ISSN 2007-9737

Drone Based Face Recognition System Using MTCNN and Facenet in ArduPilot Software Platform

Wangjam Niranjan Singh, Deepjyoti Sharma, Arunabha Dutta, Tama Roy

Assam University,
Department of Computer Science and Engineering,
India

{niranwang, deepjyotisarmah37, arunavdutta27, roytama3019}@gmail.com,

Abstract. The security and surveillance industries have seen significant changes as a result of drones, often known as unmanned aerial vehicles, or UAVs. In recent years, integrating facial recognition technology with drones has emerged as the best way to improve real-time identifying capabilities. A crucial field of study in computer vision and artificial intelligence is face recognition. In this paper, we have used MTCNN (Multi-task Cascaded Convolutional Networks) and Facenet for face recognition. Additionally, we compare the performance of the MTCNN method using an existing HOG (Histogram Of Gradient) method. For the simulation of the drone based face recognition system, we have used the ArduPilot software platform. Tools used for simulation purpose are Dronekit, Mavproxy and Mission Planner. The comparison results sheds some light on the algorithm's adaptability, accuracy, and detection rates as well as its resilience.

Keywords. Drone, face recognition, multi-task cascaded convolutional networks, facenet, ardupilot software.

1 Introduction

A sophisticated technology called a face recognition system is used to recognize and authenticate people by examining specific elements of their facial features. This system analyses facial patterns from photos or video frames and compares them to a database of recognized faces using some algorithms. The technology calculates the chance of a match between the captured face and the saved profiles using complex algorithms, enabling precise identification or authentication. When using UAVs to look for missing children or elderly people, the UAV must be aware of its targets before the search can begin [5]. Face identification, alignment, and feature extraction are three crucial elements in the procedure. The method finds facial regions in an image or video stream during face detection.

Therefore, facial alignment tries to minimize variances brought on by various positions by standardizing the orientation of the identified face. The last stage involves extracting unique facial features, such the positions of the lips, nose, and eyes, which are then converted into a unique numerical representation called a facial signature. The distance between a UAV and its targets directly affects the pixel size of the face photos it captures. Since UAVs take images from the air, their altitude and distance limit how close they can get to ground-based subjects, impacting the quality of the images.

Systems for facial recognition have many uses, from security and surveillance to user authentication and access management. It can help law enforcement organization find missing people, track down criminals, and find people who are needed for questioning,hence reduces the manual load. They assist in spotting possible dangers and preventing unauthorized access in security environments. They make it possible for consumer electronics, such cellphones, to unlock gadgets quickly and securely. Although facial recognition systems have been around for a while, there is still a lot of ongoing research in this area. The topic can be broken down into three categories: pre-processing, face detection, feature extraction and finally classification.

The desire for items with greater features has increased as a result of emerging technology. One of the most significant aspects of our lives is our face. The face is a key focus in knowledge processing, particularly in visual perception, form recognition, and design interpretation. It plays a crucial role in applications involving image analysis and observation. Combining different strategies is currently the most important factor. The idea of a face-recognition drone, which can identify the person standing next to it, has emerged in response to the growing demand for drones used for surveillance. Face detection and the integration of this component into small drones is the most difficult aspect of computer vision.

Face detection by drones has various difficulties, which has an impact on how face identification is carried out by the drone. The size of the facial photographs in pixels is directly impacted by issues like the separation between drones and their targets. Drones take pictures, hence Drone's elevations from above keep them apart from their ground targets. In addition, altitudes and pitch angles provide angles of depression from drones to their objectives. Thus, the number of facial photographs captured by drones may be substantial. Besides, flight and speed the performance could be harmed and the quality of the facial photographs could be impacted by altitude. Due to the potential effects of speed and flight attitude be offset by using the right settings on aerial cameras. The face detection problem for drones is more difficult than regular face detection due to changes in scale and perspective.

