IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 July 2024, accepted 13 August 2024, date of publication 22 August 2024, date of current version 4 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3447899

RESEARCH ARTICLE

Autonomous UAV Implementation for Facial Recognition and Tracking in GPS-Denied Environments

DIEGO A. HERRERA OLLACHICA, (Member, IEEE), BISMARK K. ASIEDU ASANTE, (Member, IEEE), AND HIROKI IMAMURA, (Member, IEEE)
Department of Information System Science, SOKA University of Japan, Hachioji, Tokyo 192-8577, Japan

Corresponding author: Diego A. Herrera Ollachica (e20d5303@soka-u.jp)

This research was supported by the Science and Technology Research Partnership for Sustainable Development (SATREPS; Grant Number JPMJSA2005) funded by the Japan Science and Technology Agency (JST) and the Japan International Cooperation Agency (JICA). The project's website is: https://www.soka.ac.jp/en/satreps-earth/project/. As a PhD student, I worked as a research assistant for this project, and the drones developed for this paper will be used for research purposes in the SATREPS-EARTh project.

ABSTRACT Surveillance with facial recognition holds immense potential as a technological tool for combating crime in Latin American countries. However, the limitations of fixed cameras in covering wide areas and tracking suspects as the evaded recognitions systems pose significant challenges. To address these limitations, we propose a facial recognition system designed to recognize faces of suspected individuals with criminal backgrounds and missing persons. Our solution combines facial recognition technology with a custom-built unmanned aerial vehicle (UAV) for the identification and tracking of targeted persons listed in a database for crimes. We utilize the inception v2 model to deploy a Siamese network on the Jetson TX2 platform for facial recognition. Additionally, we introduce a novel tracking algorithm to track suspected individuals in the event of evasion. During field test experiments, our system demonstrated strong performance in facial recognition across three different environments: stationary, indoor flight, and outdoor flight. The accuracy of our system was 94.45% for recognizing along with our tracking algorithms. An improvement of 1.5% in recognition and better tracking approach for surveillance. This indicates the versatility and effectiveness of our solution in various operational scenarios, enhancing its potential for crime prevention and law enforcement efforts in Latin American countries.

INDEX TERMS Unmanned autonomous vehicles (UAV), facial recognition, object tracking, deep learning, autonomous flight, embedded systems.

I. INTRODUCTION

The last decade, has been marked by the growth and spread of crime, violence, and the disappearance of people in Latin America, with an increase of up to 11 percent in these incidents between 2000 and 2010, which caused more than 1,000,000 deaths [1]. In 2020, Latin America reported more than 150,000 victims of intentional homicide [2]. Thus, Latin America is often described as the most violent region in the world [3]. This situation has become even more challenging, as the policies aimed at reducing crime in Latin America often rely on approaches that have proven to be ineffective. On the other hand, the promising solutions are linked to the use of information technologies that are yet to be fully exploited [3].

One of the technologies contributing to solving this problem is facial recognition, by recognizing the face of a person we can identify suspected persons whose information is available in the police database. Currently, this technology is used with fixed-position cameras but the system can be improved by using UAVs, to cover more areas and prevent culprits from invading these fixed-position cameras. Some other drone companies have achieved facial detection and

The associate editor coordinating the review of this manuscript and approving it for publication was M. Anwar Hossain.

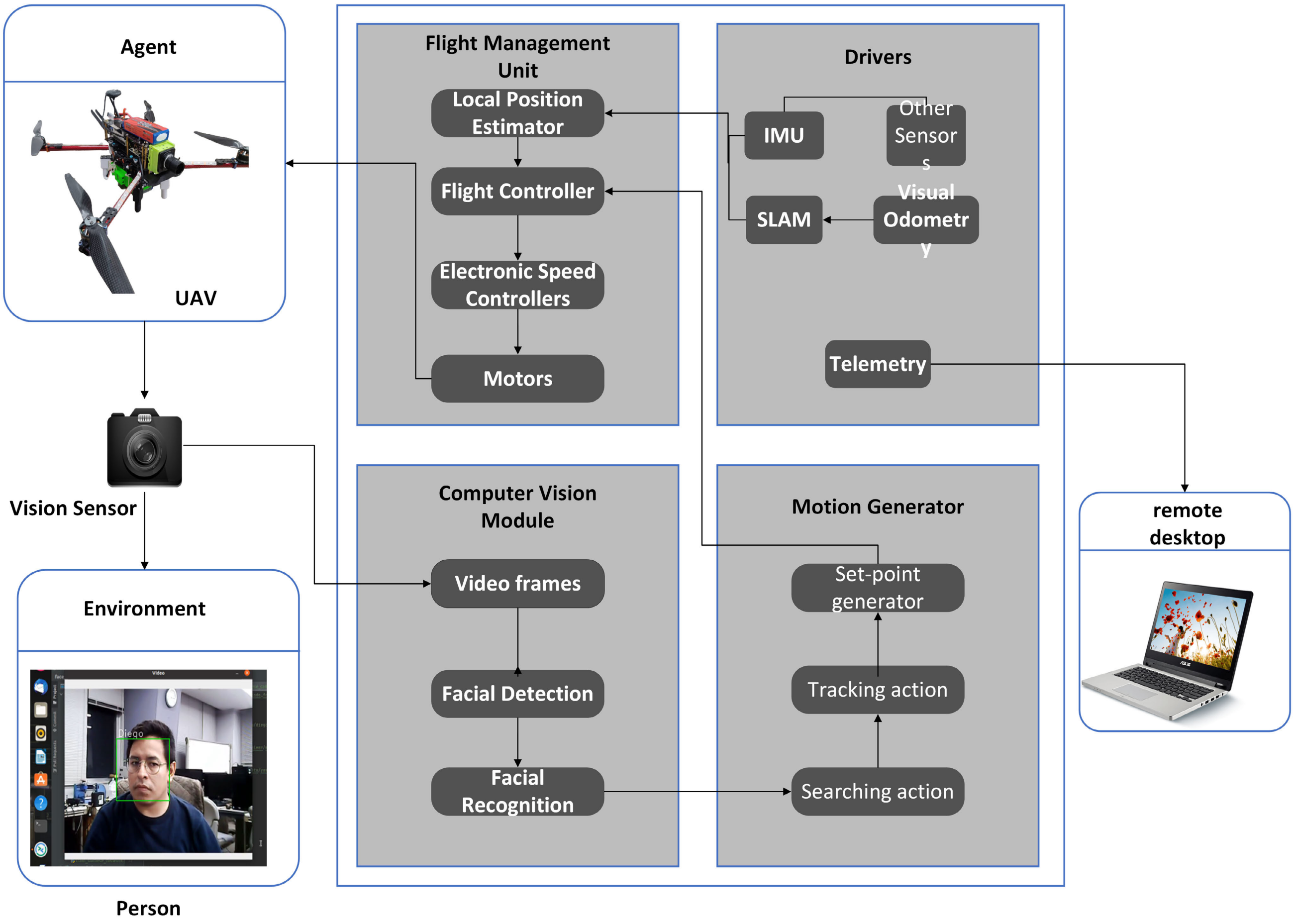

FIGURE 1. System architecture of the drone. 1) Flight Management Unit (FMU). 2) Drivers, bring all the sensors necessary for autonomous drone flight. 3) Computer Vision module for facial recognition. 4) Motion Generator module that generates new setpoints.

tracking the detected faces but they are not capable of identifying who is the person in front of the drone [3].

The facial recognition system is an advanced method designed to detect and recognize a person in a digital image or video source. The facial recognition system continues to advance each year, to the extent that it can accurately identify individuals even after they have undergone plastic surgery. [4], [5]. This achievement is due to the artificial intelligence algorithms becoming more sophisticated due to free access to vast amounts of data for training the algorithms. These artificial intelligence algorithms are expanding their capabilities to different areas of daily human life [4]. The current state-of-the-art facial recognition technology has made substantial advancements in various fields, especially security. According to recent studies, facial recognition systems are better at identifying individuals and acquiring information such as name, age, and nationality [6] and some airports are using facial recognition instead of using boarding passes [7].

Facial recognition systems have been developed with machine learning and deep learning algorithms. Several methods for developing facial recognition systems exist, for instance, support vector machine (SVM), local binary pattern histogram (LBPH), Eigenfaces, and deep neural networks [8]. One of the prevalent deep convolution models in use today is VGG16 [9], [10], which evaluates the depth of the convolutional network and its precision in extracting features for recognition tasks in large-scale images using very small convolution filters (3 × 3). Another widely adopted model is FaceNet [11], which incorporates the ‘triplet loss’ function described in section two.

Several state-of-the-art deep learning models have produced exciting results with high accuracy in various computer vision tasks [12]. The accuracy of DNN has been shown in identifying metastatic breast cancer, where it improves detection to 98.4% [13]. In the same vein, one of the several computer vision tasks is facial recognition which uses deep convolutions neural networks and is being used widely in the research community however one of the problems is the quantity of data to train the network. To overcome this challenge, we choose a Siamese network which was proposed by Koch et. al [14], the Siamese network is among the state-of-the-art models for recognizing faces, we need a few images of each subject to recognize their faces. Due to its popularity as an excellent feature extractor, we have chosen the deep convolutions neural network DCNN as the base network to build the Siamese facial recognition system.

Facial recognition systems are often integrated into fixed camera surveillance systems, this leads to a constraint on the range of view thus making the system inefficient in monitoring wide areas. With the fixed camera systems, it can only identify a person located in front of it, this leads to three main problems, 1. limited tracking ability, 2. limited coverage area, and 3. lack of adaptability. In this manner, drones can track a person from a distance using various algorithms for tracking, this technology addresses the lack of tracking capabilities of the existing systems. Drones can cover vast areas, thereby resolving the second main problem of small coverage areas. Furthermore, drones can easily adapt to diverse situations and environments by adjusting their position, changing the camera's vision angle, and altering their location. Thus, the use of drones enables us to overcome the three main problems described earlier.

In recent studies, drones integrated with deep learning systems have emerged as highly effective tools for solving problems quickly in diverse fields. Particularly in the context of rescuing injured individuals, where time sensitivity is paramount, drones play a pivotal role in the timely detection of missing persons [15]. Drones are also employed in the field of security to enable machines to interpret human behavior, for example, in surveillance, a drone can detect human poses in motion, in sports, drones can identify human behavior to obtain information using human pose estimation [16]. Additionally, employing drone-captured images for citizen safety entails analyzing human behavior patterns, adding a layer of sophistication to security measures. Our research is also related to citizen security through the use of deep learning models to help in solving the problem of identifying criminal suspects.

Using drones with integrated neural network systems can contribute to the reduction of humans making contact with disease-transmitting agents such as birds when observing the agents [17]. Targeted application of pesticides in large-scale commercial farms is another exciting use of drones which contributes to the well-being of plants and protects already healthy plants [18]. Similarly, in the study of marine species, aquatic drones can assist in recognizing different fish species using deep learning models such as googleNet [19] and AlexNets [20], thereby providing statistical data for marine life preservation efforts [21]. The interaction between humans and drones has recently received more attention in the academic field. Drones can detect a person, and the individual can send their current position and health status via a smartwatch. Subsequently, the drone can track the detected person using its own video feed and smartwatch data, allowing for the assessment of the physical condition of the person [22].

Significant progress has been made in UAV tracking and control recently. Ma et al. introduced an algorithm based on deep reinforcement learning for controlling vertical take-off and landing (VTOL) UAVs amidst wind disturbances, achieving high accuracy and robustness in tracking and flight stability [72]. Similarly, Xu et al. proposed a reinforcement learning-based control method for UAV formation in GPS-denied environments, which optimizes control policies and minimizes collision risks to enhance UAV swarm management [73]. These studies highlight the efficiency of reinforcement learning in boosting UAV tracking accuracy and operational performance, which aligns with our research aims. However, there are substantial differences between these studies and our approach. Ma et al. primarily focus on maintaining the position of the UAV and stability under environmental disturbances [72], while Xu et al. concentrate on keeping UAV formations intact in GPS-denied environments [73]. Conversely, our research centers on the detection, recognition, and tracking of specific individuals using facial recognition technology. By employing a Siamese network for facial recognition and a unique tracking algorithm that maintains the control of the drone along with the Extended Kalman filter (EKF) algorithm, our UAV system aims to follow individuals identified from a criminal database, addressing the specific challenges of crime prevention and law enforcement in Latin American countries.

In this research, we aim to provide a solution to the challenges of citizen insecurity affecting several developing nations. To achieve this, we present a novel integration of facial recognition technologies using transfer learning and autonomous UAVs. Our system stands out by incorporating deep learning algorithms and a custom-built drone platform to perform real-time detection, recognition, and continuous tracking of individuals. We use a compact Jetson TX2 computer, the Pixhawk 4 flight controller, and the T265 positioning camera. The system runs on the ROS Melodic framework, which includes all the necessary nodes to operate the system in real time.

The focus of this research is not only on face detection and recognition but also on tracking, some studies have been oriented toward object tracking using drones, in this regard, several research studies are addressing this topic, as mentioned in the work of Mukashev et. al [22], a drone detection and tracking system is developed using the YoloV3 algorithm and the CSRT algorithm provided in the OpenCV library to detect and track humans. Tracking multiple objects using videos captured by a drone is valuable for surveillance and defense purposes, which is why Kim et. al [23], an innovative algorithm for object detection and tracking has been developed, enhancing the Joint Detection and Tracking (JDT) algorithm [24]. Obstacle detection and tracking using drones constitute a broad field of research [25], with new studies continually showing promising results. However, in the area of face detection, recognition, and tracking, there are still relatively few research efforts. In many countries, crime and violence are significant concerns, and the police play a crucial role in crime prevention. However, there is a lack of technology that can assist in this regard. Therefore, this paper develops a system for face detection, recognition, and tracking using an autonomous drone. The main contribution of this research is as follows:

- We introduce a simple but novel algorithm, simple matching real-time tracking (SMRT), designed to match the ID generated by Simple Online and real-time tracking (SORT) with the identity generated by the Siamese network. This algorithm significantly enhances tracking accuracy and efficiency, providing a major improvement over existing tracking methods that often struggle with identity consistency across frames. Details are elaborated in section two of this study.
- We propose the design and implementation of a novel autonomous drone capable of recognizing faces within a database by leveraging deep learning algorithms for the detection, recognition, and innovative method for continuous tracking of recognized faces. This combination provides a unique advantage over existing systems that typically do not integrate these capabilities on a single, autonomous platform.
- To integrate the Jetson TX2 and its associated components seamlessly, we meticulously designed and assembled the drone using SolidWorks CAD software. The Intel RealSense T265 camera was strategically chosen to capture precise positional and orientational data for the drone with the Extended Kalman Filter algorithm, enhancing its ability to operate in GPS-denied environments. This customized approach was imperative to fulfill the unique project requirements and is a significant improvement over conventional systems that rely heavily on GPS.

This research presents a unique integrated solution that combines the detection, recognition, and continuous tracking of individuals through the use of an autonomous drone system. While current facial recognition and drone tracking systems have shown advancements separately, our proposal stands out by merging these capabilities into a single autonomous platform. This integration not only enhances tracking accuracy but also significantly extends the coverage and adaptability of the system, overcoming the limitations of fixed-camera systems and current drone-based solutions. This novel approach enables more efficient and effective tracking of individuals in various situations, providing a clear advantage over existing methods in the field. This paper is organized in the following way, In Section Two, we present six pieces of research that are related to our research in two main aspects, one is facial recognition used in drones, and the second is drone applications. In the third section, we present the system architecture, the drone implementation, and the robot operating system (ROS) architecture, and then we present in detail the Siamese network used in this research. In addition, the fourth section shows the experiment of the drone running in two different environments, not flying the drone, indoor flights, and one outdoor test, as well as discusses the confusion matrix. In the fifth section, we discuss the results and the findings we have obtained from the experiments. Finally, we concluded with a summary of the research and suggested future works. It is expected to reduce crime and violence and increase civil security by using this research as a plan to develop sophisticated drones [26].

