A Multimodal Multi-Drone Cooperation System for Real-Time Human Searching

Junkun Peng, Qing Li, *Senior Member, IEEE,* Yuanzheng Tan, Dan Zhao, Jinhua Chen, and Yong Jiang, *Member, IEEE*

Abstract—Aerial images from drones have been used to search individuals in the crowd. However, using a single drone for human searching faces challenges including low accuracy and long latency, due to poor visibility and limited on-board computing resources. In this paper, we propose SkyNet, a multi-drone cooperation system for real-time human searching, including locating and identifying. To locate a person, SkyNet uses *Multi-View Cross Search* with only 2D images. To achieve accurate identification, SkyNet processes faces in images from multi-view drones in three steps. First, a *Multi-Modal Face Correction* is designed to transform less useful face into desired target face, guided by text instructions. Second, an *Angle Masking Network* is developed to minimize invalid data of a single profile face. Third, the multiple face from drones are fused by a *Fusion Weight Network*. Moreover, by predicting the estimated finishing time of tasks, SkyNet schedules and balances workloads among edge devices and the cloud server to minimize processing latency. We implement SkyNet in real life, and evaluate the performance with 20 human participants. The results show that SkyNet can locate people within 0.18m error. The identification accuracy reaches 95.87%, and the system process is completed within 0.84s.

Index Terms—Unmanned Aerial Vehicle (UAV), Person Identification, Multi-Drone Cooperation, Multimodal, Mobile Edge computing.

1 Introduction

Human searching technology, including tracking and identification [1], [2], has been widely used to improve public safety [3], [4], [5]. Existing solutions mainly rely on images captured by fixed-position cameras [6], [7], which have limited field-of-views (FoVs) and are inefficient for tracking moving objects. Benefiting from the wide FoVs and high mobility, drone-based human identification and tracking solutions can be applied in many application scenarios, e.g., military actions and security services [8].

Recent years have witnessed the remarkable success of DNN-based face identification solutions [9], [10], which typically extract a face image as a high-dimensional feature vector and identify the face by the distance between the vectors. These solutions could achieve high accuracy on the premise of sufficient amounts of face pixels. However, a single drone often suffers from the limited face pixels due to its high-flying height and varying drone-person angles, which is revealed in the motivational studies in Section 2.2. DNN-based face identification technologies also consume massive on-board computing resources and bring high latency to the system, which is not ideal for real-time identification.

To track a person in the crowd, most state-of-the-art localization technologies require RGB-D cameras or LiDAR on drones, which are 10 times more expensive than conventional 2D cameras. Moreover, in outdoor and long-distance scenarios, the point cloud density of RGB-D cameras and LiDAR drop dramatically, leading to poor localization accuracy and a small working area [11], [12].

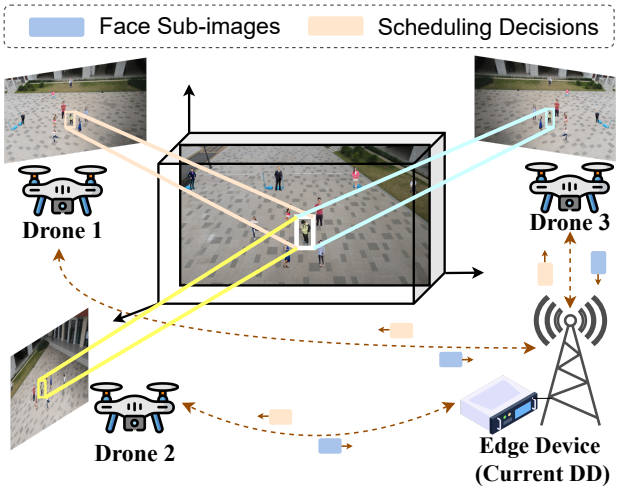


Figure 1: SkyNet uses multiple drones and edge devices to identify and locate the target person.

In this paper, we design SkyNet, a multi-drone cooperation system to achieve accurate and real-time human searching, including localization and identification. As shown in Figure 1, at a shooting instant, each drone takes a picture of the crowd and uses on-board person/face detection models to detect face sub-images, which are sent to the designated device (DD) selected from edges and the cloud server. Then the DD computes the 3D location of each face and generates identification results by exploiting the multi-view images.

To achieve SkyNet's high-level design goals, several challenges need to be addressed. i) How to find a person's 3D position and align his/her face sub-images from mul-

- *Junkun Peng, Yuanzheng Tan, and Yong Jiang are with the Shenzhen International Graduate School, Tsinghua University, Shenzhen, Guangdong 518055, China, and also with Peng Cheng Laboratory, Shenzhen, Guangdong 518055, China (e-mail: pjk23@mails.tsinghua.edu.cn, tanyz23@mails.tsinghua.edu.cn, jiangy@sz.tsinghua.edu.cn).*
- *Qing Li and Dan Zhao are with Peng Cheng Laboratory, Shenzhen, Guangdong 518055, China (e-mail: liq@pcl.ac.cn, zhaod01@pcl.ac.cn).*
- *Jinhua Chen is with Sun Yat-Sen University, Shenzhen, Guangdong 518107, China (e-mail: chenjh537@mail2.sysu.edu.cn).*

Manuscript received xxx. (Corresponding author: Qing Li.)

tiple views when there are only 2D images of him/her in different views? In fact, forming a unified perception of the scene is a common challenge faced by multi-drone/robot cooperation. ii) How to enhance the legibility of face features extracted from the face images captured by drones? iii) How to take advantage of the face information offered by multi-view images for much more accurate identification? iv) With the system taking pictures continuously, how to select the suitable DD to ensure fast processing?

To address these challenges, in SkyNet, we propose a *Multi-View Cross Search Algorithm* to find the 3D real-world location of a face and align his/her multiple 2D sub-images. To enhance the legibility of face features in drone-view images, we design a *Multi-Modal Face Correction*, which synergistically leverages the strengths of language-visual models and face feature extraction models to transfer the less illegible face feature to the legible one under the guidance of text. To further improve identification accuracy, we propose a *Mulit-View Fusion Identification*, which is a two-stage fusion face identification pipeline that contains: i) an *Angle Masking Network (AMN)* that can output mask weight to prioritize valid information and minimize invalid data of a single profile face. ii) a *Fusion Weight Network (FWN)* that can generate fusion weight for a person's multiple faces from different FoVs, based on which these sub-images are fused for the final inference result. Moreover, we propose a *Dynamic Task Scheduling Algorithm* to balance workloads over consecutive shooting instants and reduce latency.

To evaluate SkyNet, we deploy SkyNet in real life on four drones, three edge devices and a cloud server. We not only test SkyNet with datasets about drone-based face identification, but also conduct real-world experiments with 20 volunteers and obtain their consent. The evaluation results show that SkyNet achieves 95.87% accuracy and the real-time latency of localization and identification (within 0.84s).

The key contributions of this paper are as follows.

- We design a multi-drone cooperation system for accurate and real-time identification and localization.
- We design a novel pipeline for drone-view face identification, including multi-modal face correction, angle-based single face masking, and legibility-based multiple faces fusion. The pipeline also reduces processing latency by multi-view parallel computing.
- SkyNet significantly reduces the latency of task processing to achieve the real-time execution through the cooperation of heterogeneous devices and dynamic task scheduling.

2 Background and Motivation

2.1 Background

With the rapid development of smart cities, biometric identification [13], [14], especially **Face Identification**, is becoming an important basic service in scenarios such as security systems, criminal investigations, and human-computer interaction [15], [16], [17]. Face identification solutions are mainly applied to fixed-position cameras, which only have small fields of view and are easily obstructed. Therefore, such solutions are infeasible for outdoor scenes such as searching for a target person in a crowd, where the target moves frequently and the identification range is extensive. Recently, mature and affordable drones have brought new opportunities for face identification in broader application scenarios, thanks to their high maneuverability, wide FoVs, and easy deployment. Face identification based on drones is playing an irreplaceable role in security surveillance, daily patrols, finding specific people on the street, and many other scenarios. However, drones also have many challenges in face identification, which are discussed in Section 2.2.

Image-based Localization is a widely demanded service in many scenarios such as robotics, SLAM, security, and smart city [18], [19], [20], [21]. The traditional image-based solutions such as object detection and tracking use a bounding box to point out the target, which cannot provide the 3D real-world coordinates [22], [23], [24], [25], [26]. Some RGB image-based 3D reconstruction solutions [27], [28], [29] have the localization feature. But they suffer from high complexity because of dense matching [30] and take a long time to adjust when the environment changes. Several image-based 3D localization solutions using RGB-D cameras or LiDAR cameras have been proposed [31], [32], [33]. However, these special cameras cannot obtain accurate and consistent 3D data in the immense outdoor [11], [12], [34]. Moreover, these cameras have expensive prices and low resolution. For example, Intel RGB-D D455 [11] ($399) only provides a resolution of 1280×720 at an effective operation range under 4 meters. Intel LiDAR L515 [35] ($589) works at indoor only, with a resolution of 1920×1080. None of them are suitable for localization in drone-based solutions.

Enhancing the accuracy of vision recognition represents an ultimate objective that the research community persistently pursues. Recent advancements have seen the adoption of **Multimodal Models** [36], which leverage additional dimensions of information, such as textual content, to augment the process of vision recognition. The Contrastive Language-Image Pre-training (CLIP) model [37] is among the most representative multimodal models, which utilizes a text encoder and an image encoder to map the text and image into a shared vision-language embedding space. CLIP could identify the most suitable text description for an image due to the contrastive learning on a vast dataset comprising 400 million image-text pairs. Encouraged by the remarkable advancements of CLIP, several additional studies have emerged that harness CLIP for text-driven image processing, e.g., style transfer [38], instruction-controlled NeRF [39], and face attribute hallucination [40]. The given work indicates that the challenges presented by drone vision could potentially be mitigated by the guidance of text-based instructions.

2.2 Motivational Studies

We comprehensively analyze the impacts of the drone view on human identification accuracy and computational latency using a single drone, include flight height, drone-person angle and resolution. We use the NVIDIA Jetson Xavier NX [41] as the edge device to process the capturing images of the drone. The identification pipeline consists of RetinaFace [42] for face detection, and ResNet-based ArcFace [1] for face identification. We use two model configurations, RetinaFace-2.5Gf & ResNet18 and RetinaFace-10Gf

(a) Impacts of drone height on identification accuracy.

(b) Impacts of drone-person angle on identification accuracy.

(c) Impacts of resolution on identification accuracy.

(d) Impacts of resolution on computational latency.

(e) Impacts of resolution on transmission latency.

Figure 2: Identification service provided by a single drone.

& ResNet50, corresponding to the light model configuration and the heavy model configuration, respectively. We collect 567 drone images with the DJI Mavic [43] at various distances, heights, drone-person angles, and resolutions. By using these aerial images, we mainly analyze the impacts of flight height, drone-person angle, and image resolution on human identification accuracy and computational latency.