The main goal of our research paper is to perform real-time facial recognition. Here, we takes a multidisciplinary approach, combining knowledge from computer vision. Our research aims to take advantage of the collaborative properties of two well-known algorithms: MTCNN and FaceNet, in order to advance the field of face recognition. The strengths of these algorithms will be combined in order to improve the precision and effectiveness of face recognition systems. Precision localization of facial regions is made possible by MTCNN's superior face and feature detection capabilities in pictures. FaceNet, on the other hand, is built to convert facial images into compact numerical representations known as embeddings, capturing the distinctive features of each face. By combining MTCNN and FaceNet, we can accurately detect and align faces using MTCNN before utilising FaceNet to provide meaningful embeddings. The system will be optimized for real-time performance, enabling rapid identification of persons of interest, in order to solve the special problems given by the military surveillance context. Additionally, privacy and ethical considerations will be crucial in making sure the system follows set rules and security measures.

2 Related Work

Here, we provides a thorough examination of the amount of knowledge and study that has already been done in this domain. This section seeks to lay a strong foundation for the current study, uncover gaps and trends in the research environment, and give context for the ensuing analysis and findings by diving into the works of prior researchers, scholars, and experts in the subject.

Viola-Jones' face detection system serves as the foundation for a number of implementation efforts [18]. This algorithm can process images quickly and achieve effective detection rates. The authors in [2], propose a real-time face detection system implemented on an field-programmable gate array (FPGA) using the AdaBoost algorithm and Haar features, designed with Verilog HDL. In [4], a Raspberry Pi-based face detection system using the Haar cascade classifier algorithm through the OpenCV tool has been proposed. Real-world face identification requires sophisticated models, which are frequently computationally demanding, to account for differences in position, expression, and illumination. A cascade architecture employing CNNs [13] is suggested as a solution to this problem, providing great discriminative ability and excellent performance. By employing a locally normalized histogram of gradient (HOG) [3] orientation features—akin to SIFT descriptors—in a densely overlapping grid, human detection is much enhanced, with false positives being reduced by more than ten times when compared to the most effective Haar wavelet-based detector. The Histogram of Oriented Gradient (HOG) algorithm, though accurate in detecting human shapes by extracting features from a dense grid and classifying them with a linear SVM, is computationally intensive [15]. A two-class C-SVM classifier employing HOG features verifies face candidates in the real-time face identification system presented by the authors in [9]. The system uses skin color, edges, and face area estimation to reduce the search region for a moving camera in open space. The authors in [14] propose a Smart Surveillance System utilizing the Histogram of Oriented Gradients (HOG) algorithm alongside the Haar Cascade algorithm. The results of the authors' investigation in [7] shows how drone-based face recognition is affected by altitudes, distances, and angles of depression show that existing systems, such as Face++ and Rekognition, function well in these scenarios. A fuzzy rule-based algorithm is used in [1] to create a real-time tracker that uses a DJI Phantom 3 drone to follow a specific human face, like that of a criminal.

Recently, Deep Learning (DL) has been widely applied in machine learning (ML) scenarios, leading to the development of numerous models for face detection and recognition protocols. Deeb et al. employed DL frameworks to enhance performance in face detection. The authors in [6] developed a low-cost, high-performing facial recognition system using DL that was installed in a UAV and was capable of identifying offenders. The authors in [17] proposed a hybrid convolutional network (ConvNet)-Restricted Boltzmann Machine (RBM) model for face verification task.

FaceNet [16] is able to recognize target identities from a variety of angles since it can achieve high accuracy without requiring 3D modeling. In [20], a deep MTCNN architecture is presented, which makes use of the natural correlation between alignment and detection to improve performance. A lightweight version of MTCNN is proposed in [19], which achieves about 67 percent reduction in the number of parameters compared to the original model.In [12], the authors proposed a method based on MTCNN and improved convolution neural network. The authors in [11] proposed a surveillance system that uses MTCNN and FaceNet for face detection and recognition, and it is built on the Jetson TX2 platform. The study emphasises the effectiveness of the combined technique in recognising people from drone-captured imagery and focuses on real-time performance.In [10], the authors proposed the use of FaceNet for face recognition and MTCNN for face detection. For precise and effective face recognition systems, the combination of these techniques offers a viable answer. In this paper, we enhance performance by first applying MTCNN for face detection. The output from MTCNN is then used as input for FaceNet to carry out face recognition. We validated the effectiveness of the proposed method through testing on the ArduPilot Software Platform.

3 Proposed Methodology

This section outlines three key segments: the first focuses on data collection and preparation, the second involves training and evaluating the facial recognition model, and the final segment covers real-time facial recognition using the trained model. Each of these steps is detailed below.