II. RELATED WORKS

Surveillance is an important aspect in our communities to safeguard the integrity of our citizens against crime, violence, and kidnapping. The Colca Canyon is one of the deepest canyons in the world and people usually get lost which causes complicated search and rescue tasks due to the geography of the place [27], for this reason, it is necessary to deploy the facial recognition system in autonomous drones. Facial recognition is a technology that is used every day in different applications and some research is using this technology to create more sophisticated applications for different environments which means real situations. However, the versatility of using drones raises concerns regarding potential uses for malicious purposes, thus, research is underway to develop systems for drone detection using other drones for defense purposes [28], [29]. In this section, we are sharing related works concerning the three key aspects we are focusing on in our research: facial detection, facial recognition, and tracking. Notably, we've integrated these aspects into a drone that we crafted in our laboratory.

A. DRONE FACE DETECTION

Detecting faces is the first step to beginning the facial recognition system, however, detecting faces during a flight has its challenges as the vibration of the camera is caused by the rotation of the engine which can affect the recognition of the person in front of the drone. Besides, the distance between the person and the camera of the drone can affect the recognition as well. Hsu and Chen et. al [30] experimented to detect faces at different heights from the ground to the drone and different distances from the face of the person to the drone. Fig. 2 illustrates the experiment for taking pictures with the stick that represents the drone, according to Hsu and Chen et. al [30], the performance of detecting faces using deep learning methods such as Face++ [31] and Rekognition API [32] is better than some other traditional techniques [33].

In the same way, the authors mention that this is an empirical study to evaluate the different factors that may affect face detection in drones [33]. Hence, the issue using a stick is that it does not simulate real drone conditions so face detection may not be good, and facial recognition is not performed. Though the results for face detection are quite good, the inference has not been tested in a drone or onboard computer so is it not possible to analyze the performance of the system in a real situation.

B. DRONE FACE RECOGNITION

Facial recognition with drones for tasks such as monitoring, and person identification is gaining prominence.

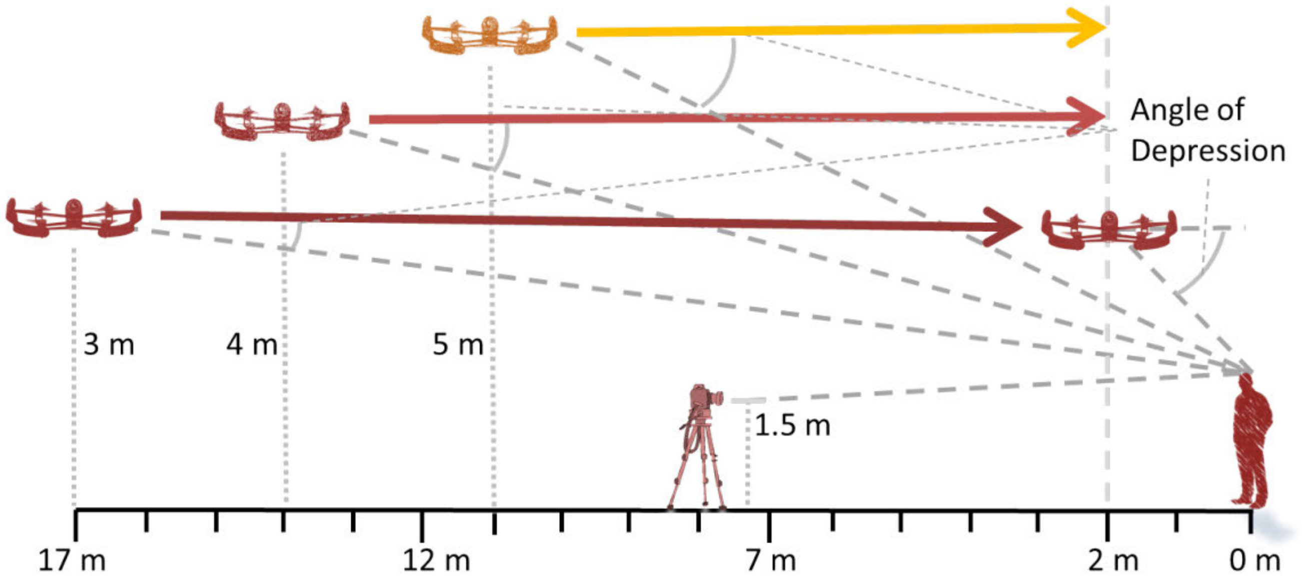

FIGURE 2. An illustration depicting the data collection experiment made by [29]. Images of faces were taken at different distances and altitudes and collected to experiment with facial recognition with drones.

Jurevičius et al. [34] in the work showed how a task to recognize faces using drones as the video source is possible. To detect and recognize a face they are using the package Dlib [35] which uses the histogram of oriented gradients (HOG), and the resnet_v1 model [36] for recognition that is included in the Dlib package. The database consists of 13233 faces that were kept in the SQL database. The video frame is captured by the Raspberry Pi camera mounted in the DJI Mavic Pro drone. The problem of using a small single-board computer such as the Raspberry Pi to transmit video usually has many delays and loss of important data in the transmitted images, in addition, image processing and facial recognition are carried out on a remote server, which means that it depends of a continuous Wi-Fi connection. Our approach is the implementation of the Siamese network model in the same drone in real-time, this drastically reduces the time of sending and receiving images and also increases exponentially the response of the drone to unwanted events.

Surveillance and violence detection are among the exciting applications of facial recognition when implemented in drones. Srivastava et. al [37], proposed a new method to detect violent situations between people and identify the individuals involved in the violent scene. They are using seven different Imagenet models VGG16 [10], VGG19 [10], ResNet101V2 [38], DenseNet201 [39], InceptionV3 [40], MobileNet [41], and NASNet [42] plus three combinations of two models to analyze the best architecture to recognize violence. Besides, they propose a new ResNet-28 architecture to do facial recognition. Hence, both, violence detection and facial recognition are not trained from scratch, instead, they use transfer learning techniques to add layers and train the last layers of the architecture. For violence detection they are using two databases, one called the hockey dataset [43], and the other one called the real-life violence situation (RLVS) dataset [44], but not mention the amount of data stored in the database for facial recognition. Nevertheless, the entire system can recognize faces with 99.20% accuracy. The main issue is that the entire system is not fully or semi-autonomous since the drone must be controlled from a ground station, also the drone cannot follow the person. Similarly, video and violent scene recognition are processed on a computer, so the real-time accuracy cannot be precisely determined by the drone itself.

Autonomous drones are being researched since the human pilot cannot fly the drone every time it is required to do a task and in search and rescue tasks, it is important to have several drones working. Hence, a UAV for detecting people and objects in cluttered indoor environments was developed by Sandino et. al [45]. The drone uses an onboard computer UP2 together with a Vision Processing Unit(VPU) to boost the computations, the research uses the Google MobileNet SSD [41], which is deployed in the framework Caffe and tuned with the pre-trained weights from PASCAL VOC2012 dataset [46]. Therefore, they use the Partially Observable Markov Decision Process(POMDP) to model the navigation problem and solve it in real-time by using the Augmented Belief Trees(ABT) algorithm [46], [47]. Since this research has a good approach to navigating indoor environments with obstacles, in future work, we can either use this approach or enhance it. However, this research does not perform people tracking, which is the main focus of our investigation.

C. DRONE FACE TRACKING

Several research studies have used drones with cameras to capture video frames and then process them on a personal computer to detect and recognize faces using different methods [48]. Tracking a face is another task that an autonomous drone must do in real-time. Hence, the DJI Tello drone can be used for face detection and tracking, the DJI Tello drone has a software development kit (SDK) where we can implement a Python script to detect faces and by reading the values of the sensor inside the drone it is possible to follow the face in front of the drone [49]. Priambodo et. al [49] mention that the system uses a haar cascade classifier to reduce the computational cost, hence, the DJI Tello drone cannot recognize faces, but it can detect and follow them.

A different research related to autonomous UAVs is a hunting drone. Wyder et al. [50] is developing a novel drone to detect, track, and follow another drone by using a pre-trained Tiny Yolo model. Besides, they implement a linear regression model to predict the next position of the target drone. Hence, they used YOLO's Darknet-53 [51] as a pre-trained model to train a tiny Yolo model, they collected a total of 58,647 pictures as a database. Therefore, this drone is autonomous since they are using the Intel realsense T265 tracking camera to obtain the position and orientation of the drone and to communicate the flight controller with the onboard computer using mavlink-ros bridge protocol [52]. The results of this research are promising since it can achieve its goal with a good performance and 77% accuracy in a cluttered environment. While this research focuses on drone tracking, our research centers on the implementation of autonomous drones that detect and recognize faces. Based on the position of the face relative to the camera frame, new XYZ coordinate setpoints are calculated for the drone to move. The novelty of our research lies in the SMRT algorithm and facial tracking using the face's relative position within the camera frame.

Object or person tracking can also be achieved through a human-machine system, Zhou and Liu [53] proposes a comprehensive human-in-the-loop tracking framework with two main modules. The Local Tracking Module employs the SiamRPN model, enhanced with a human-attention-guided approach to improve tracking accuracy around the human visual focus. The Human Attention Analysis Module identifies Targets of Human Interest (TOHI) by analyzing eye movement patterns and accumulated attention time, enabling effective tracking correction within and outside the visual focus area. Furthermore, in contrast to the mentioned research, our research specifically focuses on face detection, recognition, and tracking. Our goal is to develop an autonomous drone capable of identifying individuals independently, without human intervention, representing a comprehensive approach towards autonomy in person identification. Therefore, the detection, recognition, and tracking of objects using drones is an area that has been under research due to its numerous applications.

III. METHODOLOGY

In this work, we present an autonomous drone that recognizes faces within a database of faces using a Siamese network, a deep neural network for comparing the similarity between features of two given inputs of faces, the recognized faces are then tracked with our novel tracking algorithm. Given an image captured by the drone, the Siamese network determines whether the person belongs to a suspect listed in a database for a crime or not, the face of the recognized suspect is then tracked using our proposed SORT and SMRT algorithms, and the tracking algorithm calculates and outputs specific coordinates called setpoints (X, Y, Z) in a 3D space based on the position of the face of the recognized suspect, and then the generated setpoints are sent to the flight controller, the flight controller receives the new setpoints and begins to track the person in front of the drone and if there is no person in front of the drone, it will start rotating over the z-axis in search of a new face to detect. An illustrated overview of the proposed system is shown in Fig. 3. In this research, the subjects have given their consent to carry out the experiments.

A. SYSTEM ARCHITECTURE

Fig. 1 shows the four modules of the system architecture: the computer vision module, the motion generation module, the flight management unit module, and the driver module. Our drone system comprises a conventional USB camera connected to the onboard single-board computer, alongside an array of sensors dedicated to distinct subtasks during autonomous flight. The computer vision module, integrated into the onboard single-board computer, employs the Haar cascade function for face detection and a Siamese network for face recognition within a pre-established database of crime suspects. Coordinating with the motion generator, this module executes algorithms to search for new faces and track recognized ones within the surroundings. Both the computer vision module and the motion generator are hosted on the Jetson TX2 embedded within the drone. The driver module is responsible for reading the drone's IMU, magnetometer, and other sensors to obtain specific data, such as battery voltage and other controller data. Additionally, it is responsible for performing Simultaneous Localization and Mapping (SLAM) to obtain the drone's position and orientation. On the other hand, the flight management module is responsible for collecting all data from the previous modules and, based on that data, such as the local position, it controls the motor electronic speed controllers (ESCs) to reach the required final position. For real-time monitoring and intervention, a remote desktop station observes the drone's autonomous activities. This enables manual intervention should any anomalies arise during flight operations.

1) UAV HARDWARE DESIGN

As illustrated in Fig. 4, the hardware components of the drone consist of four platforms. The first platform houses the main battery, power management board, and electronic speed controllers (ESCs), the second platform accommodates the four arms with the engines, the third platform hosts the flight controller, the Jetson TX2, and the GPS, the last platform is dedicated to the main camera, secondary battery, and telemetry radio. The onboard computer responsible

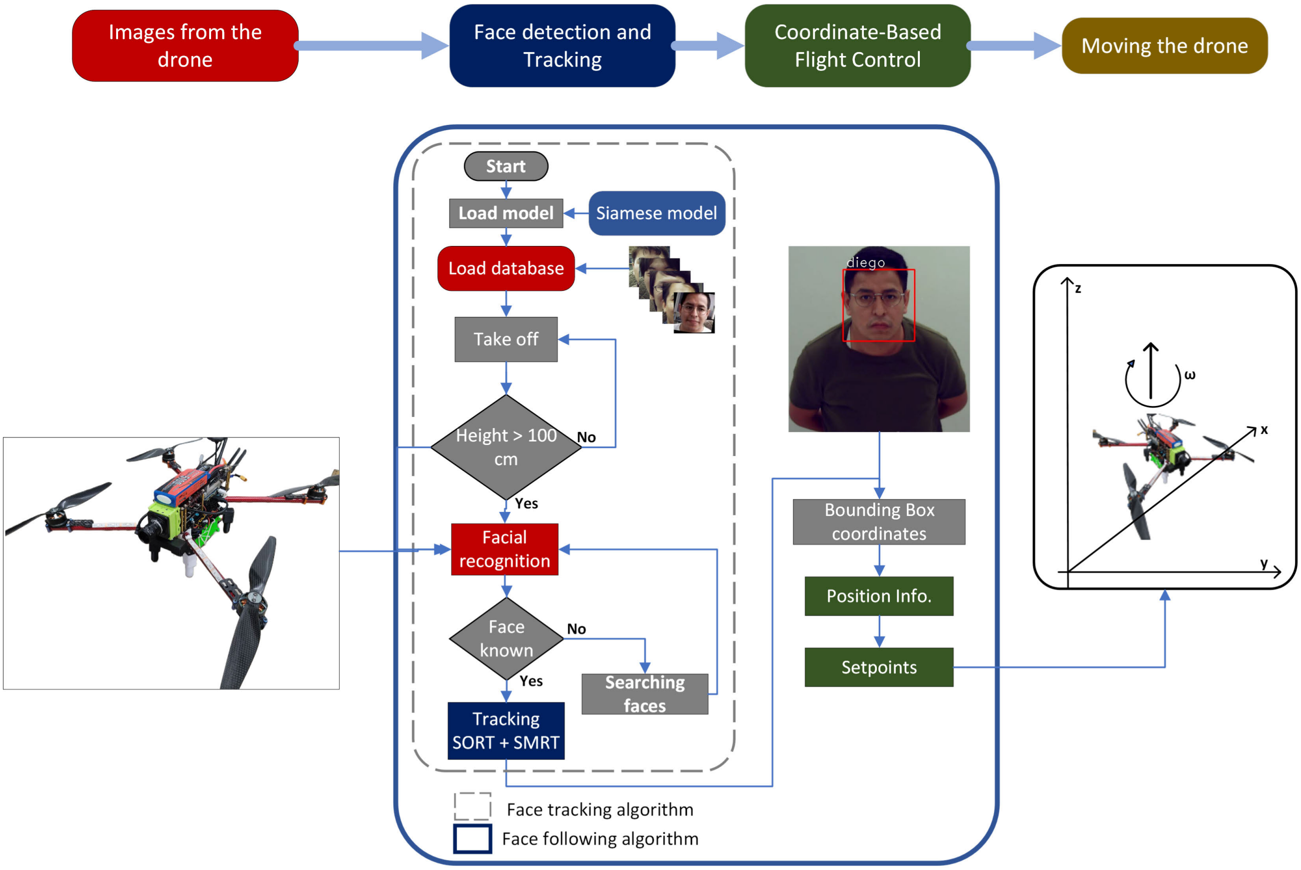

FIGURE 3. Proposed deep learning-based face recognition tracking drone: 1) Images from the camera mounted on the drone are used as input for the Siamese network to recognize faces. 2) The proposed face tracking algorithm tracks the face of the person in front of the drone and returns the setpoints information. 3) Flight control is generated based on the coordinates of the face of the person and using ROS is sent to the flight controller.

for running the computer vision module and sending the commands to the flight controller is the Jetson TX2. Featuring GPU architecture with 256 NVIDIA CUDA cores, a dual-core NVIDIA Denver 2 64-bit CPU, quad-core ARM cortex A57 MPCore, 8GB 128-bit LPDDR4 memory, 32 GB storage eMMC 5.1., the Jetson TX2 is mounted on the Orbitty carrier board. This board provides connectivity options such as USB 3.0, USB 2.0, HDMI, MicroSD, 3.3v UART, I2C, GPIO, and GbE port. The selection of the Jetson TX2 was driven by its power efficiency, and affordability, as not all small computers can run a Siamese network for facial recognition.