The high-flying height offers a broad view scope but degrades the face identification accuracy. Figure 2a presents the accuracy achieved at different heights. The horizontal distance between the drone and the person to be identified is fixed at 20 meters, and the images used are in 4K resolution. We down sample these images according to the needs of the experiment. As the drone flies higher, the identification accuracy decreases significantly in both model configurations. The accuracy is less than 10% at the height of 25m, since the over-tilted faces in high-flying drone-view make fewer effective pixels are available for the person.

The large drone-person angle results in low identification accuracy. Figure 2b illustrates the accuracy achieved at different drone-person angles (0°: the drone exactly faces the frontal face of the person, and 180°: the drone exactly faces the back of the head of the person). In this experiment, the height is 5 meters, the horizontal distance is 10 meters and the resolution is 4K. The result shows that as the drone-person angle increases, the accuracy drops significantly and nearly reaches zero at around 150°.

The high image resolution boosts the accuracy but incurs the high computational latency. Figure 2c and 2d show the identification accuracy and computational latency under 9 photo resolution settings, i.e., 4000×3000, 2666×2000, 2000×1000, 1600×1200, 1333×1000, 1142×857, 1000×750, 888×400, 800×600. For both model configurations, the better resolution leads to the higher accuracy, while causing significantly the higher computational latency.

The high image resolution impedes the transmission but can be accelerated by transmitting only face images. Figure 2e show the transmission latency under 9 photo resolution settings. The higher resolution causes significantly the higher transmission latency in transmitting full image. Due to the removal of redundant scene information, the latency of transmitting face images under different resolutions is much lower than that of full image transmission.

According to the above results, the higher the flight height, the farther the distance between the drone and the target person, and the steeper the drone-person angle, resulting in fewer effective pixels available, which in turn leads to lower accuracy. Furthermore, higher image resolutions lead to larger transmission and inference latency, but lower resolutions reduce the accuracy of face identification.

To this end, we gain two valuable insights from our motivational experiment: i) Using multiple drones instead of a single drone could be an alternative solution for human identification, offering varying drone heights and angles for improved accuracy. ii) Using powerful edge devices to assist drones can solve the issue of limited computing power, but prior on-board processing is necessary to minimize transmission latency, such as detecting faces on the drone and transmitting them to the edge device for identification.

3 SkyNet: Overview

We build a multi-drone cooperation system to locate and identify the target person in a crowd, which consists of:

i) a group of drones with heterogeneous computing capabilities that can capture images from different angles, run lightweight models, and offload computation tasks to edge devices and the cloud server.

ii) a group of interconnected edge devices, closer to drones, can run models efficiently but with finite total computing power. In SkyNet, a stable and power edge device is selected as the home edge, which can collect status information from other edges and schedule tasks.

iii) the cloud server, farther from drones and edge devices, but with sufficient computing power. When the workload of all edge devices is too heavy, the computation can be offloaded to it.

Figure 3 shows the operational flow of SkyNet. In a crowded scene, SkyNet locates the target using the Multi-Drone Person Localization (MDPL) module, and identifies he/she by the Multi-View Fusion Identification (MVFI) and the Multi-Modal Face Correction (MMFC) modules. The Dynamic Task Scheduling for Heterogeneous Devices (DTSH) module schedules tasks among edge devices and the cloud server to reduce latency and balance workload.

i) **Multi-Drone Person Localization (MDPL)**. On each drone, aerial images of the crowd are first processed onboard by a person detection model and a face detection model. These extracts face sub-images with their bounding box locations and face landmarks, and sends them to the DD selected via the DTSH module. Then SkyNet runs the MDPL module on the DD to locate each person's real-world location through a series of coordinate matrix transformations and align his/her sub-images from multiple views.

ii) **Multi-View Fusion Identification (MVFI)**. After aligning the multi-view sub-images of a person, SkyNet runs the MVFI module on the DD. The MVFI first uses

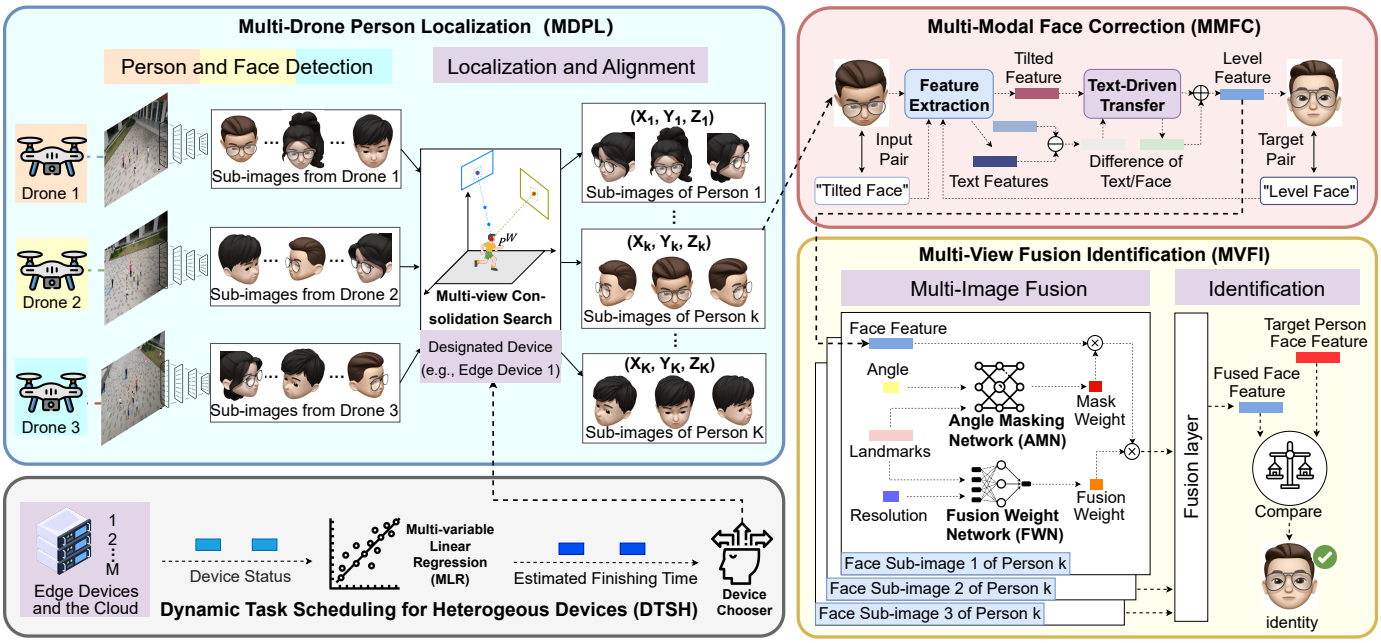


Figure 3: Operational Flow of SkyNet in a nutshell.

MMFC to extract face features from all face sub-images captured at different angles. Second, to reduce the invalid information in the face feature, the AMN conducts a mask weight for each face sub-image to mask unimportant channels in the face feature. Third, to comprehensively utilize features from these different angles, the FWN assigns a fusion weight to each sub-image to reflect its legibility. Then, the fusion layer generates the fused face feature by combining face features from different views using fusion weights. The fused face feature contains more details than those extracted from a single image. Finally, the fused face feature is compared with the facial feature of the target person through calculating their feature distance. A person whose feature distance is within the threshold is considered the target person to be found.

iii) **Multi-Modal Face Correction (MMFC)**. SkyNet runs the MMFC module to extract face features and enhance their legibility. To achieve this goal, MMFC tries to correct the illegible face features to the legible face features. Take the tilted face for example, for an input face image, MMFC matches it an input text and a target text, e.g., the 'tilted face' and 'level face' in Figure 3. MMFC first extracts the features of image and texts by multiple models. Then, by text-driven transfer, MMFC transfers the difference between input text feature and target text feature to the difference of the face features, which can be used to correct the input face feature to the target face feature that is more legible.

iv) **Dynamic Task Scheduling for Heterogeneous Device (DTSH)**. We define the entire operation flow of localization and identification as a *task*. At a shooting instant, each drone takes a photo of the crowd. The detection part of one task, i.e., face sub-image extraction in MDPL, is executed in parallel on each drone. The remaining parts of one task need to use images from all views simultaneously and thus can only be done on one device, i.e., the DD. Therefore, the remaining parts of one task are called the DD-side task, including the localization sub-task, the fusion sub-task, and the identification sub-task. In order to balance the workload of edge devices and ultimately reduce the processing latency, selecting a suitable DD for one DD-side task is the key. For this purpose, the home edge first predicts the Estimated Finishing Time (EFT) required by each edge device if it handles this DD-side task. Then, it selects the device with the shortest EFT as the DD for this task. If all the EFTs of edge devices are higher than the shooting interval (e.g., 1s), it selects the cloud server as the DD of this task. Finally, scheduling decisions are sent to all drones, which offloads data to the DD.

4 Multi-Drone Person Localization

In this section, we present the process of multi-drone person localization. First, the face sub-image extraction model is designed. Then, a multi-view cross localization and alignment strategy is proposed to find the 3D position of each person and align his/her sub-images from different views.

4.1 Face Sub-images Extraction

The first step of the MDPL module is to extract face sub-images, their bounding box locations, and face landmarks on each drone. Because drones are far from the crowd, some faces may be too small to be recognized in the image. We first use a person detection network to find the full body of each person on images. Then, we use a face detection network to detect each person's face.

Considering the limited computational and power resources of drones, we choose YOLOX-Tiny detector [44] as the person detection network and RetinaFace detector [42] as the face detection network because of their accurate detection rate [45], [46] and fast processing speed [46], [47]. We fine-tune the person detection network using datasets from drones (VisDrone [48]) to better detect pedestrians at small scales in drone images. The person detection network uses the DNN to generate the bounding box position of

each person in the original image, and the face detection generates the bounding box position and facial landmark position of each face. Next, on each drone, we can get the bounding box location of each face and its face landmarks, and then extract the face sub-images. Each drone then sends this information to the DD, which is selected by the DTSH module described in Section 7 for further processing, namely localization, alignment, feature fusion and identification.