3.1 System Design

The system is made up of the following parts:
A drone: A drone is used for taking pictures of people's faces. The drone should have a top-notch camera that can record distant portraits of people's faces.
A face detection algorithm: The drone's images are processed using a face detection algorithm, which is used to find faces in the pictures.
A face recognition algorithm: An algorithm for identifying faces in photos that have already had their faces recognised is known as a facial recognition algorithm.
Database: Google Firebase is the database that we used. The database is split into two sections: "Realtime Database" and "Storage." The photos are kept in the "Storage" area, while the face encoding and metadata of the person are stored in the "Realtime Database" section.
SMS Service: SMS service provider "Twilio" uses

ISSN 2007-9737

the person metadata sent via database. To send messages to users, the SMS service provider employs an internal API. The message is send using HTTP protocol.

The system works as follows:

1. The drone captures the face a subject.
2. The face detection algorithm detects the face in the image.
3. The system then calls for encoding that is stored in the server based database for facial recognition.
4. The system compares the identified face to the database of known faces.
5. If the identified face is found in the database, the system returns the metadata of the person associated with the face. Hence recognition is done.
6. A SMS service uses the person metadata to send SMS to user using "Twillio" SMS service provider.
7. If the identified face is not found in the database, the system does not return metadata.

3.2 Drone Section

Face detection and recognition algorithms are included in the drone section. These are utilized to find and identify faces. The face encoding is retrieved from the database after the face has been identified using the face detection technique. The face recognition algorithm then uses the face encoding to recognise faces.

3.3 Server Section

The server is divided into two primary sections. The "Realtime Database" is where the face encoding and image metadata, including the person's name, ID, and other data, are kept. The administrator enters the encoding and the metadata in the real-time section since they need to be changed frequently. The "Storage" portion does not need to update as frequently as the

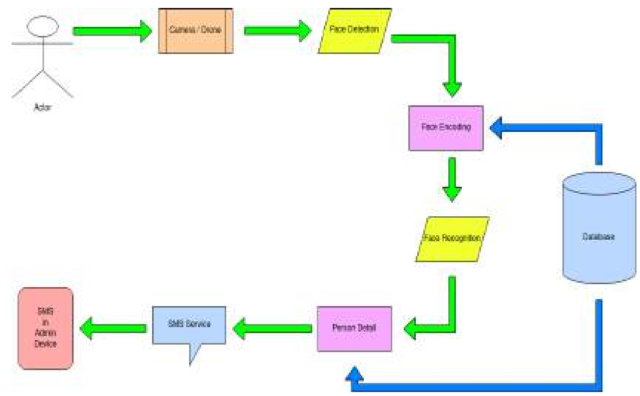

Fig. 1. Overall System Design

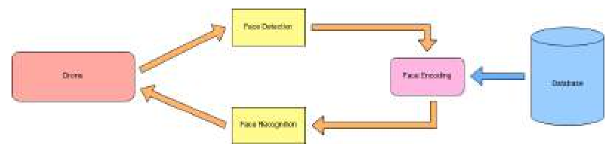

Fig. 2. Drone Section

real-time component because images are saved there. The real-time and storage sections are then used by the drone component for detection and recognition.

3.4 SMS Service

To send messages, "Twillio" SMS service has been used. The metadata is called when a face has been identified using the face recognition algorithm, and it is sent to the SMS server. The message is then created by the messaging server and sent to the administrator. The HTTP protocols are used for message retrieval and transmission to the administrator.

4 Data Collection and Preparation

Reliability and accuracy in drone-based facial recognition algorithms are largely dependent on the quality and appropriateness of the datasets used for training, validation, and assessment. The dataset is used as a typical sample of the real-world situations that facial recognition software for drones may meet. It covers a wide

ISSN 2007-9737

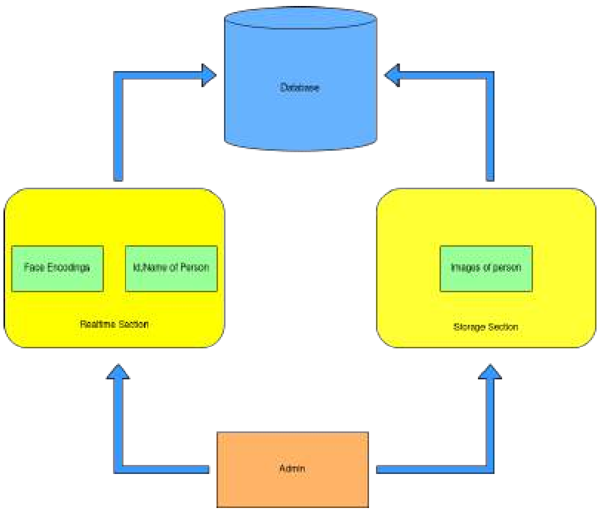

Fig. 3. Server Section

range of difficulties brought on by various lighting situations, external elements, and face emotions.