For autonomous flight capability, the Pixhawk4 flight controller was selected for its compatibility with the onboard computer and ability to modify position and orientation. While Pixhawk4 utilizes its GPS and IMU sensors for position and orientation data, occasional signal loss is inevitable. To address this issue, the Intel RealSense T265 camera was integrated to provide reliable position and orientation values. Equipped with two fisheye lens sensors, an IMU, and an Intel Movidius Myriad 2 VPU, the T265 camera enables visual SLAM processing on the VPU. In a heightened light intensity environment as well as dark environments, the camera may not be able to capture the visuals. To mitigate issues related to intense light or darkness affecting visibility, the T265 camera is positioned downward towards the landing pad, acting as the drone's eyes. Additionally, video frames are captured by the ELP USB camera mounted on the drone, as presented in Fig. 4. The final design configuration is shown in Fig. 5.

2) ROS ARCHITECTURE

The software implementation is designed to be as autonomous as possible based on the system architecture presented in subsection A. It was necessary to choose a framework to run multiple algorithms that allow the drone to be autonomous as much as possible. The robot operating system (ROS) framework provides the capacity to run each Python script and interact with each other by publishing and subscribing to topics, thus, we can run our nodes to do different tasks at the same time. MAVROS is a bridge between the MAVlink protocol and the ROS framework, MAVlink is a messaging protocol that communicates with drones and between onboard drone components, MAVROS runs in the ROS framework and converts ROS messages into MAVlink messages to be sent to the flight controller.

Fig. 6 depicts the 6 nodes and the flow of messages from one node to another. The facial recognition node is

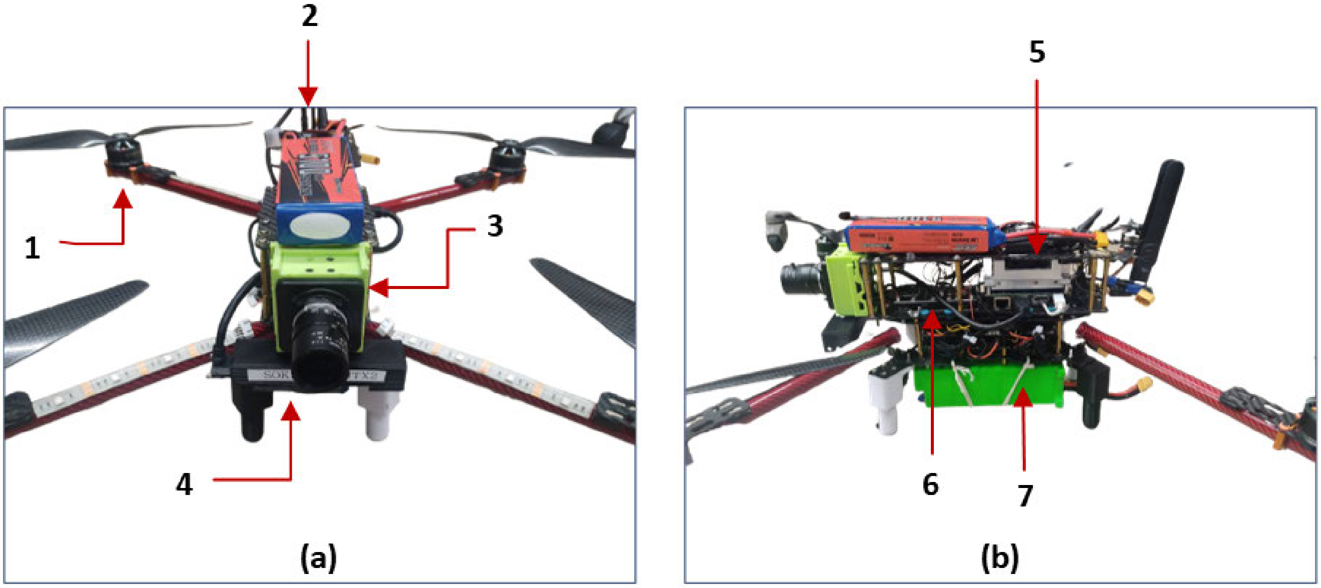

FIGURE 4. An image of our UAV built from scratch with its labeled components. (a) Front view of the drone showing: (1) T-motor MN3510 KV630; (2) Second battery 11.1v 5100mAh; (3) Video camera 1080p; (4) T265 Intel realsense tracking camera. (b) Side view showing: (5) Telemetry radio; (6) Pixhawk 4 flight controller; (7) Main battery 14.8v 5600 mAh.

FIGURE 5. Autonomous drone implementation for facial recognition and tracking.

responsible for performing facial recognition and publishing four topics depending on whether the Siamese model is loaded, if no face is detected and needs to search, or if a face is recognized and needs to halt, publishing the Cartesian coordinates of the recognized face's bounding box. The tracking node subscribes to the facial recognition node and handles person tracking, modifying the drone's position and orientation, and publishing the new drone position and orientation to another node. The searching node subscribes to the facial recognition node and is only activated when no face is detected or recognized, publishing the new drone orientation to rotate on its axis and continue searching for new faces. The takeoff and landing node is responsible for taking off and landing the drone if no face is detected for a few minutes or if it receives a landing instruction via command. The Distributor node receives all position and orientation coordinates from the takeoff and landing, searching, and tracking nodes. After receiving this data, it publishes it to a single pose-type topic to the main node. The main node is the primary node of the quadcopter, which receives the position and orientation it needs to reach and publishes that message to the flight controller using MAVROS.

B. THE OPERATION OF THE DRONE AND THE SYSTEM

The autonomous flight sequence begins with the initialization of the ROS architecture. Subsequently, the drone transitions from manual to offboard mode, autonomously arming the vehicle and initiating takeoff procedures. Once airborne, the ROS nodes responsible for facial recognition, tracking, and search functionalities become active, publishing and subscribing to topics as required.

Subsequently, the USB camera captures frames, which are processed by the computer vision module utilizing the OpenCV library for face detection. This module employs a Haar cascade algorithm to detect faces, resizes the frame to match the input size of the Siamese network (96 × 96), executes the Siamese model, and calculates the Euclidean distance. Following this, the SORT algorithm is executed in conjunction with the SMRT algorithm to enhance recognition and tracking. Upon facial recognition, the tracking node publishes the coordinates of the face's bounding box to initiate face tracking. After completing the tracking experiment, the drone autonomously initiates landing procedures.

Upon detecting a face, the Siamese network identifies the individual and transmits the bounding box coordinates to the motion generator module. The motion generator then updates the local position values based on the detected face's position relative to the camera and forwards the new setpoints to the flight controller. Both the computer vision module and the motion generator operate within the

ROS framework. An external sensor is integrated into the drone system to navigate in GPS-denied environments. The Pixhawk flight controller incorporates various sensors such as GPS, magnetometer, gyroscope, and air pressure sensors to determine the drone's position accurately.

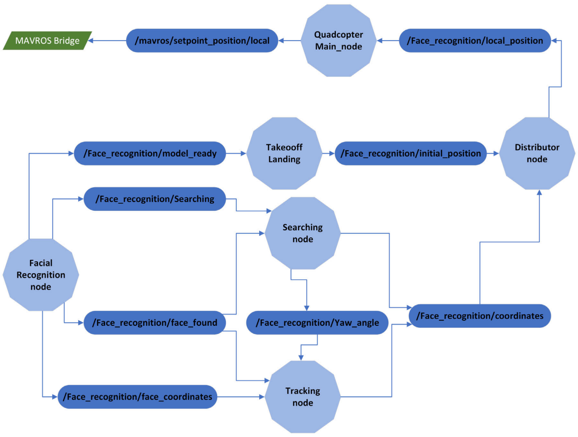
MAVROS Bridge
/mavros/setpoint_position/local
Quadcopter Main_node
/Face_recognition/local_position
/Face_recognition/model_ready
Takeooff Landing
/Face_recognition/initial_position
Distributor node
/Face_recognition/Searching
Searching node
Facial Recognition node
/Face_recognition/face_found
/Face_recognition/Yaw_angle
/Face_recognition/coordinates
/Face_recognition/face_coordinates
Tracking node

FIGURE 6. An illustration of the ROS proposed architecture for the autonomous drone showing the various nodes and the published data to the various interfaces.

C. UAV FLIGHT TIME DURATION

One of the most challenging issues when deploying a UAV in real-world applications is the limited flight time. Since this project is in the research phase, several strategies have been implemented to address this challenge:

- Energy-Efficient Components: The use of the Jetson TX2 and the Pixhawk 4 flight controller are known for their low power consumption, which helps optimize the drone's energy consumption, leading to increased flight time.
- Battery Management: The drone has been tested with two separate batteries, one for the components that make up the onboard computer, and another battery that powers the flight controller along with the motors. This increases the flight time.
- Battery chemical composition: currently, some advanced batteries include graphene. The inclusion of graphene in LiPo batteries improves electrical conductivity, increases energy storage capacity, and accelerates charging and discharging, thereby improving the overall performance, duration, and efficiency of the battery.
- The battery status monitoring function is being implemented to be able to send remote orders to the drone to return to the base station for a battery swap.
- The estimated flight time calculated in this research was between 6 to 8 minutes using a 14.8v and 6500mah battery. The estimated flight time using a 14.8v and 10000mah battery was around 15 minutes.

D. GPS-DENIED NAVIGATION

To achieve flight in GPS-denied environments, the technique of visual-inertial odometry (VIO) is required. VIO is a computer vision technique for estimating the 3D pose and velocity of a vehicle in motion relative to its initial local position. Using VIO, we can determine the position of the drone in 3D space with an Extended Kalman Filter algorithm. Implementing VIO requires the use of RGB cameras and image processing libraries. In our research, we use the Intel RealSense T265 camera, which supports ROS1 using a wrapper. The topics published by the nodes of the ros-T265 package include odometry. The ROS topic we use for odometry is */camera/realsense2_camera/camera/odom/sample*. The necessary parameters to set to use external position information with Extend Kalman Filter(EKF2) are described below:

- EKF2_AID_MASK: Configure the fusion of vision position, vision velocity, vision yaw, and external vision rotation based on the preferred fusion model.
- EKF2_HGT_MODE: Set to Vision to use visual data as the main source for altitude measurement.
- EKF2_EV_DELAY: Adjust to account for the difference between the measurement timestamp and the actual capture time.
- EKF2_EV_POS_X: Specify the location of the vision sensor relative to the vehicle's body frame in the X axis.
- EKF2_EV_POS_Y: Specify the location of the vision sensor relative to the vehicle's body frame in the Y axis.
- EKF2_EV_POS_Z: Specify the location of the vision sensor relative to the vehicle's body frame in the Z axis.

According to the tests conducted, the determined maximum altitude at which the Intel RealSense T265 camera operates correctly is 50 meters. Above this altitude, the camera cannot accurately estimate the altitude. Similarly, in low-light conditions, the camera fails to estimate position and orientation accurately. To address these challenges, a system that integrates both GPS and VIO cameras can be implemented to improve the UAV's position and orientation estimation.

E. FACE-TRACKING METHOD

In this section, we introduce our innovative face-tracking method that extends beyond basic facial recognition and bounding box assignment. When identified individuals attempt to evade the system, the drone becomes a crucial tool, enabling continuous tracking while ensuring a safe distance is maintained, thereby safeguarding the environment.

To achieve this, we impose constraints on the drone's movement trajectories. These constraints are based on the observation that the size of the detected face changes with the distance between the drone and the identified person. Therefore, we use the bounding boxes around recognized faces to gauge the proximity of the user to the drone. Algorithm 1 describes the tracking node of our system, which is implemented in Python and adheres to the specifications described in this section.

Our tracking algorithm evaluates proximity using three main criteria on the detected faces:

1) If the bounding box area of the face exceeds 29,000 px (size of 227 × 128).
2) If the bounding box area of the face is less than 5,000 px (size of 86 × 60).
3) If the bounding box area of the face falls within the range defined by the two previous criteria.

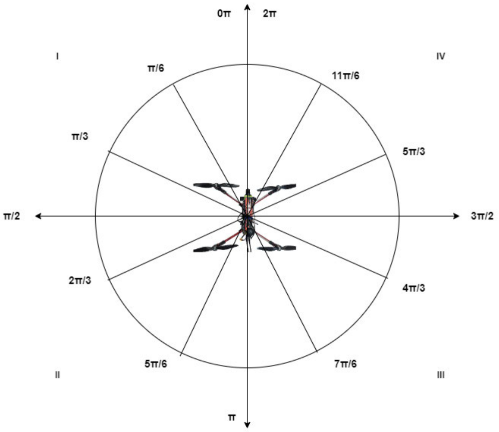

FIGURE 7. Trigonometric circle aligned with the drone coordinates system.

For controlling the drone in an optimal position for recognizing and tracking faces, we set fixed altitude, and the rotation of the drone for only yaw rotation while setting the roll rotation and pitch rotation fixed. The yaw rotation uses the trigonometric circle presented in Fig. 7 and the drone located in the central position.

In our test flights, the movement of the drone is restricted to move backward, forward, and rotate left, and right to follow the target person. The angles are in radians but before sending the set-points to the flight controller, the angles are converted to quaternions.

The frame of the camera is 640 × 480 pixels. To ensure optimal facial recognition, we considered the third criterion to be the best and safest condition for tracking the person. In case the bounding box area of the face covers more than 29000px (size of 227 × 128), it means that a face is too close to the drone and the drone must perform a backward movement not to injure the person in front of the drone. If the bounding box area is less than 5000px (size of 86 × 60), the recognition may not be accurate therefore the drone needs to move closer.

1) BACKWARD MOVEMENT

Given the position of the drone given as $P(x, y)$ at a fixed altitude, a movement from P to new position $O(x, y)$ will occur in an instance where the drone is too close to a subject and move away in the opposite direction. This movement is considered a backward movement. To move the drone backward, we consider the yaw angle, θ of the drone in the 2D plane of (x, y), and which quadrant, θ is located, then we can compute the new position O regarding the quadrant using either of (1) - (4) to determine the new position of the drone as is shown in Fig. 8. Where d is the unit distance of 0.02m moved by the drone in a backward direction repeatedly until an optimal distance between the subject and drone is attained. In (1), the θ is considered for the first quadrant, (2), (3), and (4) are considered for the second, third, and fourth quadrant respectively. An empirical value of 0.09 is used in the equation initializing and setting a secure distance.