4.2 Multi-View Cross Localization and Alignment

Face sub-images and bounding box locations from multiple views are aggregated in the DD, but it is unclear which sub-images from multiple drones belong to the same person. In this subsection, we propose a *multi-view cross search algorithm* to determine a person's 3D real-world location and align face sub-images of the same person from different drones by the location. Initially, we randomly select a drone D_i and use its view as the *primary view*. Denote the center point of the face in the pixel coordinates of the drone D_i as $P_i = [x_i, y_i]^T$. Our goal is to find the 3D real-world position of this face to align face sub-images of the same person from different drones, denoted as $P^W = [x^W, y^W, z^W]^T$. In order to find the 3D real-world position of this face, we first establish the transformation relationship between P_i and P^W, as follows:

$$z_i^C \begin{bmatrix} P_i \\ 1 \end{bmatrix} = \begin{bmatrix} C_i & \overrightarrow{0} \end{bmatrix} \begin{bmatrix} R_i & T_i \\ \overrightarrow{0} & 1 \end{bmatrix} \begin{bmatrix} P^W \\ 1 \end{bmatrix}, \quad (1)$$

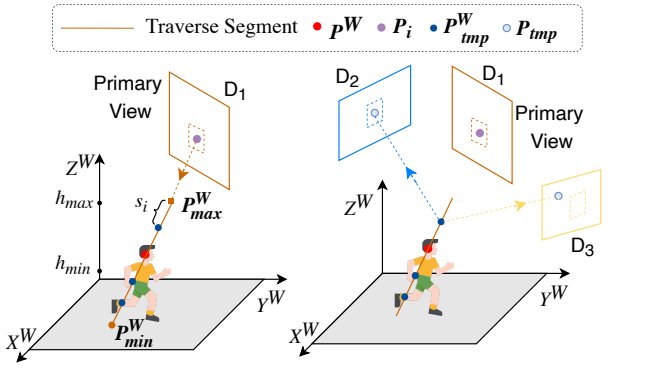
where $C_i \in \mathbb{R}^{3\times3}$ denotes the *camera internal matrix* of the drone D_i, which can establish the mapping relationship between the image pixel coordinates and the camera coordinates for D_i. $R_i \in \mathbb{R}^{3\times3}$ and $T_i \in \mathbb{R}^{3\times1}$ represent the *rotation matrix* and *translation matrix* of the drone D_i, which could establish the mapping relationship between the camera coordinates and the world coordinates for D_i. R_i and T_i can be obtained by the Perspective-N-Points (PNP) positioning[1] [49] and C_i can be obtained by the camera calibration technique [50]. Following (1), P_W is given as:

$$P^W = R_i^{-1}\left[z_i^C C_i^{-1} P_i - T_i\right]. \quad (2)$$

Due to the absence of depth information z_i^C, P^W is still a variable depending on z_i^C. Hence, we find the final piece of the puzzle by exploiting information provided by the views of other drones.

By varying z_i^C in (2), P^W forms a line in 3D space, called the inverse line, denoted as l. Without any other additional information about z_i^C, we traverse the points on l using an adaptive stride. We consider the highest and lowest possible heights of a person in the real world, denoted as $h_{\max}$ and $h_{\min}$, and get the corresponding real-world locations $P^W_{\max}$ and $P^W_{\min}$ by these two heights. The z_i^C corresponds to $P^W_{\max}$ and $P^W_{\min}$, denoted as $z_{\max}$ and $z_{\min}$, respectively, can be calculated according to (1). By restricting the inverse line l with $z_{\max}$ and $z_{\min}$, the search space l is further narrowed down to a line segment, named the *traverse segment*. The traverse stride of z_i^C is set as $s_i = \frac{z_{\max}-z_{\min}}{(h_{\max}-h_{\min})/s^W}$, where s^W is the fixed real-world height stride (e.g., 10 cm).

1. As SkyNet is designed to locate a person in a known space, it is feasible to get known calibration points.



(a) Space inverse transformation. (b) Cross check for a traverse point.

Figure 4: Multi-View Cross Localization.

Algorithm 1. Multi-view Cross Search Algorithm

Input: i, $h_{\max}$, $h_{\min}$, N, s^W, P_i, for each drone:[R, T, C]

$l \leftarrow getInverseLine(P_i)$
$z_{\max} \leftarrow$ set Z^W of P^W in (2) to $h_{\max}$
$z_{\min} \leftarrow$ set Z^W of P^W in (2) to $h_{\min}$
$z_i^C = z_{\min}$, $s_i = \frac{z_{\max}-z_{\min}}{(h_{\max}-h_{\min})/s^W}$
while $z_i^C \leq z_{\max}$ **do**
  $P^W_{tmp} \leftarrow pixel2realworld(z_i^C, P_i)$
  $result$ = true
  for k in [1,N]\{i} **do**
    $P_k = realworld2pixel_{\text{view k}}(P^W_{tmp})$
    if $!pointInBoundingBox(P_k)$ **then**
      $result$ = false
    end if
  end for
  if $result$ is true **then**
    $P^W \leftarrow P^W_{tmp}$
    break;
  end if
  $z_i^C \leftarrow z_i^C + s_i$;
end while

As shown in Figure 4, for each traverse point P^W_{tmp}, we conduct a cross-check by projecting it onto all other drones' views using (1). If, on each view, the projected point falls into a face bounding box, we assert P^W_{tmp} is the true point P^W, i.e., the 3D real-world position of the face. Note that in this process of cross-checking, the corresponding sub-images of the person in all the different views are found, effectively accomplishing the alignment. The pseudo code of the algorithm is shown in Algorithm 1.

5 Multi-Modal Face Correction

Traditional face identification like ArcFace [1] extracts face features from face sub-images and identifies these features by distance comparison. However, as shown in Section 2.2, the faces captured in drone-view typically suffer from reduced effective pixels due to issues like tilt and blurring, which ultimately diminish legibility of face features. Furthermore, variations in facial attributes due to factors such as aging, changes in hairstyle, and the use of glasses or

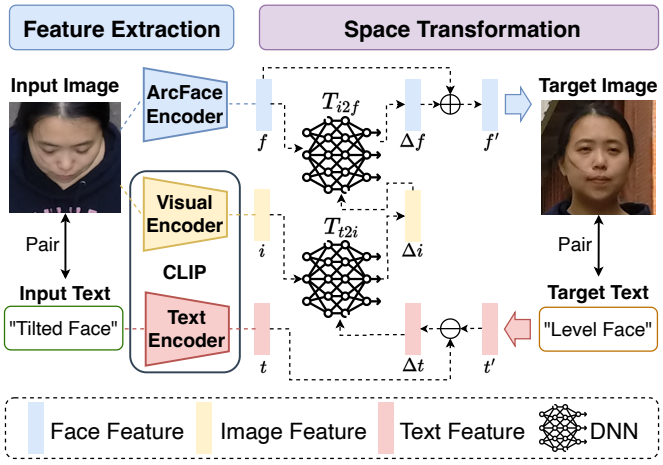


Figure 5: Multi-Modal Face Correction. The face images used are from the DroneFace dataset [51].

accessories, also contribute to the increased complexity of face identification. To overcome these obstacles and improve face identification accuracy, we propose Multi-Modal Face Correction (MMFC) to transform a less legible input face feature into a more discernible face feature. MMFC utilizes text prompts to provide additional information, thereby aiding the transformation process.

Specifically, MMFC processes a face sub-image alongside an input text of the input face's key attributes of concern. It then outputs a target face feature that aligns with a specified target text, which outlines the desired attributes of a target image. For example, the input text can be 'A photo of a male/female taken at a tilted angle', and the target text can be 'A photo of a male/female taken at a level angle'. Various text prompts can achieve the hallucination of different attributes of the face [40] for easier face identification, e.g., tilted face, age, hairstyle, etc. Note that the text prompts are fixed in the inference process. For further details, please refer to our supp. Next, we use tilted face as an example and elaborate on the pipeline and the training process of MMFC.

5.1 Pipeline of MMFC

MMFC begins by extracting features from the sub-image of the face, upon which it then performs the transformation. As shown in Figure 5, the pipeline of MMFC consists of two stages: feature extraction and space transformation.

5.1.1 Feature Extraction

Although existing multi-modal models, such as CLIP [1], have shown significant success in general visual semantic description, they fall short in extracting pivotal features specifically for face identification. Contrary to this, traditional face identification methods like ArcFace [1] can discover key identity features but are often influenced by the clarity of the face.

Hence, to exploit the benefit of multi-modal guidance while mining facial feature to the largest extent, MMFC uses two encoders to extract features from the input image, i.e., an ArcFace Encoder [1] and a text-aligned Visual Encoder from CLIP [37]. The ResNet50-based ArcFace Encoder [1] extracts a face feature $f \in \mathbb{R}^{1\times512}$ in the face identification space from the image x. The text-aligned Visual Encoder from CLIP [37] is used to extract an image feature $i \in \mathbb{R}^{1\times512}$ in the vision-language space from the image x. The Text Encoder from CLIP [37] is used to extract text features $t \in \mathbb{R}^{1\times512}$ from the input text in the vision-language space. A same Text Encoder is used to extract the text feature t' from the target text. It is worth noting that, since CLIP can align images with corresponding text, the cosine similarity between i and t is higher than that between i and t'.

5.1.2 Space Transformation

We expect to transform the input face feature f into a target face feature f' that matches the target text feature t'. MMFC leverages the text features to guide this transformation, i.e., converts the difference of text features $\Delta t = t - t'$ to the difference of face features $\Delta f = f - f'$, which is then used to obtain the target face feature f'. However, the transformation from Δt to Δf is not straightforward, as they are obtained for fundamentally different objectives under different feature spaces.

In MMFC, we use Δi to serve as a pivotal intermediary to the conversion from Δt to Δf through a dual-phase space transformation process. On the one hand, there exists an underlying connection between Δt and Δi that can be effectively modeled since CLIP aligns t and i. On the other hand, since i and f are extracted from the same person's image, there also exist an inherent relationship between Δi and Δf.

First, MMFC transforms the difference of text features Δt into the difference of image features in vision-language space Δi. The transformation is defined as follows:

$$\Delta i = T_{t2i}(i, \Delta t). \tag{3}$$

This transformation also incorporates the image feature i as input. This is because the transformation between Δt and Δi is not one-to-one mapping. Δt guides the transfer direction, and i offers a starting point for the transformation.

Second, we transform the difference of image features in vision-language space Δi into the difference of face features in face identification space Δf. The transformation is defined as follows:

$$\Delta f = T_{i2f}(f, \Delta i). \tag{4}$$

Same as (3), Δi guides the transfer direction, and f offers a starting point for the transformation. At this point, we obtain the difference Δf from the input face feature to the target face feature.

Last, we can obtain the level face features f' by

$$f' = f - \Delta f. \tag{5}$$

The justification of transformation's rationality is provided in [37], [40]. For further details, please refer to our supp.