4.1 Collecting dataset

We have use open dataset known as DnHDrone-faces [8] dataset for the purpose of drone-based face recognition . Instead of using drones to take photos, the experimental setup in this dataset used a GoPro camera positioned on a cradle to take frontal facial shots of 11 subjects. The GoPro was placed at heights of 3, 4, and 5 metres by the writers in order to imitate drones flying at various altitudes. In order to replicate a 15-meter trajectory towards the target, pictures were shot at intervals of 0.5 meters at a distance of 2 to 17 meters from the ground-based subject. The angles produced between the horizontal plane and the line that extends from the GoPro to the top of the subjects were used to define the angles of depression between the aerial camera and the subjects. To counteract the effects of movement and facial expressions, the subjects were instructed to stand still, remove their glasses, look straight ahead, and maintain a deadpan expression.

4.2 Preprocessing and Data Alignment

A dataset must go through a number of data transformation steps during the crucial preprocessing stage in data analysis and machine learning to ensure its quality, consistency, and applicability for further analysis or modelling. The acquired dataset is not clean and contains lots of noises and redundant images, varying sizes hence the dataset needs to preprocessed before using it. Various preprocessing steps that are used are mentioned below. The technique of cutting an image's face portion so that it only depicts the features of the face is known as face alignment. The process of detecting a face from a taken image or a given image from a database is known as face detection. Using the DnHDroneface dataset, a pre-trained Haar-cascade classifier is used to align images. Initially, it identified a face from the Droneface dataset. We then cropped the face detection and saved it to the appropriate class folder. The format of the cropped photos was changed to [160 X 160], where the two digits represent the image's height and width, to guarantee that the cropped face would have all of the facial features and fewer background images.

5 Model Training and Evaluation

All the information on training details, training parameters, model training for masked face recognition, model evaluation, model improvement, and architecture of MTCNN and FaceNet are discussed in the following section.

5.1 Training and Testing details

MTCNN uses a number of phases and tasks that cooperate to precisely find faces of various sizes in a picture. The MTCNN architecture is end-to-end trained, which means that each step is taught concurrently to enhance total performance. The three steps of the cascaded architecture used by MTCNN are each dedicated to a certain task: S**tage 1 (Proposal Network):** A fully convolutional neural network (FCN) is used in this stage to produce candidate face regions. These areas are known as "proposals," and faces are probably present in them.

Stage 2 (Refinement Network):The recommendations are improved by eliminating false

