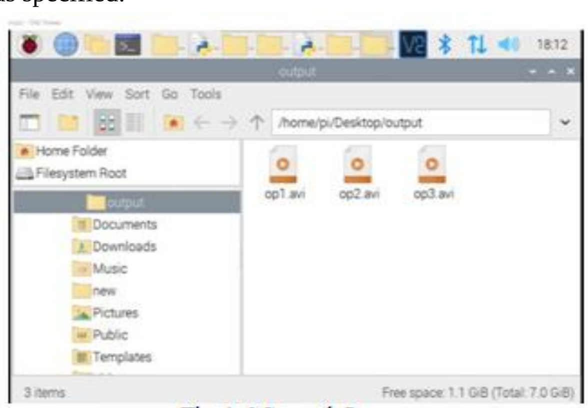

Fig 4.6 Stored Output

Condition Statement

After all these recognition processes are completed the condition statements are used to decide the movement direction of the drone. First based on the data got by the Mediapipe function, the camera position is altered and fixed to the square facing angle. After this process based on the data got by the face recognition module, the position of the face is fixed at the centre of the frame by moving the drone in a certain required direction.

V. CONCLUSION AND FUTURE SCOPE

The recognition function is performed and the required data are collected from this process. Based on the data collected control signals are generated with the help of a few condition statements. After this process, the captured video is stored for further usage. The control signal is fed to the motor and flight tested. In order to achieve face recognition and object tracking the Open CV method and face recognition module are used. By upgrading the capturing device the quality and depth of the footage will be improved also proportional to the quality the precision of the object detection and tracking will be improved.

This technology can be used for videography automation, and photography. The camera resolution can be improved to increase the clarity of the captured video. A further efficient algorithm can be created to improve the processing speed and accuracy. An android application can be developed to give beatter user experience. Additional features can be added to make it more convenient and user friendly.

REFERENCE

[1] Alexander A S Gunawan, Andry Chowanda, Aurello Patrik, Gaudi Utama, Jarot S. Suroso, Widodo Budiharto (2018) “Fast Object Detection for Quadcopter Drone Using Deep Learning”, Vol. 8, pp.13 – 15.
[2] Ali Hamdi, Du Yong Kim, Flora Salim (2005) “DroTrack: High-speed Drone-based Object Tracking Under Uncertainty”, Vol. 52, pp. 1 – 3.
[3] Anis Koubaa, Basit Qureshi (2018) “DroneTrack: Cloud-Based Real-Time Object Tracking Using Unmanned Aerial Vehicles Over the Internet”, Vol. 7, pp. 13810 – 13824.
[4] Anitha Ramachandran, Arun Kumar Sangaiaha (2021) “A review on object detection in unmanned aerial vehicle surveillance” Vol.35, pp. 215 – 228.
[5] Damilola Williams Dada, Olubukola D Adekola, Oluwatobi Tolulope Saliu, Onyedikachi Kenny Udekwu (2021) “Object Tracking-Based Follow-Me Unmanned Aerial Vehicle (UAV) System”, Vol. 41, No. 3, pp. 875 – 890.
[6] Danfeng Hong, Qian Du, Ran Tao, Wei Li, Xin Wu (2021) “Deep Learning for UAV-based Object Detection and Tracking: A Survey”, Vol. 21, pp. 91 – 124.
[7] Eduardo Sanchez Morales, Friedrich Kruber, Michael Botsch, Samarjit Chakraborty (2004) “Vehicle Position Estimation with Aerial Imagery from Unmanned Aerial Vehicles”, Vol. 12, pp. 81 –84.
[8] Fadi Al Turjman, Muhammad Sher Ramzan, Shehzad Ashraf Chaudhry, Zeeshan Ali (2020) “Securing Smart City Surveillance: A Lightweight Authentication Mechanism for Unmanned Vehicles” Vol. 08, pp. 43711 – 43724.
[9] Zixuan Wang, Zhicheng Zhao, Fei Su (2020) “Real-time Tracking with Stabilized Frame” Vol. 09, pp. 27 – 38.
[10] GaoangWang, Haotian Zhang, Jenq-Neng Hwang, Zhichao Lei (2019) “Eye in the Sky: Drone- Based Object Tracking and 3D Localization”, Vol.8, pp. 101 – 128.
[11] Hao Yang, Xiaofeng Han (2020) “Face Recognition Attendance System Based on Real-Time Video Processing” Vol. 08, pp. 159143 – 159150.
[12] Hemalatha Vadlamudi (2020) “Evaluation of Object Tracking System using Open-CV In Python” Vol. 09, Issue 09, pp. 151 – 163.
[13] Jiani Hu, Weihong Deng, Yaoyao Zhong, Yuying Zhao, Zhanf An (2019) “APA: Adaptive Pose Alignment for Pose-Invariant Face Recognition” Vol. 07, pp. 14653 – 14670.
[14] Maciej L. Pawelczyk, Marek Wojtyra (2020) “Real-World Object Detection Dataset for Quadcopter Unmanned Aerial Vehicle Detection” Vol.08, pp.174394 – 174409.
[15] Madhusmita Sahu, Rasmita Dash (2020) “Study On Face Recognition Technique” Vol. 05, pp. 127 – 140.
[16] Song-Chen Han, William Shen, Zuozhen Liu (2016) “Deep Drone: Object Detection and

Tracking for Smart Drones on Embedded System", Vol. 21, pp.140 – 160.

[17] Song Han, William Shen, Zuozhen Liu (2020) "Deep Drone: Object Detection and Tracking for Smart Drones on Embedded System" Vol. 07, pp.131 – 147.