5.2 Training of MMFC

For the two transformation in (3) and (4), we use two MLPs as function approximators. For T_{t2i}, we use the following loss function for supervised learning [40]:

$$L_{t2i} = 1 - <\Delta t, \Delta i> + 1 - <t', i - \Delta i>, \tag{6}$$

where $<\cdot,\cdot>$ is the cosine similarity that used in CLIP. Benefiting from the zero-shot vision-language representation

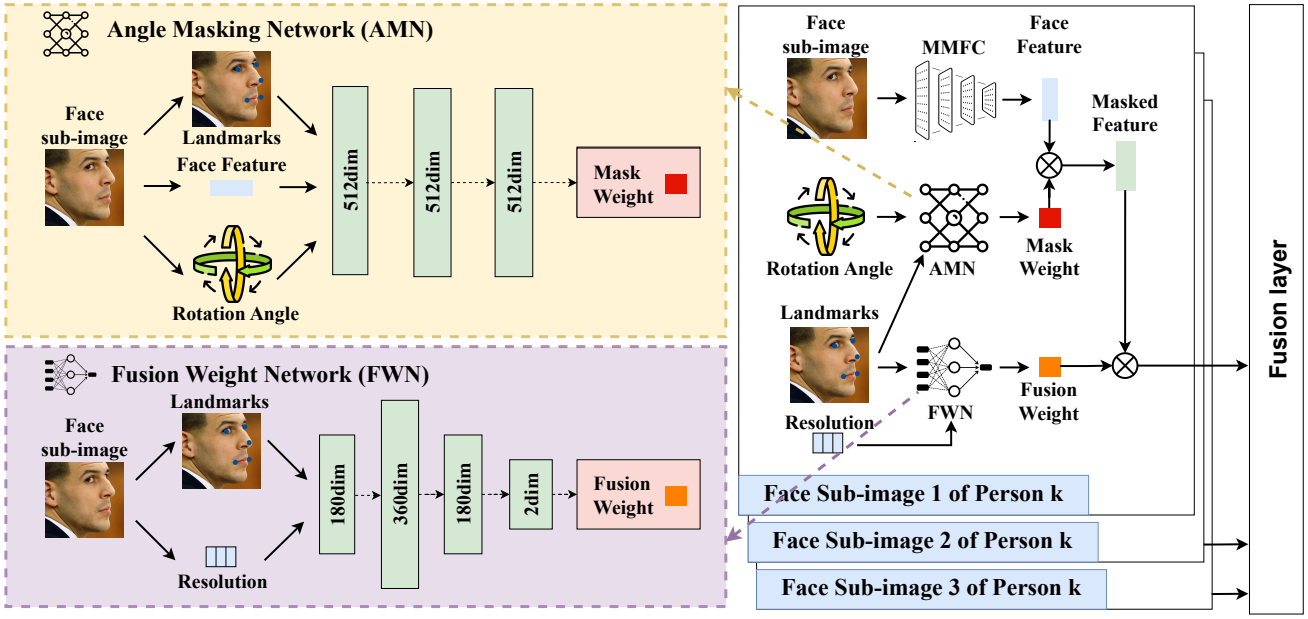


Figure 6: Multi-View Fusion Identification workflow.

capability of CLIP, we can eliminate the need for large-scale annotated data for training.

For T_{i2f}, we define the following loss function for supervised learning, which is an enhanced version of the loss function defined in [40]:

$$L_{i2f} = 1 - <f + \Delta f, \frac{1}{N}\sum_{i=1}^{N} f_i'>, \tag{7}$$

where $\frac{1}{N}\sum_{i=1}^{N} f_i'$ represents the class center of f', i.e., a series of face features that extracted from current person' images taken at level angles. The original loss function $L_{i2f} = 1 - <f + \Delta f, f'>$, as defined in [40], struggles when training on a small dataset since this loss function compromises the integrity of the original face identification space, leading to inaccurate face identification. The enhanced loss function, as specified in (7), addresses this issue by maintaining the facial features within the face identification space pertinent to the individual in question, thereby ensuring accurate identification.

6 Multi-View Fusion Identification

In this section, we detail our proposed multi-view fusion identification pipeline, including single-view face feature masking, multi-view face feature fusion, and a parallel computing strategy aimed at speeding up processing.

6.1 Face Feature Extraction

For each face sub-image, we use our designed MMFC detailed in Section 5 to extract a 512-dimensional feature vector in the face identification space, denoted as $\phi(\cdot) \in \mathbb{R}^{1\times512}$, which has small inner-class distances and large inter-class distances that can be used for identification.

6.2 Single-view Face Feature Masking

As shown in Section 2.2, drone-person angle can seriously affect the accuracy of face identification due to the invalid information about the invisible half of the face in profile images. In SkyNet, we generate weights for all dimensions of the face feature to emphasize valid information while downplaying the invalid data within the feature.

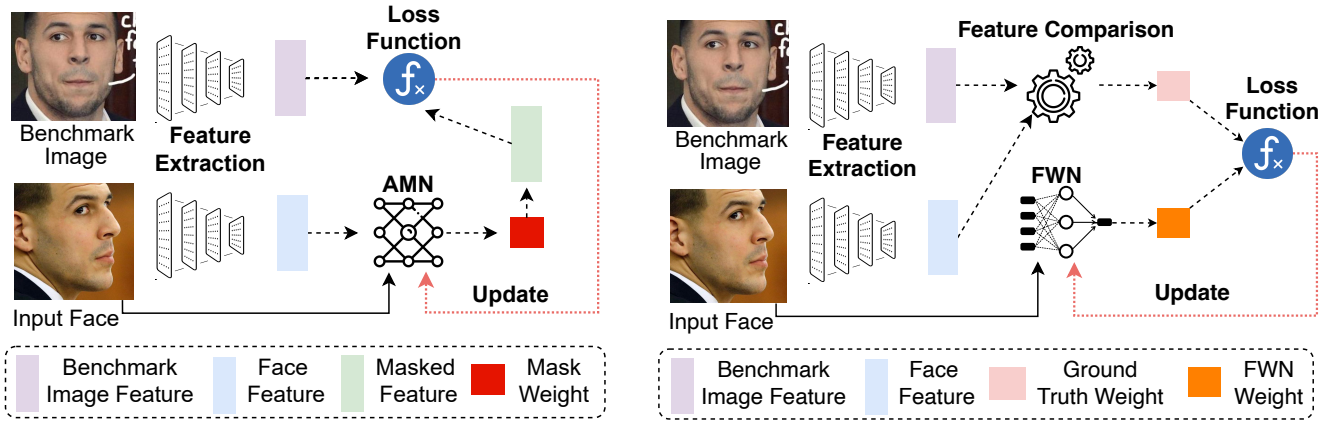
6.2.1 Angle Masking Network

Our goal is to generate weights for all dimensions of the face feature to emphasize valid information while downplaying the invalid data within the feature. Inspired by our motivational study in Section 2.2, we aim at establishing the relationship between drone-person angle and face feature to generate 512-dimensional mask weights to achieve our goal.

For this purpose, we design an Angle Masking Network (AMN), which generates a mask weight $Ma \in \mathbb{R}^{1\times512}$ from the given single face feature, the landmarks of this face, as well as the rotation angles of this face. The face landmarks can be retrieved from the face detector, which can also provide the face bounding box. The calculation of rotation angle is detailed in Section 6.2.2. The value of each dimension in the mask weight Ma ranges from 0 to 1, i.e., $Ma_i \in [0,1], i = \{1, ..., 512\}$. AMN explores the hidden relationship between face features and rotation angles using supervised learning. It emphasizes valid dimensions while downplaying invalid dimensions in the face features through the dynamic mask weight in order to improve identification accuracy. Intuitively, the dimensions displaying significant variations between the profile face features and the frontal face features should be masked, while the others should be retained. To this end, the *mask loss* for training AMN is defined as follows:

$$L_{mask} = \frac{1}{N}\sum_{1}^{N}\left(||Ma\cdot\phi(Face) - \phi(Face_{bm})||^2\right), \tag{8}$$

where N is the number of samples in a training batch, $Face$ is the input image of the person, and $Face_{bm}$ is



(a) Training process of Angle Masking Network.

(b) Training process of Fusion Weight Network.

Figure 7: Multi-View Fusion Identification. The face images used are from the CFP dataset [52].

the benchmark image of the person, i.e., a frontal image with sufficient effective pixel. By minimizing the L_{mask}, an effective AMN can be trained to select the dimensions with invalid information in face features by outputted mask weight, as shown in Figure 7a. Using the trained AMN, we can generate the mask weight for a person's face sub-image from a single view, then emphasize valid information while attenuating the invalid data within the feature.

6.2.2 Rotation Angle Calculation

To obtain the rotation angles, we estimate the three-axis rotation $[Yaw, Pitch, Roll]^T \in \mathbb{R}^{1\times 3}$ of the face through the face landmarks and face bounding box.

The yaw angle is calculated as follows:

$$Yaw = arcsin\left[\frac{2\times(x_{nose}-x_{center})}{w}\right], \tag{9}$$

where x_{nose} and x_{center} denote the x-coordinate of the nose and the face center in the face sub-image, respectively, and w denotes the width of a person's face sub-image.

The pitch angle is calculated as follows:

$$Pitch = arcsin\left[\frac{2\times(y_{nose}-y_{center})}{h}\right], \tag{10}$$

where y_{nose} and y_{center} denote the y-coordinate of the nose and the face center in the face sub-image, respectively, and h denotes the height of a person's face sub-image.

The roll angle is calculated as follows:

$$Roll = arctan\left(\frac{y_{re}-y_{le}}{x_{re}-x_{le}}\right), \tag{11}$$

where x_{le} and x_{re} denote the x-coordinate of the left eye and right eye in the face sub-image, respectively, and y_{le} and y_{re} denote the y-coordinate of the left eye and right eye in the face sub-image, respectively.

Although the above estimation may not be entirely precise, as the facial landmarks utilized for calculation are 2D image points rather than 3D real-world coordinates, the angle information implicitly conveyed by the face landmarks can act as a compensatory measure. As such, both the estimated rotation angles and the facial landmarks are input into the AMN.

6.3 Multi-view Face Feature Fusion

After processing the face features from all angles using AMN, they must then be compared with the face features of the target person. Traditional face identification algorithm compares extracted feature vectors from different angles with the target person's face features one by one and selects the vector which has minimum distance from the target person's face features as the optimal vector for identity. However, such an approach ultimately still obtains identification results based on the feature vector of one view/angle, while the information from other angles' feature vectors is wasted. In SkyNet, we fuse face features from different angles according to weights generated by a Fusion Weight Network for more accurate identification.

6.3.1 Fusion Weight Network

As demonstrated in Section 2.2, face angle and resolution can seriously affect the accuracy of face identification. In order to establish a reasonable fusion method for images of different angles and qualities, we design the FWN. As shown in Figure 6, for each face image, the FWN takes its resolution and landmarks of face features as input, and outputs the fusion weight of the image. This is achieved by neural networks capturing the hidden relationships between landmark vectors, resolution, and image legibility. Intuitively, face images that are easier to identify should be given higher fusion weights. To this end, *weight loss* for training FWN is defined as follows:

$$L_{weight} = \frac{1}{N}\sum_{1}^{N}\left(y - \frac{d}{||\phi(Face)-\phi(Face_{bm})||^2}\right), \tag{12}$$

where N is the number of samples in a training batch, y denotes the fusion weight of an input image generated by the FWN, d denotes the dimension of the feature vector, $Face$ and $Face_{bm}$ are the input image and the benchmark image (i.e., a frontal view image with sufficient effective pixel) of the person, respectively, and $\phi(\cdot)$ denotes the d-dimensional face feature vector extracted by face feature extraction. $\frac{d}{||\phi(Face)-\phi(Face_{bm})||^2}$ is also called ground truth weight. By minimizing the weight loss L_{weight}, an effective FWN can be trained to output fusion weights that can accurately reflect the legibility of each image, as shown in

Figure 7b. Using the trained FWN, we can generate the corresponding fusion weights for the same person's face sub-images from different views, which are then normalized and fed into the fusion layer.

6.3.2 Fusion Layer and Identification Result

For each person, the MVFI first uses the fusion layer to fuse his/her face sub-image features from different views to obtain a fusion feature, and then calculates the distance between it and the target person's face feature. After getting the feature distances between each person and the target person, the minimum feature distance is compared with the feature distance threshold, and if the minimum feature distance is lower than the threshold, its corresponding person is identified as the target person to be found.