doi: 10.13053/CyS-30-1-5147

ISSN 2007-9737

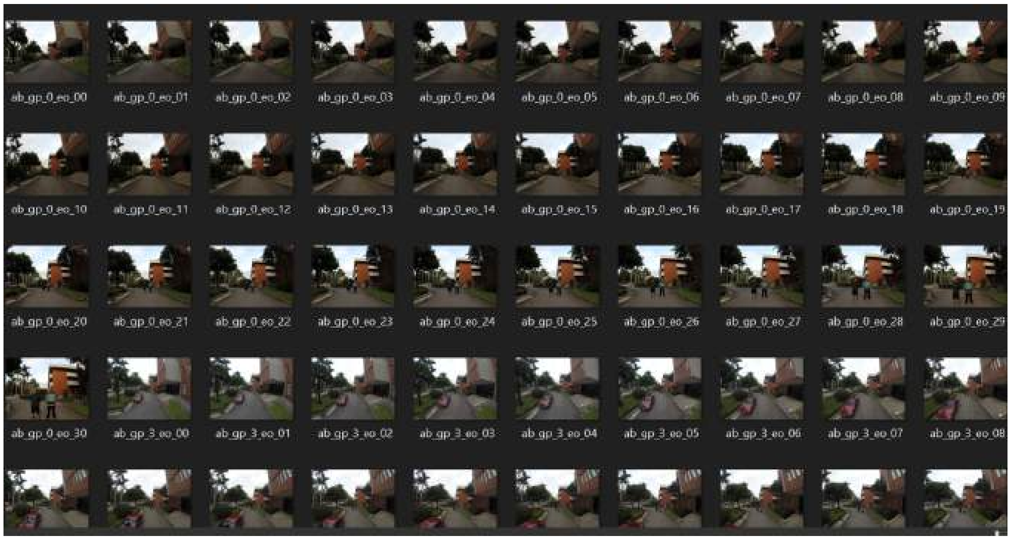

Fig. 4. DnHDronefaces dataset

Fig. 5. Area of Interest of single image

Fig. 6. Area of Interest of multiple image

positives and modifying the bounding boxes to better suit the faces.

Step 3 (Output Network): In this step, the face regions are further refined, and the final bounding boxes and facial landmarks (such as the mouth, nose, and eyes) are produced for alignment.

5.2 Generating Embeddings

Generating embeddings is a crucial step in the face recognition process that involves converting facial images into numerical vector representations. These embeddings compress and meaningfully represent the individual's distinguishing facial features and traits. Deep learning architectures, in particular FaceNet is used to extract and encode the relevant facial traits as part of the process of creating embeddings.Getting a broad and thorough face recognition dataset that includes a variety of identities, positions, lighting settings, and backdrops is the first step in the process.

The goal of this training procedure is to reduce the discrepancy between the predicted embeddings and the actual embeddings associated with each image by iteratively modifying the model's internal parameters, known as weights and biases. The model can be used to process

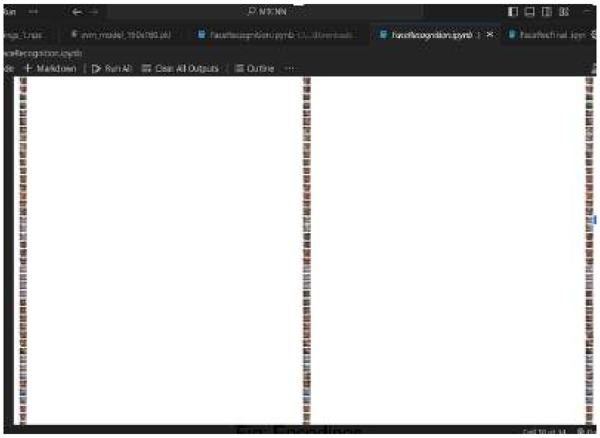

Fig. 7. Embeddings

fresh, previously unviewed facial photos once it has undergone careful training. These photos are fed through the trained CNN during the embedding creation process, which then isolates and condenses the distinguishing facial features into a numerical vector.

This vector, referred to as the embedding, contains a high-dimensional representation of the facial features, with each dimension denoting a recognized feature of the face. The resulting embeddings have a surprising quality that makes similarity comparisons possible: faces of the same person show embeddings that are closer in the vector space.

We may create embeddings of lengths 64, 128, 256, or 512, however we choose to represent the facial feature using a 128-d embedding of size 42. While the greater size will require more computing time, the smaller embedding size might not include all of the facial features. In order to represent the facial feature in our work, 128-d embeddings are used.

Over 3000 drone-shot photos of faces compose the dataset. 80 percent of the photos were used for training and 20 percent were used for testing. Moreover, different epoch operations are included in training loops, where we specify the epoch size and run the training loop repeatedly to improve the model. We repeatedly optimised the training model weights in each epoch while also calculating the accuracy and loss function of the model.

5.3 Using Embeddings to Perform Face Matching

A face image of a person is captured by a drone, which then creates facial encodings and connects with a server to carry out a matching procedure. Calculating the difference between the encoding value of the captured facial image and the encoding values kept in the database is a part of the matching method. Subtraction is chosen because of its lightweight operation and easier than comparison function. Potential matches for the captured facial image are encoded values that are closest to zero after subtraction.