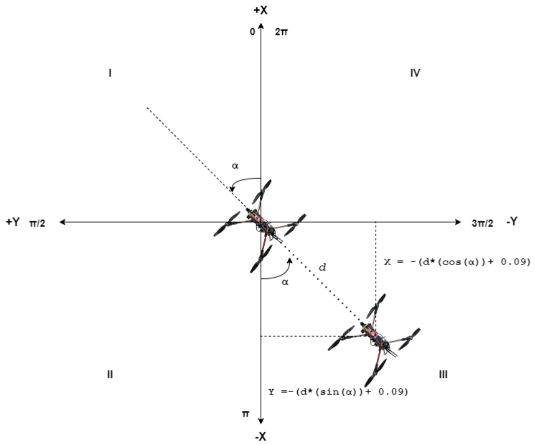

FIGURE 8. First quadrant from 0 to $\pi/2$ with the equation to move backward.

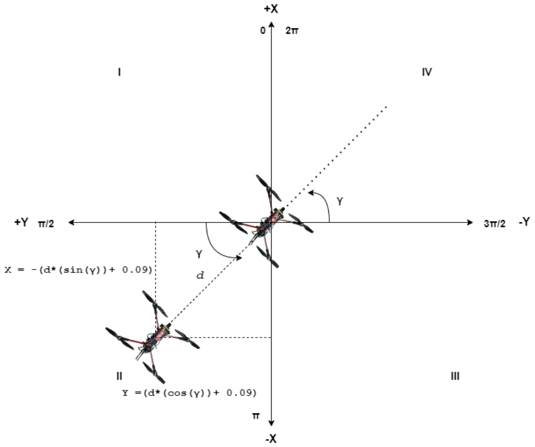

FIGURE 9. Fourth quadrant from $3\pi/2$ to 2π with the equation to move backward.

$$\mathbf{v}^1_{\text{backward}} = \begin{bmatrix} X^1_{\text{backward}} \\ Y^1_{\text{backward}} \end{bmatrix} = \begin{bmatrix} -(d\cos(\theta) + 0.09) \\ -(d\sin(\theta) + 0.09) \end{bmatrix} \tag{1}$$

$$\mathbf{v}^2_{\text{backward}} = \begin{bmatrix} X^2_{\text{backward}} \\ Y^2_{\text{backward}} \end{bmatrix} = \begin{bmatrix} d\sin(\theta) + 0.09 \\ -(d\cos(\theta) + 0.09) \end{bmatrix} \tag{2}$$

$$\mathbf{v}^3_{\text{backward}} = \begin{bmatrix} X^3_{\text{backward}} \\ Y^3_{\text{backward}} \end{bmatrix} = \begin{bmatrix} d\cos(\theta) + 0.09 \\ d\sin(\theta) + 0.09 \end{bmatrix} \tag{3}$$

$$\mathbf{v}^4_{\text{backward}} = \begin{bmatrix} X^4_{\text{backward}} \\ Y^4_{\text{backward}} \end{bmatrix} = \begin{bmatrix} -(d\sin(\theta) + 0.09) \\ d\cos(\theta) + 0.09 \end{bmatrix} \tag{4}$$

2) FORWARD MOVEMENT

The forward movement occurs when the bounding box area is less than 5000px means that a face is too far from the

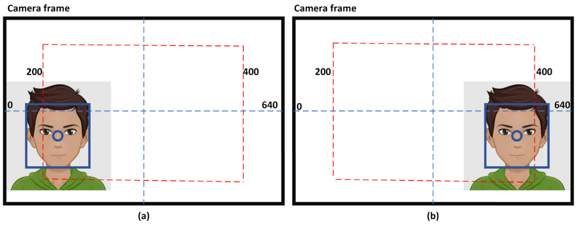

FIGURE 10. Conditions to rotate to the right or left. (a) If the center of the face is near the left edge of the red rectangle the drone will rotate 0.02 rad to the left. (b) If the center of the face is near the right edge of the red rectangle the drone will rotate 0.02 rad to the right.

drone and the drone must perform the forward movement to be able to recognize the person in front of the drone. To move the drone forward we need to identify in which quadrant the drone is located and it is only necessary to multiply by -1 (1) to (4), as a result, we have the equations to move forward as shown from (5) to (8). The variable distance d has an initial value of 0.1 meters which increases at a rate of 0.02 meters until the bounding box area is more than 5000px.

$$\mathbf{v}^1_{\text{forward}} = \begin{bmatrix} \mathrm{X}^1_{\text{forward}} \\ \mathrm{Y}^1_{\text{forward}} \end{bmatrix} = \begin{bmatrix} d\cos(\alpha) + 0.09 \\ d\sin(\alpha) + 0.09 \end{bmatrix} \tag{5}$$

$$\mathbf{v}^2_{\text{forward}} = \begin{bmatrix} \mathrm{X}^2_{\text{forward}} \\ \mathrm{Y}^2_{\text{forward}} \end{bmatrix} = \begin{bmatrix} -(d\sin(\beta) + 0.09) \\ d\cos(\beta) + 0.09 \end{bmatrix}) \tag{6}$$

$$\mathbf{v}^3_{\text{forward}} = \begin{bmatrix} \mathrm{X}^3_{\text{forward}} \\ \mathrm{Y}^3_{\text{forward}} \end{bmatrix} = \begin{bmatrix} -(d\cos(\theta) + 0.09) \\ -(d\sin(\theta) + 0.09) \end{bmatrix} \tag{7}$$

$$\mathbf{v}^4_{\text{forward}} = \begin{bmatrix} \mathrm{X}^4_{\text{forward}} \\ \mathrm{Y}^4_{\text{forward}} \end{bmatrix} = \begin{bmatrix} d\sin(\gamma) + 0.09 \\ -(d\cos(\gamma) + 0.09) \end{bmatrix} \tag{8}$$

3) HOVERING MOVEMENT

In case the bounding box area is between 5001px to 29000px means it is a safe distance between the drone and the person. Then, if the person moves to the right the drone will rotate to the right, if the person moves to the left the drone moves to the left. This action is done by modifying the yaw angle with a rate of 0.02 *rad*. The frame of the camera is 640 × 480 pixels, which means the horizontal axis is from 0px to 640px. In case the face is located near the left edge of the camera frame, which means less than 200px in the horizontal axis, the drone will rotate to the left, and in case the face is located near the right edge of the camera frame, which means more than 400px in the horizontal axis, the drone will rotate to the right as Fig. 10.

F. FACE DETECTION

To accomplish the facial recognition system, we are using and combining four topics, the haar cascade classifier, siamese network model, Inception V2 architecture, and FaceNet weights. OpenCV provides us with an easy way to detect faces by using a haar cascade classifier. OpenCV provides a training method or pre-trained model that can be loaded from the OpenCV installation folder [54]. This method of facial detection can be improved by modifying some thresholds so it can detect better a human face instead of some random image. The whole documentation about how OpenCV works and how to deploy it can be found on the OpenCV website [55]. Fig. 11 shows face detection running on Ubuntu.

Algorithm 1 Real-Time Face Tracking Control with a ROS

```
Import necessary libraries and packages
    rospy, ast, std_msgs (String, Float64)
    geometry_msgs (Point, Pose)
    gazebo_msgs (ModelStates)
    nav_msgs (Odometry)
    time (sleep), re
    tf.transformations (quaternion_from_euler)
    numpy as np, math
Define methods for callbacks:
    object_detection - Extract bbox and set flag3
    coordinate_callback - Update c1 and area
    face_found_callback - Set flag
    face_match_callback - Set flag2
    orientation_callback - Update orientation values
    kill_callback - Set kill_program
    orientation_t265_callback - Update pose orientation and position
Define movement methods: right, left, hold_position, backward, forward
Define method new_quaternion - Update orientation values
Define class data_processing
Define class constructor __init__
    Initialize constants and flags
    Initialize pose and orientation values
    Define ROS subscribers for face recognition and pose data
    Define ROS publishers for pose and yaw angle feedback
Define main control loop
    Check for kill program signal
    If face detected:
        If face too close:
            Log message and update pose to move backward
            Publish updated pose
        If face too far:
            Log message and update pose to move forward
            Publish updated pose
        If face within safety area:
            Log message and hold position.
            Adjust yaw angle and pose based on the bounding box center, then move to the right or left.
            Publish updated pose and yaw angle feedback
Define main function
    Initialize ROS node
    Create instance of data_processing
    Keep node running with rospy.spin()
```

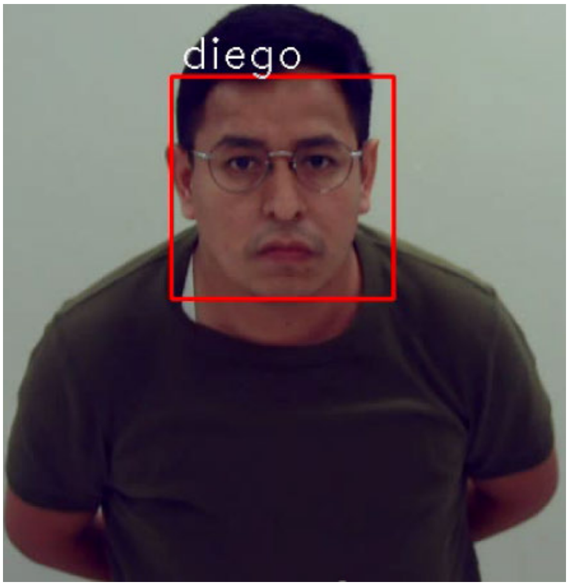

FIGURE 11. An image showing the implementation of face detection using OpenCV haar cascade classifier.

G. SIAMESE NETWORK

Convential facial recognition systems work in four main steps: detection, alignment, representation, and classification. A thousand face images are trained and then the final model can classify who the person is. This method works well but it has problems as if we want to add a new person to the database we must train the network again. Fig. 12 shows a Siamese model representation, it has two inputs that are the images we want to compare. Each image is passed through a convolution neural network to determine the 128-dimensional vector of one input. Thus, we have two 128-dimensional vectors as an output. Hence, we compute the Euclidean distance between the two outputs. This method of recognizing faces is one of the best options since it only requires a few images as inputs.

First, we collect data by taking 95 photos of 5 subjects which in total is 475 photos as Fig. 13 shows. This database is going to be one input of the Siamese network. Second, we implement a Python script to capture video and send the frame video as a second input of the Siamese network. It is called the Siamese network because there are two inputs for the same DNN and has one output which is the Euclidean distance which determines if the face in front of the drone matches the faces in the database.

We have collected in total of 475 photos of our classmates, these photos were taken from the camera installed in the drone before flying so we can obtain better face recognition accuracy. Fig. 14 shows us the height of testing and the test environment.

To adapt the InceptionV2 architecture to our specific facial recognition needs, we made several key modifications to the original structure. The following summarizes the implemented changes:

- **Input Size:** The original InceptionV2 architecture uses an input size of (299 × 299 × 3). In our version, we reduced the input size to (96 × 96 × 3) to better match the typical dimensions of facial images.
- **Initial Layers:** We retained the initial structure of convolution followed by batch normalization and ReLU activation. Specifically, we employed a `Conv2D` layer with 64 filters, a kernel size of (7 × 7), stride 2, and 'same' padding, followed by a `BatchNormalization` layer and a `ReLU` activation.
- **Initial Pooling:** Similar to InceptionV2, we used a `MaxPooling2D` layer with a pool size of (3 × 3), stride 2, and 'same' padding to reduce the dimensionality of the extracted features.
- **Inception Blocks:** We implemented several Inception blocks, although with specific configurations tailored to our task. These blocks include convolutions of different sizes (e.g., (1 × 1), (3 × 3), (5 × 5)) and pooling layers, combined in a way that maintains a balance between spatial feature extraction and computational efficiency.

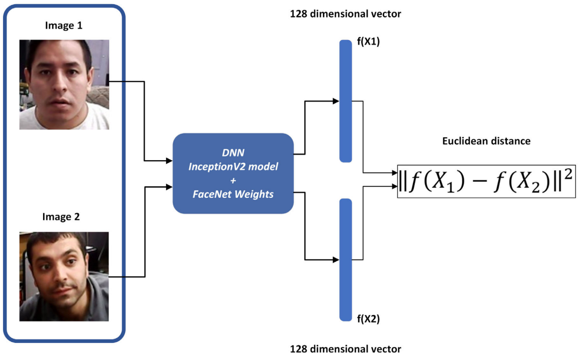

FIGURE 12. Siamese network representation used in this research. If f(X1) and f(X2) are the encoding vectors of the same person then the Euclidean distance must be small. If f(X1) and f(X2) are the encoding vectors of different persons then the Euclidean distance must be large.

FIGURE 13. Database from our classmates to be recognized by the drone.

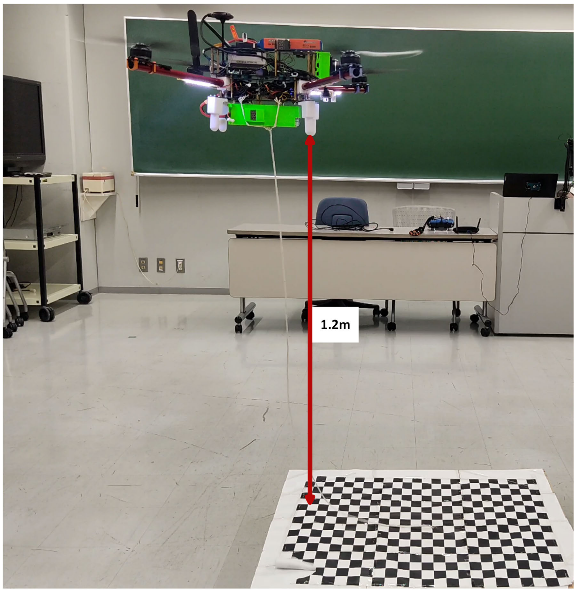

FIGURE 14. Height of 1.2 meters from the landing pad. The highest flight height will be between 1 meter and 1.5 meters high, which makes the total altitude from the ground to the camera 1.80 meters.

- **Reduction Layers:** Although the original architecture uses (1 × 1) convolutions to reduce dimensionality before applying larger convolutions, our design makes limited use of this technique due to the smaller input dimensions and the specificity of the facial recognition task.
- **Final Layers:** Unlike InceptionV2, which uses final dense layers and softmax for classification, our version employs a dense layer followed by `L2` normalization to produce embeddings. These embeddings are essential for verification and recognition tasks in a Siamese network setup.
- **Parameters and Complexity:** Our version has a total of 3,743,280 parameters, optimized to work efficiently with smaller facial images, maintaining an adequate balance between accuracy and computational efficiency.

These modifications allow the adapted network to retain the structural advantages of the Inception architecture while being specifically tailored to the needs of our facial recognition task in a Siamese network.

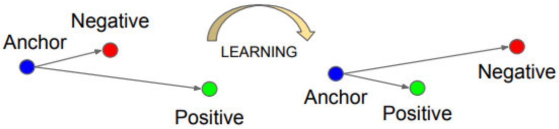

FIGURE 15. Triplet Loss representation. Maximizes the distance between the anchor and the negative and minimizes the distance between the anchor and the positive [56].