6.4 Parallel Heterogeneous Computing

To guarantee the real-time performance, we design a Parallel Heterogeneous Computing (PHC) strategy. The PHC sets up the thread pool to prepare for upcoming tasks. When face sub-images of people from multiple views are passed to the PHC, each sub-image is submitted to the thread pool as a job. The thread pool creates a thread for each job to realize multi-view parallel identification. To further reduce the pipeline latency, we employ heterogeneous computation within each thread by CPU-GPU collaboration. Specifically, the AMN, FWN are executed by the CPU, while the heavy feature extraction models are executed by the GPU to fully utilize the overall computing power of the devices.

7 Dynamic Task Scheduling for Heterogeneous Devices

The scheduling of tasks is executed by the home edge. When the home edge schedules a task, it selects the DD from all edges and the cloud server to complete the DD-side task. To predict the EFT, the following information needs to be considered: 1) the length of the current unfinished task queue for each candidate device; 2) the computing power of each candidate device, such as CPU and GPU resources; 3) the transmission latency between each candidate device and drones. To this end, the home edge needs to be up to date on these three aspects of devices.

In order to avoid frequently obtaining information from all devices, the DTSH sets a scheduling period (e.g., 1 minute), which spans a series of tasks, denoted as $Queue_a$, which can be flexibly configured according to the system needs. For further details of the scheduling period, please refer to our supp. The scheduling process in one scheduling period includes the following three steps.

1) Synchronization. At the beginning of each scheduling period, the home edge obtains the latest device state information from edge devices, including queue lengths, available GPU/CPU resources, and current transmission latency.

2) Scheduling. The scheduling decision of one scheduling period includes selecting the suitable DD for each task within the period. When a device is already assigned for multiple tasks, its queue length becomes longer, so it is less likely to be selected again. The home edge selects DDs for tasks one by one in chronological order of tasks, because there is the temporal dependency between tasks.

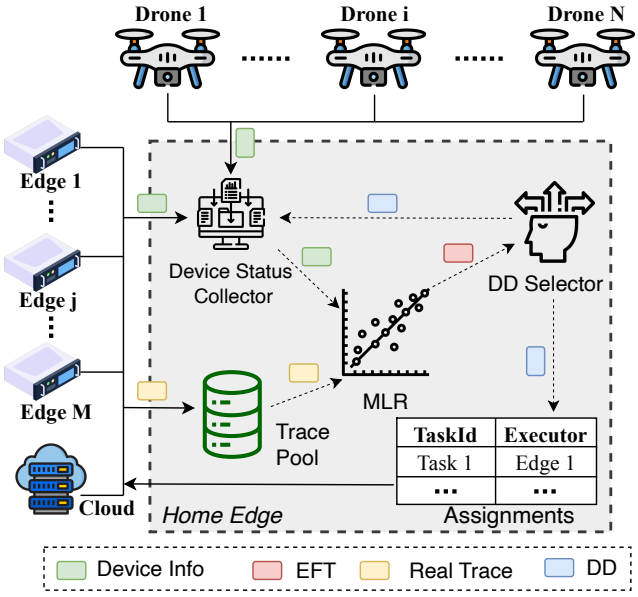


Figure 8: Dynamic Task Scheduling for Heterogeneous Devices.

3) Schedule decision dissemination. After all tasks are scheduled, the home edge sends the scheduling decision, i.e., the sequence of selected DDs, to all drones, edge devices and the cloud server.

In the following, we explain how to select the DD for a task within a scheduling period. Consider N drones $\mathbb{D} = \{D_1, \cdots, D_i, \cdots, D_N\}$, M edge devices $\mathbb{E} = \{E_1, \cdots, E_j, \cdots, E_M\}$ and a cloud server, denoted as E_0. Suppose the current unfinished tasks queue of E_j is $Queue_j$, and the CPU and GPU computing powers of device E_j are CPU_j and GPU_j (unit: TOPS), respectively. The transmission latency of E_j is calculated as the maximum transmission delay between it and all drones $Delay_j = \max_{i \in [1,N]} Delay_{j,i}$, where $Delay_{j,i}$ is the transmission delay between E_j and D_i.

For a task, the home edge selects the device with the smallest EFT as the task's DD. If the EFTs of all edge devices for one task are higher than the shooting interval (e.g., 1s), indicating all of them are busy, the cloud server is selected as the task's DD. The estimated finishing time EFT_j of device E_j can be predicted by an adaptive Multi-variable Linear Regression (MLR) model:

$$\begin{aligned} EFT_j =& MLR(Delay_j, ||Queue_j||, CPU_j, GPU_j) \\ =& \theta_0 + \theta_1 Delay_j + \theta_2 Queue_j + \theta_3 GPU_j + \theta_4 CPU_j. \end{aligned} \tag{13}$$

To train the MLR model, a trace pool is built to store the real historical traces. Each trace is a pair of data that records the MLR input vector $(Delay_j, Queue_j, CPU_j, GPU_j)$ and the real finishing time (ground truth). The MLR is updated through training with the real traces in the trace pool in each updating period (e.g., 2 minutes).

The training purpose of MLR is to find out a parameter set $\hat{\boldsymbol{\theta}} = (\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4)$, which is as close to the real parameter set $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4)$ as possible. Because the real $\boldsymbol{\theta}$ is unknown, we cannot directly compare the two. To

Algorithm 2. Dynamic Task Scheduling Algorithm (DTS)

Input: $Queue_a$, N, each D_i, M, each E_j:[CPU_j, GPU_j, $Queue_j$, $Delay_j$]

while DTS is running **do**
  assignments $\leftarrow \{\}$
  while $Queue_a$ is not empty **do**
    $taskId \leftarrow Queue_a.front()$
    for j in [1,M] **do**
      $EFT_j^{taskId} \leftarrow MLR(Delay_j, ||Queue_j||, CPU_j, GPU_j)$
    end for
    if $\min EFT_j^{taskId} <$ *shooting interval* **then**
      $DD \leftarrow \arg\min_j EFT_j^{taskId}$
    else
      $DD \leftarrow E_0$
    end if
    push (*taskId*, *DD*) into *assignments*
    push *taskId* into $Queue_{DD}$
  end while
  for i in [1,N] **do**
    sync *assignments* with D_i
  end for
end while

describe how close $\hat{\boldsymbol{\theta}}$ is to $\boldsymbol{\theta}$, we introduce a loss function:

$$Loss = \frac{1}{K}\sum_{j=1}^{K}(EFT_j - \tilde{EFT_j}), \tag{14}$$

where K is the number of traces, EFT_j is the real task finishing time of a completed task, and $\tilde{EFT_j}$ is the estimated EFT. The pseudo code of DTSH is shown in Algorithm 2.

8 Implementation

8.1 Hardware

Drones. We use the self-assembled F450 quadcopter as the drone in SkyNet. The quadcopter is equipped with a PIXHAWK [55] 2.4.8 flight control, an M8N GPS [56], and a 4K action camera. An AI edge computing platform, NVIDIA Jetson Xavier NX, is also deployed on the drone to provide 14 TOPS of computing power (low power mode).
Edge devices. We use the NVIDIA Jetson Xavier NX as the edge device in SkyNet, which can provide 21 TOPS of computing power due to the more abundant power supply.
Cloud server. The cloud server in SkyNet is implemented on a server running an Ubuntu 20.04 system with two Intel Xeno Silver 4210 @2.20GHz CPUs and four NVIDIA RTX 2080Ti GPUs with 12 GB of video memory.

8.2 Syetem Setup

We implement SkyNet on the above hardware. The source code of the SkyNet system is currently available for use [2].
Algorithms and DNNs. We use python and pytorch to implement the algorithms and DNNs in SkyNet. For the face detection pipeline on drones, we use YOLOX-Tiny [44] as the person detector and RetinaFace-10g [42] as the face detector. For the localization and identification pipeline on edges and cloud, we use ArcFace based on ResNet50 [1] and CLIP based on Transformer [37] as feature extractors. The MMFC, AMN, and FWN are all implemented by Pytorch 1.8.0. The MLR in DTSH is implemented by scikit-learn.
Wireless Connection. We use socket to achieve communication between devices, based on the TCP protocol. Data transfer between devices via WLAN (WiFi protocol 802.11 ac) at 18 Mbps upload/download rate.

2. https://doi.org/10.5281/zenodo.7467108

9 Dataset Evalution

9.1 Datasets

We evaluate SkyNet on the following public datasets. For further details of datasets, please refer to our supp.

- **CFP** [52] provides frontal and side images for each person. SkyNet uses 3 input channels for identification, corresponding to 3 images of a person.
- **DroneFace** [51] provides face images captured from different distances and depression angles, which could be used to simulate the FoV of drones.
- **TinyFace** [53] provides low-resolution images of faces gathered from public web data, which covers a wide range of imaging scenarios captured under uncontrolled viewing conditions.
- **SurvFace** [54] provides surveillance facial images captured in real-world uncooperative surveillance scenes, which can be used to simulate finding targets from a large crowd in complex scenarios.
- **AgeDB** [57] provides images of the same person with different ages, which could be used to evaluate the attribute hallucination of MMFC.

9.2 Baseline Methods

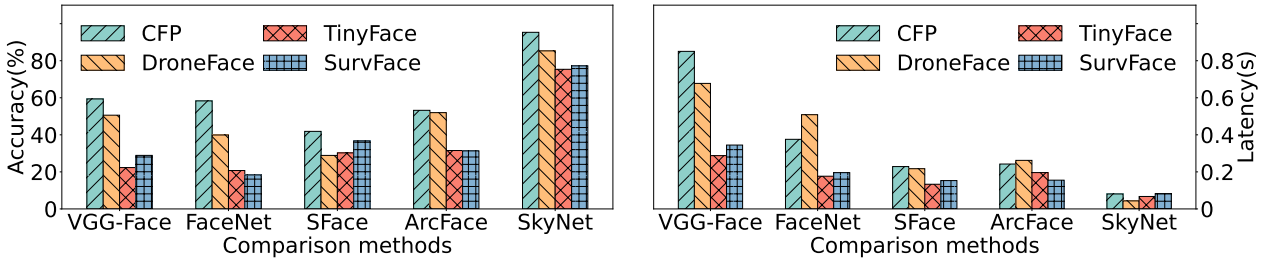
We select four mainstream face identification algorithms as baselines for comparison, which are implemented in the standard face identification library [58]. We use RetinaFace-10g for face detection in each baseline. The difference between baselines lies in the face extractors used, which include the following extractors:

- **VGG-Face (VGG)** [59]. A heavyweight face identification algorithm which use a "very deep" convolutional network [60] trained on a large scale dataset.
- **FaceNet (FN)** [61]. A classical face identification algorithm that learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity.
- **SFace (SF)** [62]. A lightweight algorithm trained by the sigmoid-constrained hypersphere loss.
- **ArcFace (AF)** [1]. A powerful face identification algorithm trained by the additive angular margin loss.