6 Drone Simulation

An innovative approach for simulating drone operations in a controlled virtual environment is drone simulation using the ArduPilot platform. Unmanned aerial vehicles (UAVs) and drones frequently employ the open-source autopilot software package known as ArduPilot. There are many advantages to using ArduPilot for simulation when doing training, testing, and research. Instruments used in simulation involves
a) Dronekit: An open-source framework called Dronekit provides programmers with the tools and APIs they need to interact with and operate drones.
b) Mavproxy: MAVProxy, also known as Micro Air Vehicle Proxy, is a potent command-line tool and ground station software that enables interaction with and management of drones.
c) Mission Planner: An effective and well-liked ground control station (GCS) programme for scheduling, observing, and managing drones is called Mission Planner. It offers a user-friendly graphical interface that enables drone pilots to communicate with their aircraft, organise missions, track telemetry data, and manage several flight-related functions.

6.1 Simulation Setup

The drone's home base is located at latitude 28.3678576, and longitude 77.3168729 using Dronekit. Over a TCP connection, Dronekit and Mavproxy are connected. The UDP connection for

ISSN 2007-9737

the Mission Planner is set up in Mavproxy. The drone's flying route is indicated by waypoints. Each waypoint has a predetermined altitude. The drone has a number of waypoints, each of which has a delay.

6.2 Message Sending and Detection Result

When a face is recognised by the drone, the person's metadata is collected from the "Realtime Database" and displayed in the terminal. The metadata is subsequently sent to "Twillio", a messaging service provider, which sends it to the admin by SMS.

Fig. 8. Drone Simulation Image

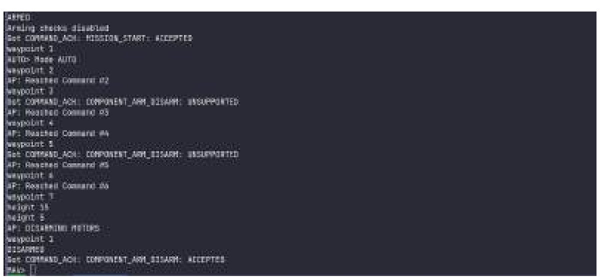

Fig. 9. Simulation Datapoints

7 Simulation Results and Discussion

It offers a thorough examination of multiple tests intended to ascertain the feasibility of our proposition. It also provides further information regarding our experimental setup, training and testing datasets, advantages and disadvantages of our approach, comparisons with other approaches, performance measures, training settings, and performance evaluation for different environmental setups.

Fig. 10. Image and its Metadata

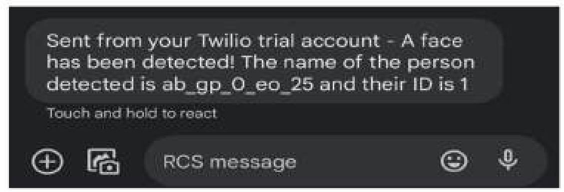

Fig. 11. Twillio sends SMS

7.1 ROC AUC Curve

ROC Curve (Receiver Operating Characteristic Curve): In binary classification, the ROC curve is a graphical representation used to assess how well a machine learning model is performing. At different decision thresholds, it demonstrates the trade-off between a model's true positive rate (sensitivity) and its false positive rate (1 - specificity).

Sensitivity and 1-specificity are plotted on the curve's y-axis and x-axis, respectively. The top-left corner of the ROC curve, which denotes great sensitivity and a low false positive rate, would be present in a perfect classifier. The ROC curve is particularly helpful for comparing various models and is used to evaluate a model's capacity to distinguish between positive and negative example.

AUC (Area Under the Curve): The AUC, evaluates the overall effectiveness of a binary classification model as illustrated by its ROC curve. It calculates the ROC curve's area under the curve, which ranges from 0.5 (which denotes a random