H. FACIAL RECOGNITION

As we mentioned before, we are using a Siamese network to obtain good accuracy in recognizing people. We are not training a network from scratch to recognize faces because it will take a lot of time and computation, instead, we do transfer-learning using the weights of the FaceNet model [56] that was trained with thousands of images from the Labeled Faces in the Wild database [57]. FaceNet weight can be downloaded from GitHub since it is an open source [58].

To load the FaceNet weights we need to implement a network architecture using TensorFlow-Keras, thus, we implement the network architecture following the inception model that has been published and can be found on GitHub. This network architecture follows the Inception model which was tested with image classification and detection. The inception architecture we have used in this research can be found in [58]. We have implemented an Inception network with three inputs: anchor, negative, and positive, and a single output of a 128-dimensional vector. After 100 epochs, a loss of 0.0017 was achieved on the training data and 0.0388 on the validation data. Figures 23 and 24 display the loss results for the training and validation data, respectively. As shown, the use of weights from a pre-trained network facilitates faster convergence of the loss, which indicates that the model achieves superior accuracy in recognizing the faces within our database. The architecture is implemented using Keras and TensorFlow. Besides, this implementation has the triplet loss as a loss function. The Triplet loss equation is shown in (9), where A is the anchor which means the database, P is positive which means random images of the same person in the database, and N is negative which means random images from different people not included in the database; triplet loss has these three parameters, anchor, positives, and negatives. The Anchor and the positives must be the encoding of the same image person while negatives must be the encoding of random image faces as Fig. 15 shows.

$$J = \sum_{i=1}^{m}[\underbrace{||\mathbf{f}(\mathbf{A^i}) - \mathbf{f}(\mathbf{P^i})||_2^2} - \underbrace{||\mathbf{f}(\mathbf{A^i}) - \mathbf{f}(\mathbf{N^i})||_2^2} + \alpha] \tag{9}$$

The goal of this research was to detect, recognize, and track the target person within the database. As a video input sensor, we use an ELP USB camera connected to the Jetson TX2, Jetson TX2 is an onboard computer installed in the drone. Currently, there are various deep learning algorithms to recognize faces such as vgg19 [10], we have tested vgg19 in an Ubuntu computer and the performance was enough good with a 94% accuracy but since vgg19 is a heavy model, Jetson TX2 cannot run a siamese network using vgg19 as the main architecture, Thus, we have chosen the inception v2 model plus the weights of the FaceNet unified system [56]. To summarize, our customized Inception network has been modified to include three inputs: anchor, positive, and negative. We then perform transfer learning using the weights from FaceNet and our own dataset. After training, we obtain a model in TensorFlow-Keras format. This model, which now has updated weights, is used in the Siamese network. The Siamese network is composed of the base network, which is our customized Inception network with the trained weights.

The operation of the Siamese network is explained in the subsection on the Siamese network.

OpenCV has a haar cascade classifier to detect faces, we have implemented a Python code to run this classifier and we have modified the minimum neighbor's parameter to obtain better facial detection. The model plus the weights are loaded beforehand and then we create a database dictionary where we pass the whole database through the new model and obtain the 128-dimensional vector of each picture within the database. After that, we capture a photo within the bounding box generated by the haar cascade classifier, and each photo taken by the camera is passed through the model to obtain the 128-dimensional vector. Finally, we compute the Euclidean distance between the database encoding vector and the photo encoding vector taken by the camera. Then, the Python script selects the minimum Euclidean distance between the encoding vector from the photo against the encoding vectors from the database. Thus, if the minimum distance is less than 0.66 means that the subject in front of the camera matches someone in the database, in case the minimum distance is more than 0.66 means the subject in front of the camera is not in the database; the minimum distance can be changed to increase the number of subjects recognized. Real-time facial recognition is running in the ROS framework, in case the drone detects a face it will stop for a few seconds to catch the face better, so it can crop and send the face image to the Siamese network, if the face is not in the database the drone will rotate a few radians ignoring the face in front of the drone, otherwise will follow the face.

I. TRAINING DATASET

All the subjects signed a consent form agreeing to the use of their facial data for this research, but not for it to be made into a public dataset.

To perform transfer learning, we utilized the pre-trained weights of the FaceNet model [56]. Our dataset is divided into three subsets: Anchor, Positive, and Negative. The Anchor subset contains 5 folders, each corresponding to one of the 5 subjects. Each folder includes 48 images, resulting in a total of 240 images for the Anchor subset. Similarly, the Positive subset comprises the same 5 subjects with 48 images per folder, totaling 240 images. For the Negative subset, we used the Labeled Faces in the Wild (LFW) dataset [57], which originally contains 1,473 images. We randomly selected 240 images from the LFW dataset to ensure a balanced training set.

The images were resized to 96 × 96 pixels to match the input size required by the Siamese network. The Anchor and Positive images were captured in controlled indoor environments to maintain consistent lighting and background conditions, enhancing the uniformity of the dataset. Each subject was photographed from various angles and with different expressions to create a comprehensive and varied dataset. This meticulous preparation ensures that the model is robust and can generalize well to new data.

J. SIMPLE MATCHING REAL-TIME TRACKING - SMRT

One of the primary goals of this research was to enable the drone to follow a specific person in front of it, provided that the individual is recognized within the database. To achieve this objective, we implemented the Simple Online and Realtime Tracking (SORT) algorithm, which is noted for its ease of use [59]. While this algorithm is capable of tracking various objects across frames, our application required tracking individuals by name rather than by ID. For example, in frame n, the SORT algorithm may track an individual as ID 2, and the facial recognition system might identify this person as Juan. However, in the subsequent frame n+1, the SORT algorithm could continue tracking the same ID 2, but the facial recognition system might label the person differently or fail to recognize the individual, even though it is still Juan. To address this issue, we developed an algorithm that matches the tracked ID with the facial recognition name, as detailed in Algorithm 2.

The functioning of the entire facial recognition system, including Algorithm 2, is described in Algorithm 3. First, the "triplet loss" function is defined and a custom function for stacking embeddings is registered. Subsequently, the pre-trained model "siamesemodelv2.keras" is loaded and compiled using the Adam optimizer and the "triplet loss" function. A database of facial embeddings is created from images stored in the specified directory. Next, video capture is initialized, and video output recording is configured. In a continuous loop, the system processes each video frame, detects faces using a Haar classifier, and for each detected face, performs facial recognition by comparing it with the database of embeddings. The SMRT algorithm is utilized to enable the drone to follow a specific person, provided that the individual is recognized within the database. Finally, the recognition result is displayed in real-time video and the processed video is stored until execution is terminated.

If the Euclidean distance between the camera image and the database is greater than 0.53, then the person is considered unknown. However, if the person was recognized in the previous frame, we need to ensure the consistency of the result. Let's assume that the person in front of the camera is not recognized, so the average distance value would be greater than 0.53. This would cause the value in dictionary A to be 0. If the person's ID is 1, since SORT assigns this ID, then the value of B in the dictionary would be ID 1, and the key in dictionary B would be 'Unknown'. In the next frame, with the same unknown person, if the Euclidean distance is less than 0.53, then the value of B in the dictionary, which is 1, is compared with the ID generated by SORT, which would also be 1 (since we are only detecting one person and ignoring the others). Thus, the result would be the same as the previous frame: the value of B in the dictionary would be ID 1, and the key would be 'Unknown'. On the other hand, if the person is recognized and the average distance is less than 0.53, the value of B in the dictionary, which is 1, is compared with the new ID generated by SORT, which

Algorithm 2 Simple Matching Real-Time Tracking

```
Require: $avg$ - Average Euclidian distance between image
    path encoding and the encodings from the database.
Require: $A.value, B.value$ - Dictionaries to store the recognized values for the current and previous frame.
Require: $id$ - tracking ID from SORT algorithm.
Ensure: $A.key, B.key$ - final determined identity keys.
Ensure: $A.value, B.value$ - final determined identity values.
if $avg > 0.53$ then
    if $A.value == id$ then
        $A.key \leftarrow$ Identity
        $A.value \leftarrow id_A$
    else
        $B.key \leftarrow$ Unknown
        $B.value \leftarrow id_B$
    end if
else
    if $B.value == id$ then
        $B.key \leftarrow$ Unknown
        $B.value \leftarrow id_B$
    else
        $A.key \leftarrow$ Identity
        $A.value \leftarrow id_A$
    end if
end if
```

would be 2 (since it is a new person). This would result in the value of A in the dictionary being 2, and the key in dictionary A being the identity given by the facial recognition system. Similarly, in the following frame with the same person, but when the average distance is greater than 0.53, the value of A in the dictionary would be 2 and would match the ID generated by SORT, which is 2. This would result in the value of A in the dictionary being 2, and the key being the identity given by the facial recognition system. This ensures that even though the recognized person moves and the average distance changes, the drone can follow the known person in front of it.

IV. EXPERIMENTS

In this section, we detail the experiment conducted to evaluate the recognition and tracking of faces in three different environments during real flight test mode. These environments include flying indoors, characterized as a GPS-denied environment; flying outdoors, distinguished as a GPS-enabled environment; and no-flying mode. The following subsection describes the environment setup and the system setup.

A. ENVIRONMENT SETUP

Three environments are used in this research to analyze the drone behavior, response time, and accuracy of the facial recognition system. Fig. 16 shows the first environment, the drone is located on the desk. This environment is just to obtain the accuracy of the facial recognition system in the ideal scenario. The ideal scenario refers to not having vibration caused by the drone or some other disturbance that can affect the facial recognition system.

Algorithm 3 Facial Recognition and Tracking Using Siamese Model With SMRT Algorithm

```
Require: $avg\_val$ - Average Euclidean distance between
    image path encoding and the encodings from the database.
Require: $A\_dict, B\_dict$ - Dictionaries to store the recognized values for the current and previous frame.
Require: $id\_N$ - Tracking ID from SORT algorithm.
Ensure: $A\_dict.key, B\_dict.key$ - Final determined identity keys.
Ensure: $A\_dict.value, B\_dict.value$ - Final determined identity values.
Initialize video capture and output configuration.
Initialize face detector and tracker.
Initialize identity tracking variables.
while True do
    Read a frame from the video capture.
    if frame is read successfully then
        Detect faces in the grayscale frame using the Haar classifier.
        Initialize an empty list for detections.
        if faces are detected then
            for each detected face do
                Extract and resize the region of interest (ROI).
                Perform facial recognition using the model to obtain the minimum distance and identity.
                Save the minimum distance in a CSV file.
                Round the $avg\_val$.
            end for
            Update the tracker with detections.
            for each box in the updated tracker do
                Extract coordinates and ID.
                Call $SMRT\_Algorithm1(avg_{v}al, A.value, B.value)$ with the current parameters to update the tracking state.
            end for
        else
            Display "No faces detected" on the frame.
        end if
        if frame is read successfully then
            Write the frame to the video output.
        end if
        Display the frame.
    end if
    if exit condition is met (e.g., 'q' key is pressed) then
        Break the loop.
    end if
end while
Release video capture and output resources.
Destroy all windows.
```

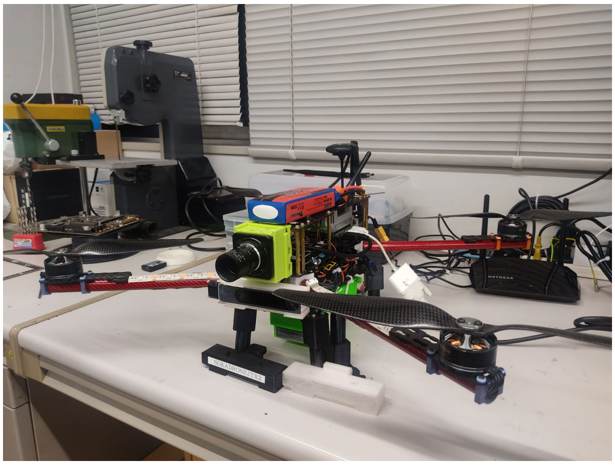

FIGURE 16. The first environment: the drone is seen in a position with no disturbance that would affect the face recognition system.

The second environment is set up as shown in Fig. 17. In this environment, the GPS signal does not work because it is a closed environment, and we need to rely on visual odometry from the T265 camera. The searching-tracking mode is the complete test we have done, in this mode, the drone rotates searching for faces and then must stop when a face is in front of the drone and track the face only if the face is within the database otherwise, it must rotate to look for other faces, as well as the drone, moves backward and forward to maintain a safety distance from recognized faces.

Fig. 18 shows the third test environment, in this experiment, we test the performance of the system in situations where natural lights can affect the system. In all three environments set up, the facial recognition system experiment was performed with 5 participants. During the test in the first environment, the participants stood in front of the camera of the drone. In the setups for the second and third environments, the participants positioned themselves in front of the drone while it was flying. Subsequently, the drone operator, in this case the author of this research, gave instructions to the participant to walk towards the right until reaching point B (explained in section V-D), and then proceed to point C. The participant walked while looking directly at the drone camera. For the third environment, GPS was not used; instead, we relied solely on the T265 sensor. However, in the initial experiments, we observed that the sensor struggled to obtain its position due to the sandy terrain. To address this, we added markers of various colors and shapes so that the T265 camera could use the patterns on the markers as reference points.

B. SYSTEM SETUP

The autonomous drone requires a specific setup before takeoff. Since we are using an onboard computer, we need to connect it to the flight controller and modify certain parameters, as shown in Table 3. Additionally, we must set the parameter MAV_1_FORWARD to 1 in order to observe the MAVLink messages in the QGroundControl software on the local PC. Table 1 displays the pin-out of the Jetson TX2 carrier board.

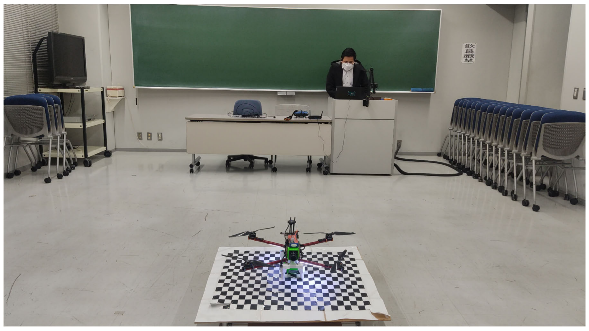

FIGURE 17. The second environment: the drone is seen in a position before attaining the fixed altitude in a closed area where GPS does not work.

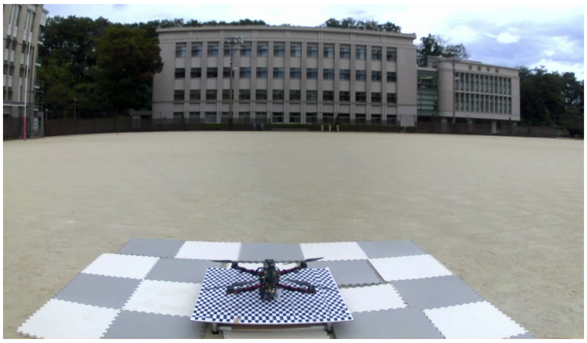

FIGURE 18. The third environment: the drone is seen in a position before attaining the fixed altitude in an open area where GPS does work.

Table 1 shows the pin map of the extension connector, we are using the UART1 port to connect the carrier board with the pixhawk4. The flight controller Pixhawk is connected to the jetson TX2 via UART and Table 2 shows the connection between the pins of the jetson TX2 and the Pixhawk. A complete guide on how to connect the Pixhawk and the Jetson TX2 devkit can be found in [60]. The last step before flying the drone is to modify a few parameters in the Pixhawk firmware. Table 3 shows the parameters to be modified and its values. After this setup, the drone is ready to fly in the onboard mode.