9.3 Evaluation Results on Public Dataset

9.3.1 Evaluation of face identification

Overall accuracy on various datasets. As shown in Figure 9a, the accuracy of SkyNet on four datasets (CFP, DroneFace, TinyFace, and SurvFace datasets) is 95.3%, 85.3%, 75.4%, and 77.2% respectively, which is 36.0%, 33.4%, 43.8%, and 40.5% higher than the optimal baseline method respectively.



(a) Accuracy comparison on four dataset. (b) Latency comparison on four dataset.

Figure 9: Overall Comparison on Four Public Datasets: CFP [52], DroneFace [51], TinyFace [53], and SurvFace [54].

On one hand, the facial images in the CFP dataset are high-resolution that have large proportions and more abundant features, so the VGG-Face and FaceNet with heavy networks achieve better accuracy than lightweight SFace and ArcFace. But the profile faces in the CFP dataset have more invalid information, which interferes with the extracted face features by VGG-Face and FaceNet, resulting in unsatisfactory accuracy. SkyNet uses single-view face feature masking and multi-view face fusion to ensemble feature information from multiple views, thus achieving the highest accuracy.

On the other hand, the facial images in DroneFace, TinyFace, and SurvFace are low-resolution, occupying only a small portion of the entire image and lacking many details. Therefore, the large-scale features extracted by heavyweight VGG-Face and FaceNet even have more noise, resulting in a decrease in accuracy. In addition, the images in TinyFace and SurvFace are captured in complex and uncontrolled scenarios, e.g., stations, campus, and stadium stands, which present significant challenges for small face detection. SkyNet's advantages shine even brighter in these complex scenarios, because it begins by searching for areas where faces are likely to appear with a person detector, then utilizes a face detector for precise face detection, successfully detecting small faces. The results also show the robustness of SkyNet in diverse commonly used scenarios.

Overall latency on various datasets. As shown in Figure 9b, the latency of SkyNet on four datasets is on average 3 times faster than the fastest baseline method. In general, the inference latency of all methods is longer on higher-resolution facial images, while lower-resolution facial images can speed up DNNs inference. Heavyweight models such as VGG-Face and FaceNet can bring high accuracy on high-resolution facial images, but their inference latency is intolerable. Lightweight models like ArcFace and SFace can infer fast but are stuck in the sequential execution of numerous images. Benefiting from the PHC, SkyNet's latency is significantly lower than other algorithms. In addition, even if the resolution of the facial images is low, full image input at high resolution slows down the traditional face identification pipeline. SkyNet uses pre-emptive lightweight person detection to make the face detector unnecessary to detect all areas in high-resolution full images. Therefore, even if the same face detector and extractor are used, SkyNet is still much faster than traditional pipelines.

9.3.2 Evaluation of the attribute hallucination

Accuracy improvement on CFP dataset. We evaluate the face identification accuracy improvement from using MMFC for attribute hallucination on the CFP dataset. For each person, we use one of his/her frontal images as the target image for identification. During identification, MMFC corrects profile face features into frontal face features to improve the accuracy of identification. Table 1 shows the improvement in identification accuracy resulting from using MMFC on the CFP dataset. The accuracy of SkyNet with MMFC is 95.3%, which is 4.2% higher than SkyNet without MMFC. By transforming the profile face into the frontal face, we can improve the accuracy of face identification.

TABLE 1: Identification accuracy improvement of the attribute hallucination on dataset evaluation.

Dataset	Opt. Baseline	SN w/o MMFC	SN with MMFC
CFP	59.4%	91.1%	95.3%
AgeDB	82.9%	82.9%	97.2%

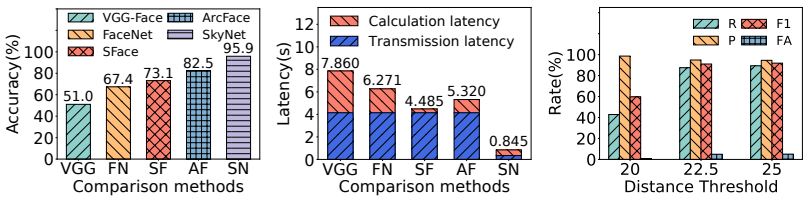
Accuracy improvement on AgeDB dataset. We evaluate the face identification accuracy improvement from using MMFC for attribute hallucination on the AgeDB dataset. For each person, we use an image of his/her youth (10–30) as the target image for identification. During identification, MMFC corrects old age face features into young age face features to improve accuracy (since the target image used for identification is the young image). Table 1 shows the improvement in identification accuracy resulting from using MMFC on the AgeDB dataset. SkyNet without MMFC is not optimized for age, which does not differ from the baseline on the AgeDB dataset. After hallucinating age by MMFC, the facial features at old age are transformed into facial features at a young age, so that the features are closer to the target images at a young age.

10 REAL-WORLD EVALUATION

10.1 Real-flight Experiment Setup

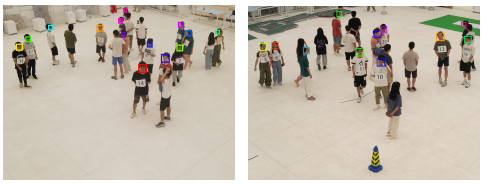
We deploy and evaluate SkyNet on four drones, three edge devices and a cloud server in real-world experiments. In two scenes with 20 people moving freely indoors at 81m^2 and outdoors at 554m^2, we use three drones to capture 4K images and one drone to capture 1080p images, and each experiment is performed for 5 minutes. The four drones are located at the four corners of each scene, 5m above the ground indoors and 10m above the ground outdoors. We run a total of nine experiments, including five indoor experiments and four outdoor experiments. [3]

3. We will open source the image dataset collected during the experiments, which are taken simultaneously by multiple drones on the same site, totaling 48.7K images and 736.8 GB.



(a) Accuracy. (b) Latency. (c) Other metrics.

Figure 10: The Overall Performance of SkyNet.



(a) Image captured from drone-view at 45°. (b) Image captured from drone-view at 225°.

Figure 11: On-Board Output Example.

10.2 SkyNet Overall Performance

We first evaluate the overall performance of SkyNet.

10.2.1 Baseline Methods and Operations

We select the four mainstream face identification algorithms as baselines for comparison, which are the same as the baseline methods in §9. At a once-per-second shooting instant, each drone takes a photo of the crowd. The baseline methods offload photos from all drones to an edge device for face identification, which sequentially searches for the target person from all the photos. We define this entire operation flow of transmitting and recognizing the target person to be found as a *task*.

10.2.2 Evaluation Metrics

- **Latency**. the time spent executing one task, including transmission latency and computational latency.
- **Accuracy**. Top-1 identification accuracy at a specific False Accept Rate (FAR), e.g., FAR=10^{-6}.
- **Precision (P)**. the percentage of tasks that correctly identify the target person among the tasks which identify a target person (note that the identified person may not be the actual target).
- **Recall (R)**. the percentage of tasks that correctly identify the target person among the tasks in which the target person exists.
- **False Alarm (FA)**. the percentage of tasks which incorrectly identify the target person among all tasks.
- **F1 score (F1)**. the harmonic average of precision and recall, i.e., $F1 = \frac{2\times P\times R}{P+R}$.

10.2.3 Evaluation Results

Accuracy. Figure 10a shows the overall accuracy of SkyNet and baselines. SkyNet has the highest accuracy of 95.87%, which is 44.83%, 28.52%, 22.76%, and 13.33% higher than VGG-Face, FaceNet, SFace and ArcFace baselines, respectively. VGG and FaceNet use traditional Euclidean Distance Loss for learning, resulting in poor accuracy in identifying small faces in drone views. SFace and ArcFace achieve acceptable accuracy of face identification on drones by their designed loss function. However, some of their success is due to the multi-angle faces provided by the multiple drones (i.e., even if a person's face cannot be found in a drone's photo, his/her identity can still be identified from other drones' photos that may capture his/her face), which are also wasted by these single-view face identification baselines. SkyNet uses single-view face feature mask to filter out invalid information of a single face and multi-view fusion identification to fuse the feature from multi-angle faces, which makes full use of the effective information from all angle faces and achieves the highest accuracy.

Latency. Figure 10b shows the comparison of the latency in completing once face identification (i.e., one *task*) between SkyNet and the baselines. SkyNet has the lowest latency of 0.845 seconds, which is 9.31 times, 7.42 times, 5.31 times, and 6.31 times faster than VGG-Face, FaceNet, SFace and ArcFace baselines, respectively. Heavy neural networks bring high calculation latency to VGG and FaceNet. The lightweight SFace has the lowest calculation latency, but transmitting a full 4K image makes it have a high transmission latency. By transmitting only face sub-images and fast PHC for processing multiple faces simultaneously, SkyNet reduces the transmission latency and calculation latency respectively, thus achieves the lowest overall latency.

Precision and recall. As shown in Figure 10c, when the threshold is 25, SkyNet can achieve an 89.33% recall while maintaining a 94.56% precision. In general, smaller distance thresholds increase the number of tasks that miss the target, which leads to a lower recall. On the contrary, higher distance thresholds bring more false positives, which leads to lower precision. With the threshold increasing, SkyNet's precision decreases slightly, but the recall increases greatly.

False alarm and F1 score. As shown in Figure 10c, when the threshold is 25, SkyNet can maintain a 5.06% false alarm rate and can achieve a 91.87% F1 score. Even with a large distance threshold, the false alarm is still within an acceptable range, meaning that SkyNet correctly detects the target person in most of the tasks.

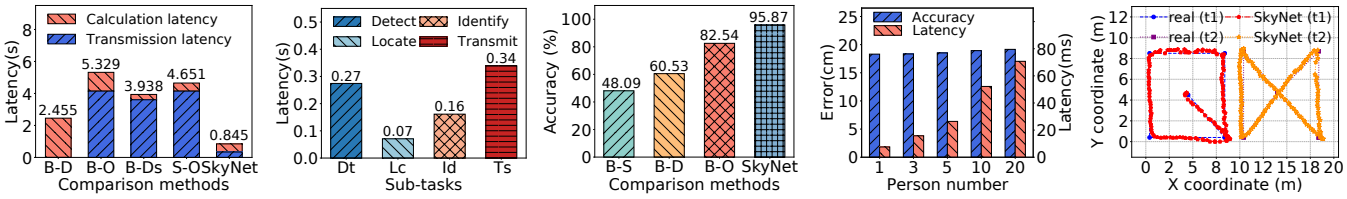
10.3 SkyNet Framework Performance

We analyze the effect of SkyNet's framework on accuracy and latency. At a once-per-second shooting instant, each drone takes a photo of the crowd. Using the same definition as in §10.2, we define the entire operation flow of locating and recognizing the target person to be found as a *task*.

10.3.1 Baseline Frameworks

We compare SkyNet with the following four baseline frameworks that for drone vision analysis:

- **Baseline of single drone (B-D)**. This baseline runs RetinaFace for face detection and ArcFace for face identification using a single drone with on-board computing power.
- **Baseline with full offloading (B-O)**. In this baseline, four drones transmit images to an edge device. The edge device then runs the face detection model, RetinaFace, to find all the people in the drone's view and calculate the feature distance between each person and the target person. If the shortest feature distance is lower than the threshold, the corresponding person is regarded as the target person.