ISSN 2007-9737

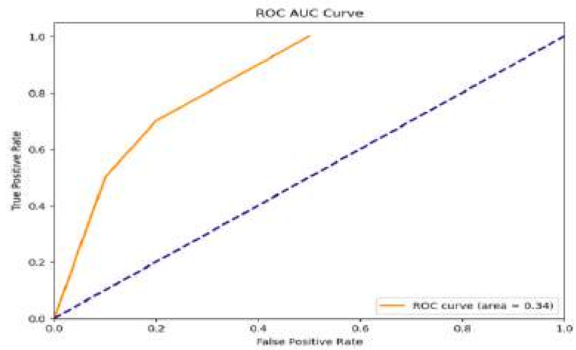

Fig. 12. ROC AUC Curce

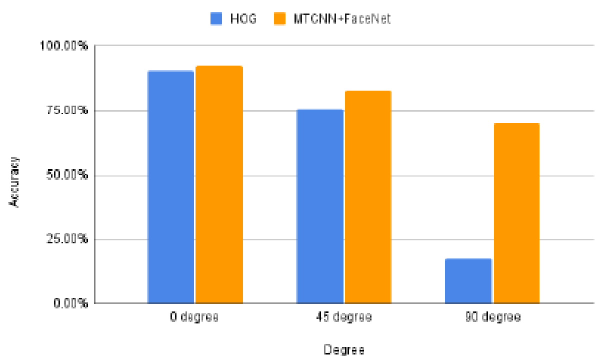

Fig. 13. Accuracy at different angles

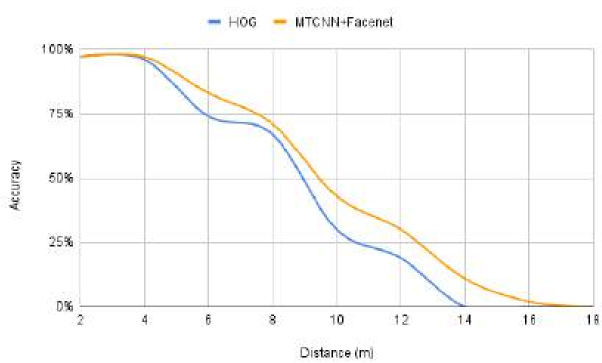

Fig. 14. Accuracy at different distances

classifier) to 1 (which denotes a flawless classifier). A model with an AUC of 1 is perfect, whereas one with an AUC of 0.5 performs no better than random chance. AUC is a useful indicator for comparing and choosing models because a higher AUC denotes better overall model performance. Figure 12 shows the ROC AUC curve of our proposed model which indicates a good model.

7.2 Accuracy at different Angles

Figure 13 shows the accuracy at different angles for both the models. The horizontal axis displays the range of different angles, while the vertical axis indicates the accuracy range. As we can see from the figure, both the HOG and our proposed model provide accuracy at 0 degrees that is considerably above 90 percent. The HOG provides an accuracy of about 75 degree at 45 degrees, while the our model provides an accuracy of more than 80 percent. The accuracy of HOG falls below 20 percent at 90 degrees, however the our model maintains a 70 percent accuracy. Overall, as the angle increases, both the HOG and our model accuracy declines.

7.3 Accuracy at Different Distances

Figure 14 shows the Accuracy at different distances for both the model. The vertical axis provides the range of accuracy starting from 0 percent to 100 percent while the horizontal axis displays the distance in meters starting from 2 m to 18 m.

The accuracy of both our model and the HOG is greater than 95 percent at a distance of 2 to 4 meters. Then, after the accuracy of HOG dropped below 75 percent at a distance of 4 to 6 meters, but the model maintains an accuracy of over 80 percent. Our method's accuracy drops from 80 percent to below 45 percent while HOG's accuracy drops from 75 percent to 30 percent at a distance of 6 to 10 meters. The accuracy of HOG drops from 30 percent to 0 percent at a distance of 10 to 14 meters, whereas our model drops from 45 percent to under 15 percent. The accuracy of the HOG remains at 0 percent from 14 to 18 meters away, and our approach drops from below 15 percent to 0 percent. Overall, both the accuracy decline as the distances increases.

8 Conclusion

In this paper, we offer a method for integrating and MTCNN and FaceNet model in Drone. Furthermore, it enables the system design to link the drone to a database and transfer the operation to the server, thereby enhancing the system's overall performance. Additionally, our approach displays the system's overall accuracy at various heights, angles, and distances.

A message can be sent to the administrator once a person is located and identified using the suggested system, which is useful for the security industry. The administrator can store the name, detection number, and ID of the person to be recognized on the server. Data updates are always possible for the admin. As a result the system can be adapted in many different forms and can be used in many different areas of operation.

The proposed approach has a lot of potential for use in the actual world, especially when it comes to real-time drone operations. While simulations are useful for preliminary testing and development, they frequently fail to accurately capture the subtleties and complexities of real-world surroundings, which are essential for efficient surveillance systems. We can use this model's ability to considerably improve security measures by converting it to real-time drone usage.