V. EXPERIMENTAL RESULTS

This section presents the findings of our research, organized into three subsections corresponding to the first, second, and third experimental environments. Besides, we show the results of the SMRT algorithm, and without using the algorithm in each subsection.

A. FIRST ENVIRONMENT

To replicate the ideal conditions for facial recognition without interference from drone vibrations, we positioned the drone on a desk and executed the facial recognition system. In this setup, we assessed the performance of the facial recognition system.

TABLE 1. Pinout of the expansion IO connector.

Pin	Description	Pin	Description
1	+3.3V output	2	+5V output
3	UART0 TX	4	UART0 RX
5	UART1 TX	6	UART1 RX
7	GPIO-0	8	GPIO-1
9	GPIO-2	10	GPIO-3
11	I2C CLK	12	I2C SDA
13	RECOVERY	14	RTC BAT INPUT
15	RESET	16	GND
17	POWER BUTTON	18	GND
19	GND	20	GND

Description of the expansion pins of the Orbitty carrier board for communication between the flight controller and the Jetson TX2, where UART0 is used.

TABLE 2. Pixhawk telemetry to Jetson TX2 UART0 pin mapping.

Pin	Pixhawk Telem2	Pin	Jetson TX2 Uart0
1	Not connected	6	CTS
2	Tx	4	UART0 RX
3	Rx	5	UART0 TX
4	Not connected	3	Not connected
5	Not connected	2	RTS
6	GND	1	GND

Connection between the jetson tx2 uart1 and the pixhawk telemetry port.

1) SIAMESE FACIAL RECOGNITION SYSTEM

The drone captured a total of 170 images of each person, totaling 850 images. The model has been trained with these images using transfer learning techniques, after running the model in real-time, it achieves an overall accuracy of approximately 98.21%, meaning the model can recognize the person in front of the camera of the drone with a 98.21% accuracy, This accuracy was calculated as the division between the total number of correct predictions and the total number of captured images. Table 4 shows us the results for each person. The Siamese network model proves to be a good tool for face recognition, offering acceptable precision, recall, and F1-Score results.

2) SIAMESE FACIAL RECOGNITION SYSTEM USING SMRT ALGORITHM

After running the model in real-time with the Siamese network combined with the SMRT algorithm, the model can recognize the person in front of the camera attached to the drone with a 99.62% accuracy. This demonstrates an improvement over the initial expectations, highlighting the effectiveness of the SMRT algorithm in enhancing the recognition of the model capabilities. Table 5 shows us the results for each person. The combination of the Siamese network model and the SMRT algorithm proves to be a powerful tool for face recognition, offering robust precision, recall, and F1-Score results across different individuals. The high overall accuracy and the detailed performance metrics for each individual, underscore the ability of the model to identify persons in diverse conditions reliably. From the confusion matrix in Fig. 19, we can estimate the accuracy, precision, recall, and F1-score of the Siamese facial recognition system using the SMRT algorithm and compare them against the values of the Siamese facial recognition without using the novel algorithm as shown in Table 6.

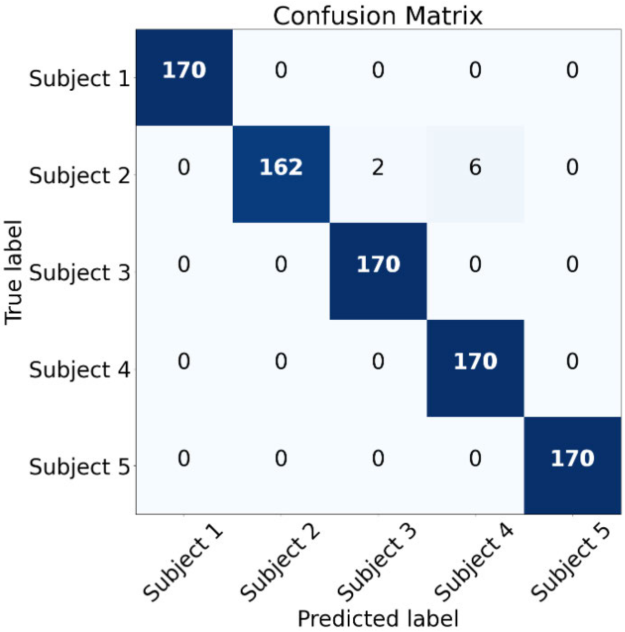

FIGURE 19. Confusion matrix. Siamese facial recognition system using SMRT algorithm in the first environment.

B. SECOND ENVIRONMENT

The second environment is designed to evaluate the facial recognition system while the drone is flying. In this way, the vibration of the drone and the task of tracking the face of the person can disrupt the facial recognition system. The drone is located in a classroom where GPS signal is not available, and automatic takeoff is initiated followed by the execution of the facial recognition system.

1) SIAMESE FACIAL RECOGNITION SYSTEM

Similarly to the previous experiment, a total of 200 images were captured for each person, resulting in 1000 images in total. The model was trained with these images using transfer learning techniques. Upon executing the model in real time, it achieved an overall accuracy of approximately 97.72%. Table 7 presents the individual results for each person. The Siamese network model proves to be a valuable tool for face recognition, delivering acceptable precision, recall, and F1-Score outcomes.

2) SIAMESE FACIAL RECOGNITION SYSTEM USING SMRT ALGORITHM

After running the model in real-time with the Siamese network combined with the SMRT algorithm, it achieves an overall accuracy of approximately 99.32%. This demonstrates an improvement over the initial expectations, highlighting the effectiveness of the SMRT algorithm in enhancing the model's recognition capabilities. Table 8 shows us the results for each person. The combination of the

TABLE 3. Mavlink parameter settings.

Pin	Pixhawk Telem2	Pin
MAV 1 CONFIG	TELEM 2	Serial configuration for MAVLink (instance 1)
MAV 1 FORWARD	Disabled	Enable Mavlink message forwarding for instance 1
MAV 1 MODE	Onboard	MAVlink mode for instance 1
MAV 1 RATE	80000 B/s	Maximum MAVlink sending rate for instance 1
SER TEL2 BAUD	921600 8N1	Baudrate for the TELEM 2 Serial port

QGround Control Mavlink settings. MAV is the abbreviation of MAVlink protocol, and it must be modified in the flight controller.

TABLE 4. Performance metrics for the siamese model.

Person	Acc (%)	Precision (%)	Recall (%)	F1-Score (%)
Subject 1	99.64	98.26	100.0	99.12
Subject 2	97.88	94.18	95.29	94.73
Subject 3	98.47	93.85	98.82	96.27
Subject 4	95.88	95.30	83.52	89.02
Subject 5	99.17	96.04	100.0	97.98

TABLE 5. Performance metrics for the siamese model + SMRT algorithm.

Person	Acc (%)	Precision (%)	Recall (%)	F1-Score (%)
Subject 1	100.0	100.0	100.0	100.0
Subject 2	99.05	100.0	95.29	97.59
Subject 3	99.76	98.84	100.0	99.41
Subject 4	99.29	96.59	100.0	98.26
Subject 5	100.0	100.0	100.0	100.0

TABLE 7. Performance metrics for the siamese model.

Person	Acc (%)	Precision (%)	Recall (%)	F1-Score (%)
Subject 1	(100.0	(100.0	(100.0	(100.0
Subject 2	96.60	100.0	83.01	90.71
Subject 3	96.89	86.89	99.51	92.77
Subject 4	96.21	90.90	90.01	90.45
Subject 5	98.91	95.65	99.01	97.29

TABLE 8. Performance metrics for the siamese model + SMRT algorithm.

Person	Acc (%)	Precision (%)	Recall (%)	F1-Score (%)
Subject 1	100.0	100.0	100.0	100.0
Subject 2	98.72	100.0	93.51	96.64
Subject 3	99.91	100.0	99.51	99.74
Subject 4	98.41	93.81	98.51	96.09
Subject 5	99.61	98.03	100.0	99.01

Siamese network model and the SMRT algorithm proves to be a powerful tool for face recognition, offering robust precision, recall, and F1-Score results across different individuals. The high overall accuracy and the detailed performance metrics for each individual underscore the ability of the model to identify persons in diverse conditions reliably. From the confusion matrix in Fig. 20, we can estimate the accuracy, precision, recall, and F1-score of the Siamese facial recognition system using the SMRT algorithm and compare them against the values of the Siamese facial recognition without using the novel algorithm as shown in Table 9, in the siamese facial recognition system, precision is higher than accuracy, which in turn indicates that the model is good at identifying positive cases for one or more classes but not as good at correctly classifying certain specific classes.

C. THIRD ENVIRONMENT

The third environment is located in an open area with natural lighting. The purpose of this experiment was to evaluate the performance of the facial recognition system under natural lighting and real weather conditions. On the day of the experiment, the sky was partly cloudy with light rain. We chose to conduct this experiment with only two subjects: the first subject being the author of this investigation, and the second subject being the laboratory research assistant. The experiment was limited to two individuals for safety reasons, as it was conducted in a real-world scenario with winds that could potentially cause the drone to move towards a person.

1) SIAMESE FACIAL RECOGNITION SYSTEM

The drone captured a total of 130 images of each person, totaling 260 images. The model has not been trained with these images, so a low accuracy was expected. After running the model in real-time, it achieves an overall accuracy of approximately 90.0%. Table 10 shows us the results for each person. The Siamese network model proves to be a good tool for face recognition, offering acceptable precision, recall, and F1-Score results. However, the variability in performance among different individuals highlights the need to train the model with our data.

2) SIAMESE FACIAL RECOGNITION SYSTEM USING SMRT ALGORITHM

After running the model in real-time with the Siamese network combined with the SMRT algorithm, the model does not show the expected results, as it can recognize the person in front of the drone's camera with an 81.15% accuracy. Table 11 shows us the results for each person. The combination of the Siamese network model and the SMRT algorithm proves to be a powerful tool for face recognition,

TABLE 6. Performance metrics for our siamese model against siamese model+SMRT algorithm in the first environment.

Metrics	Siamese model	Siamese model + SMRT
Accuracy (%)	98.21	99.62
Precision (%)	95.53	99.08
Recall (%)	95.52	99.05
F1-Score (%)	95.53	99.07

TABLE 9. Performance metrics for our siamese model against siamese model+SMRT algorithm in the second environment.

Metrics	Siamese model	Siamese model + SMRT
Accuracy (%)	97.72	99.32
Precision (%)	94.69	98.37
Recall (%)	94.3	98.31
F1-Score (%)	94.49	98.33

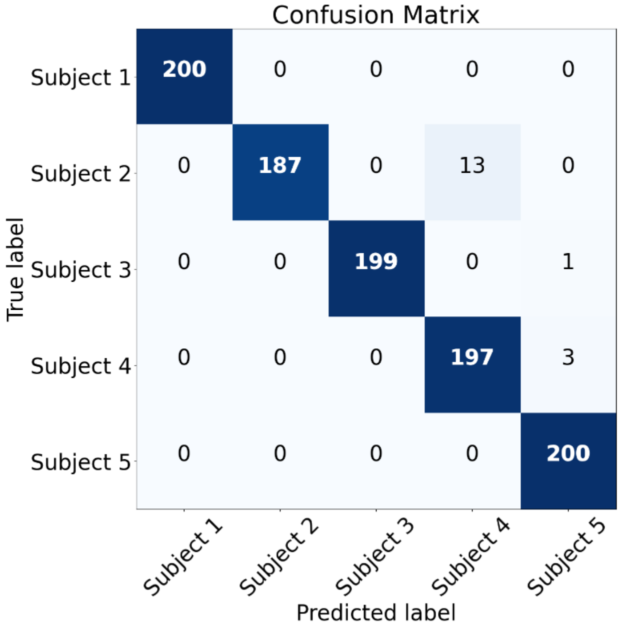

FIGURE 20. Confusion matrix. Siamese facial recognition system using SMRT algorithm in the second environment.

but this will depend on other factors such as the number of images obtained by the drone, the quality of the camera, the light intensity on a cloudy day, etc. From the confusion matrix in Fig. 21, we can estimate the accuracy, precision, recall, and F1-score of the Siamese facial recognition system using the SMRT algorithm and compare them against the values of the Siamese facial recognition without using the novel algorithm as shown in Table 12.

D. STATE OF THE ART IN FACIAL RECOGNITION

In the results presented, we focused more on the first two testing environments since the number of images, light intensity, and environment are parameters we can control. Using the SMRT algorithm enhances facial recognition, especially during drone tracking. In this subsection, We compare our method with other state-of-the-art facial recognition models employing Siamese networks and present our results from the second environment. Table13 presents several facial recognition models. Most of these models have been trained on thousands of data points. In our case, we use transfer learning, which means we can use the weights of an already trained network, such as FaceNet [56], and only train the final layers of the Inception network by updating their weights. This approach allows us to achieve our goal of recognizing only the individuals who are in our database.

E. FACE TRACKING TIME

The drone tracks faces in front of the camera. To display the tracking results, we conducted an experiment as shown in Fig. 22, where the person stands in front of the drone at point A, then moves to point B, returns to point A, and finally moves to point C. The elapsed time between each point is calculated. Table 14 shows the response times from one point to another. It can be observed that for short distances, the time is long, which may be due to the need for improvement in the SMRT + SORT algorithms to increase their efficiency. Additionally, the time from point B to point A is slightly different because the person's face is not detected correctly for a few seconds. The same occurs from point A to point C, where the drone fails to detect the face, and the person needs to move slightly to be detected. These errors can be addressed in future research by training the Siamese network with more data and improving the tracking algorithm. Furthermore, it is necessary to analyze whether the processing of the SMRT algorithm is performed on the CPU or GPU of the Jetson TX2. Finally, it can be appreciated that the drone is capable of recognizing a person in a database and tracking their movement.

TABLE 10. Performance metrics for the siamese model.

Person	Acc (%)	Precision (%)	Recall (%)	F1-Score (%)
Subject 1	90.00	92.62	86.92	89.68
Subject 2	90.00	87.68	93.08	90.30

TABLE 11. Performance metrics for the siamese model + SMRT algorithm.

Person	Acc (%)	Precision (%)	Recall (%)	F1-Score (%)
Subject 1	81.15	95.51	65.38	77.63
Subject 2	81.15	73.68	96.92	83.72

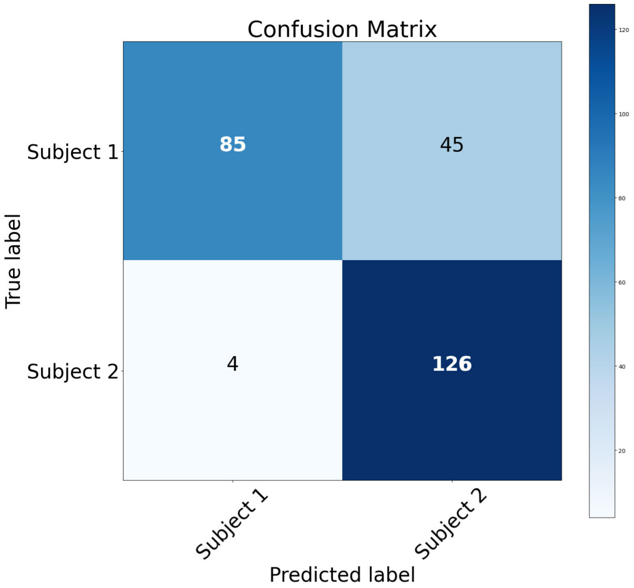

FIGURE 21. Confusion matrix. Siamese facial recognition system using SMRT algorithm in the third environment.

TABLE 12. Performance metrics for our siamese model against siamese model+SMRT algorithm in the third environment.