(a) Latency comparison. (b) Latency of sub-tasks. (c) Accuracy comparison. (a) Localization error. (b) Track visualization.

Figure 12: The Framework Performance of SkyNet.

Figure 13: Evaluation of Localization.

- **Baseline with down sampling (B-S)**. This baseline is different from B-O because the drone transmits the down-sampled image (640×640) to the edge device.
- **SkyNet with full offloading (S-O)**. SkyNet runs without DTSH, i.e., the cloud always serves as DD.

10.3.2 Evaluation Metrics

- **Task latency**. the time spent executing a *task*, including transmission latency and computational latency.
- **Accuracy**. Top-1 identification accuracy at a specific False Accept Rate (FAR), e.g., FAR=10^{-6}.

10.3.3 Evaluation Results

Task latency. The latency of SkyNet is much smaller than all baselines. As shown in Figure 12a, SkyNet is 6.31 times faster than the B-O baseline, which transmits 4K images with a transmission latency of up to 4.154 seconds. SkyNet is also 2.91 times faster than the B-D baseline. Due to the limited computing resources of drones, B-D has a large computational latency, reaching 2.455s. We further investigate the cost of each sub-task (i.e., detection, localization, identification, and transfer) in the SkyNet task pipeline. As shown in Figure 12b, the detection sub-task has the largest latency, while the optimized localization and identification sub-tasks have less latency. It can be found that the transmission latency is greater than the computational latency. SkyNet with task scheduling can utilize the computing resources of multiple devices, thereby reducing the computational latency. Moreover, because only face sub-images are transmitted, the transmission latency is also reduced.

Accuracy. As shown in Figure 12c, SkyNet has the highest accuracy of 95.87%, which is 13.33%, 35.34% and 47.78% higher than the B-O, B-D, and B-S baselines, respectively. The accuracy of the B-S baseline is the lowest because it loses much information during down sampling. Figure 11 shows two images taken by two drones with the face bounding boxes generated by drone on-board computations. In SkyNet, full-resolution face sub-images are transferred to the DD. Furthermore, the multi-view fused face feature provides more information than any single-view face feature. Therefore, even if a person's face cannot be detected in a drone's FoV (e.g., a person with his/her back to the drone), his/her identity can still be identified because other drones may be able to capture his/her face.

10.4 Evaluation of SkyNet's Correction Performance

We evaluate the accuracy improvement from using MMFC on real-world evaluation. For each person, we use his/her face image taken at level angle as the target image for identification. MMFC converts the tilted face features into level face features to improve the accuracy of identification. Table 2 shows the improvement in accuracy resulting from using MMFC on the real-world evaluation. The accuracy of SkyNet with MMFC is 95.87%, which is 4.51% higher than SkyNet without MMFC. In addition, SkyNet with MMFC can achieve a 96.23% recall while maintaining a 95.66% precision, which means that SkyNet with MMFC can identify the faces missed by SkyNet without MMFC.

TABLE 2: Identification accuracy improvement of the attribute hallucination on real-world evaluation.

Metrics	SN w/o MMFC	SN with MMFC
Accuracy	91.36%	95.87%
Precision	94.87%	95.66%
Recall	87.52%	96.23%
F1-score	91.05%	95.79%

10.5 Evaluation of SkyNet's Localization Performance

We evaluate the localization performance of SkyNet in 5 experiments with different numbers of people. We calculate the localization error, i.e., the error between the MDPL output position and the true position, and the latency in completing the localization task.

Localization error. As shown in Figure 13a, the average error of each task collection is about 18.65 cm. Figure 13b visually shows the comparison of the two real tracks of the experimental participant and a series of consecutive positions output by SkyNet in one minute. The SkyNet-outputted positions are very close to the real tracks, which means that SkyNet has a high positioning accuracy.

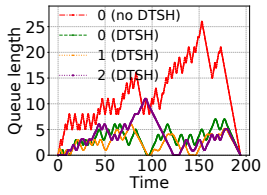
Localization latency. As shown in Figure 13a, the latency of MDPL increases with the number of people. With the increase in the number of people, the latency of MDPL positioning is also growing. The MDPL latency for twenty people is 0.071 seconds, which means that MDPL can locate a person in real time and align faces under multiple drones.

10.6 Evaluation of SkyNet's Scheduling Performance

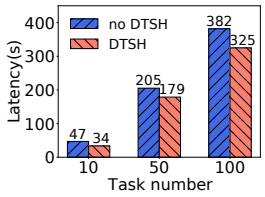
We analyze the impact of the DTSH module to evaluate the performance improvement brought by scheduling.

Latency. We compare the latency of SkyNet with and without the DTSH under different numbers of *tasks*. As shown in Figure 14b, the latency of SkyNet with the DTSH is about 15% lower than that of SkyNet without the DTSH.

Device queue length. Given 100 tasks, as shown in the figure 14a, if SkyNet runs without the DTSH, all tasks are assigned to the cloud server (device0 in the figure). If SkyNet runs with the DTSH module, the 100 tasks are assigned almost evenly to each device, so the length of the task waiting queue for each device over time is shorter when SkyNet runs with the DTSH module.

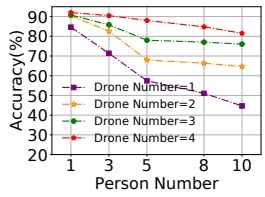


(a) Workload visualization of each device

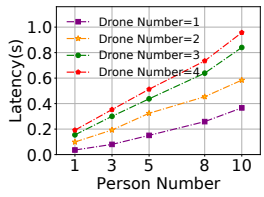


(b) Effect of DTSH on task latency

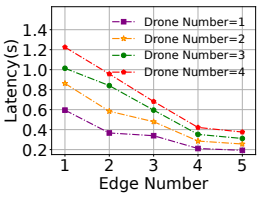
Figure 14: Evaluation of Scheduling.



(a) Effect of crowd size and drone number on accuracy.



(b) Effect of crowd size and drone number on latency.



(c) Effect of edge and drone number on latency.

Figure 15: Effect of Different Parameters on Performance.

10.7 Effect of Different Parameters

We analyze the effect of the crowd size, the drone number, and the edge number on SkyNet's accuracy and latency. To achieve this goal, we run SkyNet on different-sized crowds (from 1 to 10) using different numbers of drones (from 1 to 4) and edge devices (from 1 to 5).

Effect of crowd size. As shown in Figure 15a, larger crowds lead to lower accuracy due to mutual occlusion between people. Figure 15b also shows that as the number of people increases, the end-to-end latency increases due to the increase in sub-images to be processed and data to be transmitted due to computational offloading. End-to-end latency increases faster when the crowd size exceeds 5 people. Notice that the edge number is fixed to 3.

Effect of drone number. As shown in Figure 15a, using more drones can alleviate the problem of accuracy degradation caused by a large crowd as more details from multiple drones are captured. When the number of drones exceeds 3, the performance gain from adding drones decreases. Figure 15b shows that using more drones leads to higher end-to-end latency. Notice that the edge number is fixed to 3.

Effect of edge number. As shown in Figure 15c, utilizing additional edges can help alleviate the issue of latency surges caused by more drones, as the workload is distributed among more edge devices. When the number of edges exceeds 4, the performance gain from adding edges decreases. In addition, adding edges could not lead to higher accuracy. Notice that the crowd size is fixed at 10.

10.8 Deployment Recommendations

In both public datasets and the real-life flight experiment, SkyNet demonstrated its superior performance. Based on the above results, we provide recommended deployment methods for readers to better deploy SkyNet. To identify more than 10 people in an urban area of $500m^2$ and achieve 80% accuracy and decimeter-level positioning error, SkyNet needs 4 drones to complete identification in about 0.9s. These drones should be deployed in the four corners of the area, between 5m and 10m above the ground. After reaching the designated position, the flight speed of drones should be kept below 1m/s. The vertical view of the camera should be between 20° and 70° and the horizontal view should be between $\pm 20^\circ$ to the center axis, and WLAN is recommended for communication.

11 Related Work

11.1 Multi-drone Cooperation Systems

Multi-drone systems are increasingly used to collaboratively perform various visual tasks, such as crowd monitoring [63], [64], target identification [65], tracking [26], [66], etc. By utilizing redundancy and complementarity between the visual information of different drones, the multi-drone system can often outperform a single drone. Moraes et al. [64] design a multi-drone-based crowd monitoring system to periodically monitor a group of moving walking individuals, which uses auction protocols to assign different monitoring targets to drones to maximize the monitoring time of all targets in the scene. Patrizi et al. [66] propose a multi-drone framework to track and sense multiple targets. By capturing the tracking preferences of drones and targets, the system intelligently determines the target each drone should track. The above studies aim to reduce the redundancy of visual information between different drones, e.g., assigning each drone to track different targets. In contrast, SkyNet leverages the redundancy of visual information between multiple drones, i.e., multi-view facial images of the same target. By exploring the complementarity of these images, SkyNet can further improve the performance of person identification.

11.2 Face Identification on Drones

Recently, computer vision techniques have been applied to drones to detect faces [67]. Face identification on drones is an admittedly challenging task, due to noise, motion blur, and other factors unique to aerial photography. Fysh et al. [68] demonstrate recent work about face identification on drones, and focus on factors that are likely to affect identification performance from drone-recorded footage, such as image quality, and additional person-related information from the body and gait. This work suggests that person identification from drones is likely to be very challenging indeed, especially in real-world settings. Naser et al. [69] demonstrate how drones can be used for crowd surveillance by applying face identification technology. Erina et al. [70] propose a method for facial identification from drone-view images. Hwai-Jung Hsu et al. [71] propose that $Face^{++}$ and *ReKognition* are limited by the height, distance, and depression angle of drones. Triantafyllidou et al. [72] design a lightweight CNN for face detection, which is suitable for deployment in resources-limited drones. Amato et al. [73] focus on the impact of face multi-resolution on facial recognition and address the issue of facial recognition on drone video footage. There are some research projects aimed at improving the performance of face identification in normal scenes and completing high-precision face identification [74]. However, a single drone often suffers from limited face pixels due to its high-flying height and varying drone-person angles, which prevents general face identification algorithms from being deployed directly on drones.

Another limitation of face identification on drones is the limited computing resources of drones, which results that the powerful and heavy models can not be deployed on drones. An effective solution is to offload the computing tasks to the cloud server [75], latency requirements of which is a concern. Dick et al. [76] propose the use of a server on the network edge to optimize both processing capability as well as latency for applications requiring real-time communication between a drone and a cloud server. FCSD [77] architecture is proposed to minimize the energy consumption under the constraints of latency and reliability. Wang et al. [78] propose a fog-networking based system architecture to associate each drone to an optimally matched server. However, the above work generally make simulation tests in a predetermined environment, which make them difficult to be applied to dynamic real drones-based systems.