References

1. **Cetinkaya, O. T., Sandal, S., Bostancı, E., Güzel, M. S., Osmanoğlu, M., Kanwal, N. (2019).** A fuzzy rule based visual human tracking system for drones. 2019 4th International Conference on Computer Science and Engineering (UBMK), IEEE, pp. 1–6.

2. **Cho, J., Mirzaei, S., Oberg, J., Kastner, R. (2009).** Fpga-based face detection system using haar classifiers. Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Association for Computing Machinery, New York, NY, USA, pp. 103–112. DOI: 10.1145/1508128.1508144.

3. **Dalal, N., Triggs, B. (2005).** Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Ieee, Vol. 1, pp. 886–893.

4. **Daryanavard, H., Harifi, A. (2018).** Implementing face detection system on uav using raspberry pi platform. Electrical Engineering (ICEE), Iranian Conference on, pp. 1720–1723. DOI: 10.1109/ICEE.2018.8472476.

5. **Deeb, A., Roy, K., Edoh, K. D. (2021).** Drone-based face recognition using deep learning. **Hassanien, A. E., Bhatnagar, R., Darwish, A.**, editors, Advanced Machine Learning Technologies and Applications, Springer Singapore, Singapore, pp. 197–206.

6. **Herrera, Diego, Imamura, Hiroki (2019).** Design of facial recognition system implemented in an unmanned aerial vehicle for citizen security in latin america. ITM Web Conf., Vol. 27, pp. 04002. DOI: 10.1051/itmconf/20192704002.

7. **Hsu, H.-J., Chen, K.-T. (2015).** Face recognition on drones: Issues and limitations. Proceedings of the first workshop on micro aerial vehicle networks, systems, and applications for civilian use, pp. 39–44.

8. **Hsu, H.-J., Chen, K.-T. (2017).** Droneface: An open dataset for drone research.

9. **Huang, D.-Y., Chen, C.-H., Chen, T.-Y., Wu, J.-H., Ko, C.-C. (2018).** Real-time face detection using a moving camera. 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 609–614. DOI: 10.1109/WAINA.2018.00153.

10. **Jin, R., Li, H., Pan, J., Ma, W., Lin, J. (2020).** Face recognition based on mtcnn and facenet.

11. **Jose, E., Greeshma, M., Haridas, M. T., Supriya, M. (2019).** Face recognition based surveillance system using facenet and mtcnn on jetson tx2. 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), IEEE, pp. 608–613.

12. **Ku, H., Dong, W. (2020).** Face recognition based on mtcnn and convolutional neural network. Frontiers in Signal Processing, Vol. 4, No. 1, pp. 37–42.

13. **Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G. (2015).** A convolutional neural network cascade for face detection. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5325–5334. DOI: 10.1109/CVPR.2015.7299170.

14. **Patel, H. A., Rajput, R. D. (2018).** Smart surveillance system using histogram of oriented gradients (hog) algorithm and haar cascade algorithm. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1–4. DOI: 10.1109/ICCUBEA.2018.8697464.

15. **Patwary, M. J. A., Parvin, S., Akter, S. (2015).** Significant hog-histogram of oriented gradient feature selection for human detection. International Journal of Computer Applications, Vol. 132, No. 17.

16. **Schroff, F., Kalenichenko, D., Philbin, J. (2015).** Facenet: A unified embedding for face recognition and clustering. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

17. **Sun, Y., Wang, X., Tang, X. (2013).** Hybrid deep learning for face verification. Proceedings of the IEEE international conference on computer vision, pp. 1489–1496.

18. **Viola, P., Jones, M. J. (2004).** Robust real-time face detection. International Journal of Computer Vision, Vol. 57, No. 2, pp. 137–154.

19. **Wang, Y., Huang, L., Li, J., Sun, T. (2023).** Research on face detection based on lightweight mtcnn. 2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Vol. 11, pp. 345–348. DOI: 10.1109/ITAIC58329.2023.10408812.

20. **Zhang, K., Zhang, Z., Li, Z., Qiao, Y. (2016).** Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, Vol. 23, No. 10, pp. 1499–1503. DOI: 10.1109/LSP.2016.2603342.

Article received on 01/09/2024; accepted on 21/08/2025.
**Corresponding author is Wangjam Niranjan Singh.*