Metrics	Siamese model	Siamese model + SMRT
Accuracy (%)	90.00	81.15
Precision (%)	90.15	84.60
Recall (%)	90.00	81.15
F1-Score (%)	90.08	82.83

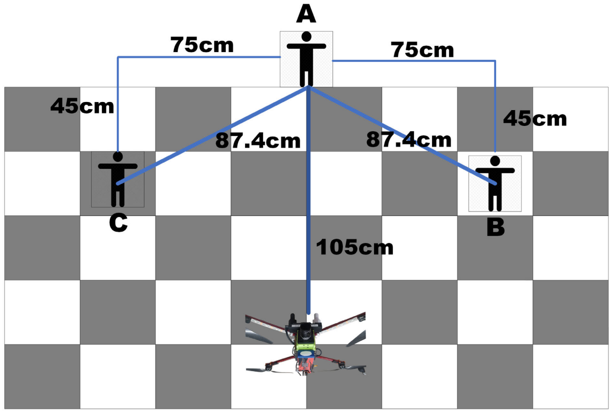

FIGURE 22. Face tracking experiment.

VI. PERFORMANCE EVALUATION

To address the computational cost, inference time, and running time of our proposed method, we conducted several evaluations.

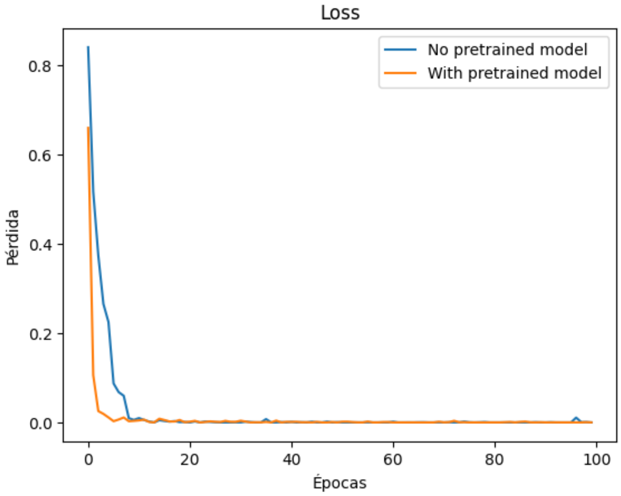

FIGURE 23. Comparison of loss results between the pretrained model and the non-pretrained model.

A. COMPUTATIONAL COST

The computational cost is evaluated in terms of FLOPS (Floating Point Operations Per Second). Our facial recognition system comprises a Siamese network for training. For inference, we only use the InceptionV2 model with the weights from the trained Siamese model. Therefore, we have measured the FLOPS for each layer of the inference InceptionV2 network, as shown in Table 15. The total computational cost of the inference model is approximately 0.48 billion FLOPS, indicating its efficiency and feasibility for real-time applications on the Jetson TX2 platform.

B. INFERENCE TIME

Inference time refers to the time it takes for the model to process an input and produce an output. In our experiments, the inference time is measured by running the model on the Jetson TX2 and calculating the time taken to process a frame. The inference time for our model was approximately 0.12 milliseconds per frame, which is sufficient for real-time facial recognition applications.

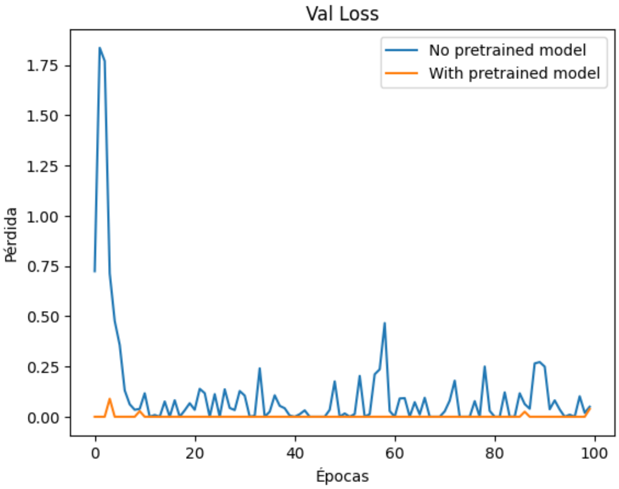

FIGURE 24. Comparison of validation loss results between the pretrained model and the non-pretrained model.

C. RUNNING TIME

The running time encompasses the total time taken for the entire process, including facial detection, facial recognition, the use of the SMRT algorithm, and data storage. The tracking time is presented in Table 14. The average execution time, excluding tracking, was approximately 0.24 milliseconds, demonstrating the system's capability to operate effectively in real-time scenarios. The tracking time is higher because the speed of the movement of the subject in front of the camera is slow, allowing for better control of the UAV in case of an emergency.

VII. DISCUSSION

The facial recognition system consists of three important components: facial detection, facial recognition, and face tracking. Each task is essential for the operation of the system. For facial detection, we have utilized the Haar Cascade algorithm from OpenCV. Although this algorithm is not very effective compared to others using deep learning, it remains useful for conserving computational resources on the Jetson TX2.

Three experiments have been conducted where it is necessary to measure the accuracy of the facial recognition system and the time it takes to track a person. In the first environment, the drone was positioned above the desk. This is because we need to simulate a setting with ideal flight conditions, free from vibrations or other disturbances that could affect facial recognition.

During the first experiment, the drone captured 850 images of the person in front of it. The facial recognition system was trained using the weights of FaceNet. As a result, the Siamese network achieved an accuracy of 98.21%. Subsequently, we incorporated our new algorithm SORT+SMRT, achieving an accuracy of 99.62%.

TABLE 13. Performance comparison of different methods.

Method	Authors	Year	Architecture	Networks	Verif. Metric	Training Set	Accuracy (%)
Ben Fredj	Ben Fredj et al. [66]	2020	GoogleNet	1	Softmax with center loss	CASIA WebFace (494 k, 10 k)	99.
MML	Wei et al. [67]	2020	Inception ResNet-V1 [156]	1	MML Loss	VGGFace2 (3.05 M, 8 K)	99.63
IAM	Sun et al. [68]	2020	Inception ResNet-V1 [156]	1	IAM loss	CASIA WebFace (494 k, 10 k)	99.12
RCM loss	Wu et al. [69]	2020	ResNet-18	1	Rotation Consistent Margin loss	CASIA WebFace (494 k, 10 k)	98.91
ACNN	Ling et al. [70]	2020	ResNet-100	1	ArcFace Loss	DeepGlint-MS1M (3.9 M, 86 K)	99.83
LMC	Wu and Wu [71]	2020	ResNet32	1	LMC loss	CASIA WebFace (494 k, 10 k)	98.13
SDLMC	Wu and Wu [71]	2020	ResNet32	1	SDLMC loss	CASIA WebFace (494 k, 10 k)	98.39
DLMC	Wu and Wu [71]	2020	ResNet32	1	DLMC loss	CASIA WebFace (494 k, 10 k)	98.07
SI (Ours)	Diego Herrera	2024	Siamese-Inceptionv2	1	Triplet loss	Transfer-learning FaceNet weight	97.72
SISMRT (Ours)	Diego Herrera	2024	Siamese-Inceptionv2 + SMRT	1	Triplet loss	Transfer-learning FaceNet weight	99.32

TABLE 14. Tracking time between specific points.

Point	Time (Sec)	Distance (cm)
From A to B	20.5	87.4
From B to A	17.4	87.4
From A to C	34.8	87.4

This improvement suggests that integrating the SMRT algorithm with the Siamese network significantly enhances its facial recognition capabilities, potentially eliminating the need for extensive additional training with new data to achieve high levels of accuracy. This is because the SMRT algorithm tracks the name of the person in front of the camera and, even if the Siamese model predicts a different person in front of the camera, the SMRT algorithm will continue to assign the previous name unless the person disappears from the camera frame.

For the second experiment, the environment is a classroom where the drone is flown safely. Two parameters are measured: first, the accuracy of facial recognition while the drone is flying and tracking a person, and second, the time elapsed between point A and point C.

A total of 1000 images were captured during the drone's flight in the second environment, achieving an accuracy of 97.72%. The accuracy was lower compared to the first environment, demonstrating that external factors such as drone vibration, rotation, light intensity, etc., significantly affect facial recognition.

Next, we applied the SORT+SMRT algorithm to analyze how much the accuracy improves. We obtained an accuracy of 99.32%, showing an improvement of almost 2%. This is interesting because we were able to increase the accuracy of the facial recognition system twice more than the first environment and achieve similar and better results than other models. The SMRT algorithm can still be improved by incorporating image processing techniques such as noise removal and reducing algorithm execution time, as well as optimizing how the SMRT algorithm is integrated.

For the third experiment, we conducted our research on a football field. For safety reasons, we chose to conduct this experiment with only two participants: the research author and a lab partner. The objective was to analyze the behavior of the facial recognition system in a real environment and under real conditions. The day was cloudy, causing significant variations in light intensity. Additionally, there was light rain, which could have affected facial recognition. It is important to note that part of the facial recognition system includes the drone taking precautions if a person is very close to it, as explained in Section III-C. The system also features automatic landing.

During takeoff, there were no strong winds, allowing the drone to maintain its position. Throughout the flight, the drone captured 130 images of each person, totaling 260 images. The obtained result was an accuracy level of approximately 90%. However, compared to the previous two experiments, the model exhibited decreased accuracy due to various external factors such as vibrations, changes in natural light intensity, and light winds that could have shifted the drone during flight. Simultaneously, the SORT+SMRT algorithm was executed, achieving an accuracy level of 81.15%, which did not meet our expectations.

This is attributed to the operation of the siamese network, which has two inputs: a database and images captured by the camera. Given that the database was created in an environment with artificial light, the accuracy level is better indoors than outdoors. The lower accuracy of the SMRT algorithm may be due to using two previous images before the current one to obtain a better reference. If these two previous images were classified as belonging to a different person, the SMRT algorithm may maintain this incorrect classification, even if the siamese network correctly identifies the person

TABLE 15. Details of the facial recognition model Layers.

Type	Output Size	Depth	#1x1	#3x3	#5x5	Params	FLOPS
input_layer_1	(None, 3, 96, 96)	0	0	0	0	0	0
zero_padding2d_23	(None, 3, 102, 102)	0	0	0	0	0	0
Conv1	(None, 64, 48, 48)	1	0	0	0	9K	43M
max_pool + norm	(None, 64, 24, 24)	0	0	0	0	0	0
Conv2	(None, 64, 24, 24)	1	0	0	0	4K	4M
zero_padding2d_25	(None, 64, 26, 26)	0	0	0	0	0	0
Conv3	(None, 192, 24, 24)	1	0	0	0	111K	127M
max_pool + norm	(None, 192, 12, 12)	0	0	0	0	0	0
Inception_3a	(None, 256, 12, 12)	6	64	96, 128	16, 32	165K	46M
Inception_3b	(None, 320, 12, 12)	6	64	96, 128	32, 64	230K	61M
Inception_3c	(None, 640, 6, 6)	4	0	128, 256	0	400K	40M
Inception_4a	(None, 640, 6, 6)	6	256	96, 192	32, 64	549K	34M
Inception_4e	(None, 1024, 3, 3)	4	0	160, 256	64, 128	720K	21M
Inception_5b	(None, 736, 3, 3)	4	256	96, 384	0	666K	11M
Avg_pool	(None, 736, 1, 1)	0	0	0	0	0	0
Fully conn	(None, 128)	1	0	0	0	94K	0
L2 normalization	(None, 128)	0	0	0	0	0	0
Total						**3.7M**	**0.4B**

again. Subsequently, when the siamese network briefly stops detecting and then resumes, or when the SORT algorithm loses track of the face, the SMRT algorithm may reassign the correct person's name.

Both the siamese network and the SORT and SMRT algorithms work together for precise tracking and identification of individuals during drone flights.

The precision level of this latest experiment could be enhanced by adding more photos of people in natural lighting environments or applying data augmentation techniques to obtain a variety of images. Despite this, the achieved precision level is close to that obtained by other facial recognition methods, as shown in Table 12. It is important to highlight that our research presents results from experiments conducted while the drone was flying and tracking individuals, unlike other studies that only show images captured by a drone without considering its behavior during flight.

Next, we present the results of the time it takes for the drone to travel from point A to point B, from point B back to point A, and finally from point A to point C. We observed that the time required to travel from A to B was approximately 20 seconds, primarily because the person in front of the drone was moving at a similar speed to the drone. If the person moved too quickly, the drone could not keep up, as seen when the person moved from B to A in 17.4 seconds, a faster time because the person was facing the drone while moving. However, when traveling from point A to point B, the drone momentarily lost track of the person's face and could not detect it until the person slightly moved their head, after which the drone quickly recognized and resumed tracking. The same occurred when traveling from point A to point C; the drone lost track of the person's face because they were moving faster than the drone and went out of the camera's field of view. The person had to step back for the drone to detect their face again and resume tracking.

One reason the drone loses sight of faces is due to using the Haar cascade algorithm from OpenCV for facial detection, which is only effective with frontal faces and cannot detect rotated faces. Another reason for prolonged times is the execution time required for the Siamese network along with the SORT and SMRT algorithms. In summary, we have identified two key areas requiring improvement: first, changing the face detection from Haar cascade to another deep learning-based algorithm; second, optimizing the execution time of the siamese network + SORT + SMRT. Additionally, we plan to train the Siamese neural network with more diverse data, including different light intensities and environments, which will also incorporate deep learning for facial detection, human pose recognition, tracking control accuracy measurement, and robust tracking algorithm.

In conclusion, our work demonstrates the potential to contribute to Latin American society, which faces high crime rates, through the use of drones capable of detecting, recognizing, and tracking wanted individuals. Our implementation of the Siamese network + SORT + SMRT contributes to achieving the system's ultimate goal. It is important to mention that this research is conducted to contribute to society and strictly prohibits its use for purposes that threaten the lives of living beings.

VIII. CONCLUSION

In this paper, we have developed an autonomous drone capable of recognizing a person's face and following them in GPS-denied environments. The facial recognition system includes our new algorithm SMRT, which enhances facial recognition accuracy. Our proposed method achieves an accuracy of 94.45% using the SMRT algorithm, which is acceptable compared to other conventional algorithms given that the Siamese network is untrained. Field test results indicate that the proposed method performs well in indoor environments with artificial lighting, although the dataset lacks diversity. The drone has demonstrated the ability to perform autonomous flights and autonomous person tracking. The benefits obtained from this research allow us to implement a new version of our drone with gait recognition and human pose estimation for improved tracking capability. The implementation of the facial recognition system in drones issues a deeper understanding of the potential use of drones to reduce crime and violence in the world.

REFERENCES

[1] G. Sánchez-Rentería, F. J. Bonilla-Escobar, A. Fandiño-Losada, and M. I. Gutiérrez-Martinez, "Observatorios de convivencia y seguridad ciudadana: Herramientas para la toma de decisiones y gobernabilidad," *Revista Peruana de Medicina Experim. Salud Pública*, vol. 33, no. 2, p. 362, Jun. 2016, doi: 10.17843/rpmesp.2016.332.2203.

[2] *Homicide-Estimates | DataUNODC*. Accessed: May 18, 2022. [Online]. Available: https://dataunodc.un.org/content/homicide-estimates

[3] A. Izquierdo, C. Pessino, and G. Vuletin, *Better Spending for Better Lives: How Latin America and the Caribbean Can Do More with Less | Publications*. Accessed: Aug. 24, 2024. [Online]. Available: https://publications.iadb.org/en/publications/english/viewer/Better-Spending-for-Better-Lives-How-Latin-America-and-the-Caribbean-Can-Do-More-with-Less.pdfLives-How-Latin-America-and-the-Caribbean-Can-Do-More-with-Less.pdf

[4] A. Devi and A. Marimuthu, "An efficient self-updating face recognition system for plastic surgery face," *ICTACT J. Image Video Process.*, vol. 7, no. 1, pp. 1307–1317, Aug. 2016, doi: 10.21917/ijivp.2016.0191.

[5] *Counseling With Artificial Intelligence—Counseling Today*. Accessed: Nov. 1, 2018. [Online]. Available: https://ct.counseling.org/2018/01/counseling-artificial-intelligence/

[6] *Real-Time Facial Recognition Technology | Oosto*. Accessed: Nov. 14, 2023. [Online]. Available: https://oosto.com/

[7] *Your Face is, or Will be, Your Boarding Pass—The New York Times*. Accessed: Oct. 16, 2023. [Online]. Available: https://www.nytimes.com/2021/12/07/travel/biometrics-airports-security.html

[8] N. Delbiaggio, "A comparison of facial recognition's algorithms," M.S. thesis, Degree Programme Bus. Inf. Technol., Haaga-Helia Univ. Appl. Sci., Helsinki, Finland, 2017. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/132808/Delbiaggio_Nicolas.pdf?sequence=1

[9] K. Simonyan and A. Zisserman. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Accessed: Jun. 1, 2022. [Online]. Available: http://www.robots.ox.ac.uk/

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015, *arXiv:1503.03832*.

[12] Z. He, "Deep learning in image classification: A survey report," in *Proc. 2nd Int. Conf. Inf. Technol. Comput. Appl. (ITCA)*, Dec. 2020, pp. 174–177, doi: 10.1109/ITCA52113.2020.00043.

[13] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," 2016, *arXiv:1606.05718*.

[14] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, 2015, pp. 1–8.

[15] S. Sambolek and M. Ivasic-Kos, "Automatic person detection in search and rescue operations using deep CNN detectors," *IEEE Access*, vol. 9, pp. 37905–37922, 2021, doi: 10.1109/ACCESS.2021.3063681.

[16] U. Azmat, S. S. Alotaibi, M. Abdelhaq, N. Alsufyani, M. Shorfuzzaman, A. Jalal, and J. Park, "Aerial insights: Deep learning-based human action recognition in drone imagery," *IEEE Access*, vol. 11, pp. 83946–83961, 2023, doi: 10.1109/ACCESS.2023.3302353.

[17] F. Schiano, D. Natter, D. Zambrano, and D. Floreano, "Autonomous detection and deterrence of pigeons on buildings by drones," *IEEE Access*, vol. 10, pp. 1745–1755, 2022, doi: 10.1109/ACCESS.2021.3137031.

[18] C.-J. Chen, Y.-Y. Huang, Y.-S. Li, Y.-C. Chen, C.-Y. Chang, and Y.-M. Huang, "Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying," *IEEE Access*, vol. 9, pp. 21986–21997, 2021, doi: 10.1109/ACCESS.2021.3056082.

[19] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, "Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube: Preparation of novel high copper ions removal membranes," *J. Chem. Technol. Biotechnol.*, vol. 91, no. 8, pp. 2322–2330, Aug. 2016, doi: 10.1002/jctb.4820.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1–9. [Online]. Available: http://code.google.com/p/cuda-convnet/

[21] L. Meng, T. Hirayama, and S. Oyanagi, "Underwater-drone with panoramic camera for automatic fish recognition based on deep learning," *IEEE Access*, vol. 6, pp. 17880–17886, 2018, doi: 10.1109/ACCESS.2018.2820326.

[22] A. Mukashev, L.-D. Van, S. Sharma, M. F. Tandia, and Y.-C. Tseng, "Person tracking by fusing posture data from UAV video and wearable sensors," *IEEE Sensors J.*, vol. 22, no. 24, pp. 24150–24160, Dec. 2022, doi: 10.1109/JSEN.2022.3218373.

[23] K. Kim, J. Kim, H.-G. Lee, J. Choi, J. Fan, and J. Joung, "UAV chasing based on YOLOv3 and object tracker for counter UAV systems," *IEEE Access*, vol. 11, pp. 34659–34673, 2023, doi: 10.1109/ACCESS.2023.3264603.

[24] T. Keawboontan and M. Thammawichai, "Toward real-time UAV multi-target tracking using joint detection and tracking," *IEEE Access*, vol. 11, pp. 65238–65254, 2023, doi: 10.1109/ACCESS.2023.3283411.

[25] M. Alhafnawi, H. A. B. Salameh, A. Masadeh, H. Al-Obiedollah, M. Ayyash, R. El-Khazali, and H. Elgala, "A survey of indoor and outdoor UAV-based target tracking systems: Current status, challenges, technologies, and future directions," *IEEE Access*, vol. 11, pp. 68324–68339, 2023, doi: 10.1109/ACCESS.2023.3292302.

[26] D. Herrera and H. Imamura, "Design of facial recognition system implemented in an unmanned aerial vehicle for citizen security in Latin America," *ITM Web Conf.*, vol. 27, May 2019, Art. no. 04002, doi: 10.1051/itmconf/20192704002.

[27] *El Cañón Del Colca Registra El Mayor Número De Turistas Perdidos en Arequipa | RPP Noticias*. Accessed: Nov. 15, 2023. [Online]. Available: https://rpp.pe/peru/actualidad/el-canon-del-colca-registra-el-mayor-numero-de-turistas-perdidos-en-arequipa-noticia-1166600?ref=rpp

[28] E. Çintas, B. Özyer, and E. Simsek, "Vision-based moving UAV tracking by another UAV on low-cost hardware and a new ground control station," *IEEE Access*, vol. 8, pp. 194601–194611, 2020, doi: 10.1109/ACCESS.2020.3033481.

[29] J. Li, D. H. Ye, M. Kolsch, J. P. Wachs, and C. A. Bouman, "Fast and robust UAV to UAV detection and tracking from video," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 3, pp. 1519–1531, Jul. 2022, doi: 10.1109/TETC.2021.3104555.

[30] H.-J. Hsu and K.-T. Chen, "DroneFace: An open dataset for drone research," in *Proc. 8th ACM Multimedia Syst. Conf.*, Jun. 2017, pp. 187–192, doi: 10.1145/3083187.3083214.

[31] *Face++—Face++ Cognitive Services*. Accessed: Nov. 15, 2023. [Online]. Available: https://www.faceplusplus.com/

[32] *Image Recognition Software, Ml Image & Video Analysis—Amazon Rekognition—AWS*. Accessed: Nov. 15, 2023. [Online]. Available: https://aws.amazon.com/rekognition/

[33] H.-J. Hsu and K.-T. Chen, "Face recognition on drones: Issues and limitations," in *Proc. 1st Workshop Micro Aerial Vehicle Netw., Syst., Appl. Civilian Use*, May 2015, pp. 39–44, doi: 10.1145/2750675.2750679.

[34] R. Jurevičius, N. Goranin, J. Janulevičius, J. Nugaras, I. Suzdalev, and A. Lapusinskij, "Method for real time face recognition application in unmanned aerial vehicles," *Aviation*, vol. 23, no. 2, pp. 65–70, Dec. 2019, doi: 10.3846/aviation.2019.10681.

[35] *Dlib C++ Library*. Accessed: Nov. 15, 2023. [Online]. Available: http://dlib.net

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[37] A. Srivastava, T. Badal, P. Saxena, A. Vidyarthi, and R. Singh, "UAV surveillance for violence detection and individual identification," *Automated Softw. Eng.*, vol. 29, no. 1, May 2022, doi: 10.1007/s10515-022-00323-3.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. *Densely Connected Convolutional Networks*. Accessed: Jan. 4, 2024. [Online]. Available: https://github.com/liuzhuang13/DenseNet.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[41] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[42] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," 2017, *arXiv:1707.07012*.

[43] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *Computer Analysis of Images and Patterns* (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)), vol. 6855, 2011, pp. 332–339, doi: 10.1007/978-3-642-23678-5_39.

[44] M. M. Soliman, M. H. Kamal, M. A. E.-M. Nashed, Y. M. Mostafa, B. S. Chawky, and D. Khattab, "Violence recognition from videos using deep learning techniques," in *Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2019, pp. 80–85, doi: 10.1109/ICICIS46948.2019.9014714.

[45] J. Sandino, F. Vanegas, F. Maire, P. Caccetta, C. Sanderson, and F. Gonzalez, "UAV framework for autonomous onboard navigation and people/object detection in cluttered indoor environments," *Remote Sens.*, vol. 12, no. 20, p. 3386, Oct. 2020, doi: 10.3390/rs12203386.

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[47] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015, doi: 10.1007/s11263-014-0733-5.

[48] N. Davis, F. Pittaluga, and K. Panetta, "Facial recognition using human visual system algorithms for robotic and UAV platforms," in *Proc. IEEE Conf. Technol. Practical Robot Appl. (TePRA)*, Apr. 2013, pp. 1–5.

[49] A. S. Priambodo, F. Arifin, A. Nasuha, and A. Winursito, "Face tracking for flying robot quadcopter based on Haar cascade classifier and PID controller," *J. Phys., Conf. Ser.*, vol. 2111, no. 1, Nov. 2021, Art. no. 012046, doi: 10.1088/1742-6596/2111/1/012046.

[50] P. M. Wyder et al., "Autonomous drone hunter operating by deep learning and all-onboard computations in GPS-denied environments," *PLoS One*, vol. 14, no. 11, 2019, Art. no. e0225092, doi: 10.1371/journal.pone.0225092.

[51] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[52] *Mavros—ROS Wiki*. Accessed: Nov. 15, 2023. [Online]. Available: http://wiki.ros.org/mavros

[53] T. Zhou and Y. Liu, "Long-term person tracking for unmanned aerial vehicle based on human–machine collaboration," *IEEE Access*, vol. 9, pp. 161181–161193, 2021, doi: 10.1109/ACCESS.2021.3132077.

[54] OpenCV. *Cascade Classifier—OpenCV 3.4 Documentation*. Accessed: Apr. 13, 2024. [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[55] OpenCV Contributors. *OpenCV Documentation*. Accessed: Apr. 13, 2024. [Online]. Available: https://docs.opencv.org/4.5.5/

[56] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[57] Univ. Massachusetts, Boston, MA, USA. *LFW Face Database: Main*. Accessed: May 31, 2022. [Online]. Available: http://vis-www.cs.umass.edu/lfw/

[58] D. Herrera. *Face Recognition Inception KERAS—GitHub Repository*. Accessed: Jul. 5, 2022. [Online]. Available: https://github.com/DiegoHerrera1890/facial-recognition-system-implemented-in-an-unmanned-aerial-vehicle/tree/master/Face_recognition_Inception_KERAS

[59] A. Bewley. *Sort/Sort.py at Master—GitHub Repository*. Accessed: Apr. 13, 2024. [Online]. Available: https://github.com/abewley/sort/blob/master/sort.py

[60] D. Herrera. *Pixhawk Connected to Jetson Tx2 Devkit—GitHub Repository*. Accessed: Apr. 13, 2024. [Online]. Available: https://github.com/DiegoHerrera1890/Pixhawk-connected-to-Jetson-Tx2-devkit

[61] S.-C. Chong, A. B. J. Teoh, and T.-S. Ong, "Unconstrained face verification with a dual-layer block-based metric learning," *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 1703–1719, Jan. 2017.

[62] C. Xiong, L. Liu, X. Zhao, S. Yan, and T.-K. Kim, "Convolutional fusion network for face verification in the wild," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 517–528, Mar. 2016.

[63] J. Zhang, X. Jin, Y. Liu, A. K. Sangaiah, and J. Wang, "Small sample face recognition algorithm based on novel Siamese network," *J. Inf. Process. Syst.*, vol. 14, no. 6, pp. 1464–1479, 2018, doi: 10.3745/JIPS.02.0101.

[64] M. Heidari and K. Fouladi-Ghaleh, "Using Siamese networks with transfer learning for face recognition on small-samples datasets," in *Proc. Int. Conf. Mach. Vis. Image Process. (MVIP)*, 2020, pp. 1–4, doi: 10.1109/MVIP49855.2020.9116915.

[65] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[66] H. Ben Fredj, S. Bouguezzi, and C. Souani, "Face recognition in unconstrained environment with CNN," *Vis. Comput.*, vol. 37, no. 2, pp. 217–226, 2021.

[67] X. Wei, H. Wang, B. Scotney, and H. Wan, "Minimum margin loss for deep face recognition," *Pattern Recognit.*, vol. 97, Jan. 2020, Art. no. 107012.

[68] J. Sun, W. Yang, R. Gao, J.-H. Xue, and Q. Liao, "Inter-class angular margin loss for face recognition," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115636.

[69] Y. Wu, Y. Wu, R. Gong, Y. Lv, K. Chen, D. Liang, X. Hu, X. Liu, and J. Yan, "Rotation consistent margin loss for efficient low-bit face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6865–6875.

[70] H. Ling, J. Wu, J. Huang, J. Chen, and P. Li, "Attention-based convolutional neural network for deep face recognition," *Multimedia Tools Appl.*, vol. 79, nos. 9–10, pp. 5595–5616, Mar. 2020.

[71] B. Wu and H. Wu, "Angular discriminative deep feature learning for face verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2133–2137.

[72] B. Ma, Z. Liu, W. Zhao, J. Yuan, H. Long, X. Wang, and Z. Yuan, "Target tracking control of UAV through deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 5983–6000, Jun. 2023, doi: 10.1109/TITS.2023.3249900.

[73] B. Ma, Z. Liu, F. Jiang, W. Zhao, Q. Dang, X. Wang, J. Zhang, and L. Wang, "Reinforcement learning based UAV formation control in GPS-denied environment," *Chin. J. Aeronaut.*, vol. 36, no. 11, pp. 281–296, Nov. 2023, doi: 10.1016/j.cja.2023.07.006.

DIEGO A. HERRERA OLLACHICA (Member, IEEE) received the B.S. degree in mechatronic engineering from the Technological University of Peru, Lima, Peru, in 2017, and the M.S. degree in information system science from Soka University, Tokyo, Japan, in 2020, where he is currently pursuing the Ph.D. degree in information system science.

From 2016 to 2018, he was a Research and Development Engineer at LabTop Peru Inc., Lima. From 2020 to 2023, he was a Research Assistant for the JICA-JST SATREPS-EARTH project in Tokyo. His research interests include drones, artificial intelligence, deep learning, and robotics applied to help society. He was awarded as the Best Oral Presentation at the 6th International Postgraduate Conference on Biotechnology, in 2023, at the National University of Singapore.

BISMARK K. ASIEDU ASANTE (Member, IEEE) received the B.S. degree in computer science and physics and the M.Phil. degree in computer science from the University of Ghana, in 2012 and 2017, respectively, and the Ph.D. degree in information system science engineering from Soka University, in 2024.

In 2024, he assumed the position of an Assistant Professor at Soka University, specifically with the Department of Information System Science Engineering. During his Ph.D. research, he published papers on speech enhancement and obstacle avoidance strategies for the visually impaired. His research interests include artificial intelligence and deep learning, with a focus on applying these technologies to address human and environmental challenges.

HIROKI IMAMURA (Member, IEEE) received the B.S. degree in engineering from Soka University, Japan, in 1997, and the M.S. and Ph.D. degrees in information science from JAIST, Japan, in 1999 and 2023, respectively.

From 2003 to 2009, he was an Assistant Professor with Nagasaki University, Japan. From 2009 to 2020, he was an Associate Professor at Soka University, where he has been a Professor, since 2020. His research interests include image processing, artificial intelligence, and XR.