11.3 Multi-camera Computer Vision

Compared with single cameras, multiple cameras/drones can expand the field of view and get more information [79]. Pierre Baque et al. [80] combines CNN and CRF to model potential occlusions and combine multi-view information. DyGLIP [81] uses both link prediction and dynamic graph in the MC-MOT framework and produces highly accurate link prediction results. MDOT [26] proposes the first multi-drone single target tracking benchmark dataset and an integrated framework ASNet to improve the tracking accuracy.

One main challenge in face identification using multiple cameras/drones is aligning multi-view information [82]. Many studies of multi-view alignment use carefully selected feature markers to extract object features and match them into another view, such as Scale-Invariant Feature Transform (SIFT) [83], Speeded-Up Robust Features (SURF) [84], and Histogram of Oriented Gradients (HOG) [85]. These studies use machine learning models such as SVM [86], Adaboost [87] for feature matching, and use the sliding window technique [88] to search images of other views, with high matching complexity and computational latency. These limit their effectiveness and applicability.

11.4 Localization and tracking

Most 3D localization solutions acquire additional sensory information through specialized devices such as RGB-D cameras and LiDARs. Knoppe et al. [89] propose a drone system with a stereo camera that collects spectral image patches with stereoscopic overlaps to get ground surface scanning data. Guo et al. [90] propose a framework to implement 3D object co-localization from mobile LiDAR point clouds, which can extract the objects of the same category from different point-cloud scenes. Li et al. [32] propose a camera localization workflow based on a 3D prior map optimized by RGB-D SLAM method. Wang et al. [91] use the collaboration of drones equipped with MIMO radar to locate marine targets based on triangulation. However, the reliance on specialized equipment makes these solutions expensive and difficult to deploy widely. Moreover, these advanced sensors could increase the power consumption of drones and RGB-D cameras cannot obtain accurate and consistent 3D data in the immense outdoor.

Object tracking is critical for scenarios that require continuous targeting of objects, such as capture and child searches. Zhang et al. [92] back-project target objects from drone's 2D image plane to 3D world coordinates by camera geometry, thus achieve object tracking. Silva et al. propose a face identification and tracking system [93], in which the same person in different video frames is re-identified based on the face embedding vectors obtained through CNN. ML-LocNet [94] performs multi-view co-training to enhance localization and tracking performance. PML-LocNet [95] exploits both view diversity and sample diver sity which performs more stable compared with ML-LocNet. However, the target's position is relative to image coordinates rather than world coordinates, making it difficult to accurately track targets in the real world.

11.5 Vision-Language Models for Face Identification

In recent years, there has been a growing interest in developing vision-language models [36], which aim to bridge the gap between visual understanding and natural language processing. Radford et al. [37] propose the Contrastive Language-Image Pre-training (CLIP) model, which uses a text encoder and an image encoder to represent the text and image to an image-language embedding space. CLIP is trained by contrastive learning on a vast dataset of 400 million image-text pairs, resulting in a cohesive image-language space and has demonstrated robust zero-shot classification capabilities.

Motivated by the impressive performance of these vision-language models, some follow-ups have been proposed for face identification. Srivatsan et al. [96] use the image encoder in CLIP for Face anti-spoofing (FAS), achieving better zero-shot transfer performance on unseen spoof types. Shahid et al. [97] evaluate the performance of the CLIP model as a zero-shot face recognizer and proposed an unsupervised dual modality prompt learning framework for facial expression recognition. Shen et al. [40] propose an attribute hallucination framework named CLIP-Cluster, which could cluster the images of the same identity but with different face attributes, e.g., age, pose, and expression. The above work demonstrates promising results in face identification tasks, showing the potential of utilizing the visual-language models for enhanced recognition performance.

12 Conclusion

In this paper, we propose SkyNet, a multi-drone cooperation system for accurate and real-time identification and localization. SkyNet can accurately locate a person in 3D real world using only conventional 2D cameras and can align face sub-images of one person from different drone views. To improve identification accuracy, we design a novel fusion identification pipeline, which exploits images from different views by fusing them according to weights reflecting legibility. SkyNet can also correct the tilted faces in drone-view image to enhance legibility. Moreover, SkyNet can achieve real-time localization and identification through its ability of dynamic task scheduling. We implement and evaluate SkyNet in real life, and the result shows that SkyNet achieves 91.36% identification accuracy, less than 0.18m localization error and less than 0.84s latency.

13 Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under grant No. 2022YFB3105000, the Major Key Project of PCL under grant NO. PCL2023A06, and the Shenzhen Key Lab of Software Defined Networking under grant No. ZDSYS20140509172959 98. Preliminary results in this paper are presented at the IEEE International Conference on Computer Communications Conference, 2023 [65].

References

[1] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 4685–4694.

[2] J. Lezama, Q. Qiu, and G. Sapiro, "Not afraid of the dark: Nir-vis face recognition via cross-spectral hallucination and low-rank embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Jul. 2017, pp. 6628–6637.

[3] C.-H. Chu and Y.-K. Feng, "Study of eye blinking to improve face recognition for screen unlock on mobile devices," *Journal of Electrical Engineering and Technology*, vol. 13, no. 2, pp. 953–960, Mar. 2018.

[4] S. Girmay, F. Samsom, and A. M. Khattak, "Ai based login system using facial recognition," in *2021 5th Cyber Security in Networking Conference (CSNet)*. IEEE, Oct. 2021, pp. 107–109.

[5] S. Li, X. Ning, L. Yu, L. Zhang, X. Dong, Y. Shi, and W. He, "Multi-angle head pose classification when wearing the mask for face recognition under the covid-19 coronavirus epidemic," in *2020 international conference on high performance big data and intelligent systems (HPBD&IS)*. IEEE, May. 2020, pp. 1–5.

[6] H.-M. Hsu, Y. Wang, and J.-N. Hwang, "Traffic-aware multi-camera tracking of vehicles based on reid and camera link model," in *Proceedings of the 28th ACM International Conference on Multimedia*, Oct. 2020, pp. 964–972.

[7] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp. 8797–8806.

[8] H. P. D. Nguyen and D. D. Nguyen, "Drone application in smart cities: The general overview of security vulnerabilities and countermeasures for data communication," *Development and Future of Internet of Drones (IoD): Insights, Trends and Road Ahead*, pp. 185–210, 2021.

[9] L. B. Neto, F. Grijalva, V. R. M. L. Maike, L. C. Martini, D. Florencio, M. C. C. Baranauskas, A. Rocha, and S. Goldenstein, "A kinect-based wearable face recognition system to aid visually impaired users," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 1, pp. 52–64, Feb. 2017.

[10] S. Panchanathan, S. Chakraborty, and T. McDaniel, "Social interaction assistant: a person-centered approach to enrich social interactions for individuals with visual impairments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 5, pp. 942–951, Aug. 2016.

[11] "Intel depth camera d455," https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d455.html, 2022.

[12] Y. W. Kuan, N. O. Ee, and L. S. Wei, "Comparative study of intel r200, kinect v2, and primesense rgb-d sensors performance outdoors," *IEEE Sensors Journal*, vol. 19, no. 19, pp. 8741–8750, Oct. 2019.

[13] Y. Moolla, A. De Kock, G. Mabuza-Hocquet, C. S. Ntshangase, N. Nelufule, and P. Khanyile, "Biometric recognition of infants using fingerprint, iris, and ear biometrics," *IEEE Access*, vol. 9, pp. 38 269–38 286, 2021.

[14] K. Shaheed, A. Mao, I. Qureshi, M. Kumar, Q. Abbas, I. Ullah, and X. Zhang, "A systematic review on physiological-based biometric recognition systems: Current and future trends," *Archives of Computational Methods in Engineering*, vol. 28, no. 7, pp. 4917–4960, 2021.

[15] N. Nanthini, K. Jeyalakshm, S. Kamalakannan, A. V. S, and A. T, "Face clustering approach for crime rate reduction in smart city development," in *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2021, pp. 890–895.

[16] M. Sahani, C. Nanda, A. K. Sahu, and B. Pattnaik, "Web-based online embedded door access control and home security system based on face recognition," in *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. IEEE, 2015, pp. 1–6.

[17] Y. Zhao, S. Wu, L. Reynolds, and S. Azenkot, "A face recognition application for people with visual impairments: Understanding use beyond the lab," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Apr. 2018, pp. 1–14.

[18] S. Ramnath, A. Javali, B. Narang, P. Mishra, and S. K. Routray, "Iot based localization and tracking," in *2017 International Conference on IoT and Application (ICIOT)*, 2017, pp. 1–4.

[19] S.-H. Chan, P.-T. Wu, and L.-C. Fu, "Robust 2d indoor localization through laser slam and visual slam fusion," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 1263–1268.

[20] S. Krul, C. Pantos, M. Frangulea, and J. Valente, "Visual slam for indoor livestock and farming using a small drone with a monocular camera: A feasibility study," *Drones*, vol. 5, no. 2, p. 41, 2021.

[21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[22] T. Chen and S. Lu, "Object-level motion detection from moving cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2333–2343, 2016.

[23] B. Kalantar, S. B. Mansor, A. A. Halin, H. Z. M. Shafri, and M. Zand, "Multiple moving object detection from uav videos using trajectories of matched regional adjacency graphs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5198–5213, 2017.

[24] K. Yun, J. Lim, and J. Y. Choi, "Scene conditional background update for moving object detection in a moving camera," *Pattern Recognition Letters*, vol. 88, pp. 57–63, 2017.

[25] A. Zheng, L. Zhang, W. Zhang, C. Li, J. Tang, and B. Luo, "Local-to-global background modeling for moving object detection from non-static cameras," *Multimedia Tools and Applications*, vol. 76, no. 8, pp. 11 003–11 019, 2017.

[26] P. Zhu, J. Zheng, D. Du, L. Wen, Y. Sun, and Q. Hu, "Multi-drone-based single object tracking with agent sharing network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 4058–4070, 2021.

[27] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[28] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[29] H. Taira, I. Rocco, J. Sedlar, M. Okutomi, J. Sivic, T. Pajdla, T. Sattler, and A. Torii, "Is this the right place? geometric-semantic pose verification for indoor visual localization," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, nov 2019, pp. 4372–4382.

[30] X. Xin, J. Jiang, and Y. Zou, "A review of visual-based localization," in *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*, 2019, pp. 94–105.

[31] L. Du, X. Ye, X. Tan, J. Feng, Z. Xu, E. Ding, and S. Wen, "Associate-3ddet: Perceptual-to-conceptual association for 3d point cloud object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 329–13 338.

[32] J. Li, C. Wang, X. Kang, and Q. Zhao, "Camera localization for augmented reality and indoor positioning: a vision-based 3d feature database approach," *International journal of digital earth*, 2019.

[33] B. S. Zapata-Impata, P. Gil, J. Pomares, and F. Torres, "Fast geometry-based computation of grasping points on three-dimensional point clouds," *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, p. 1729881419831846, 2019.

[34] J. Kempfle and K. Van Laerhoven, "Respiration rate estimation with depth cameras: An evaluation of parameters," in *Proceedings*

of the 5th international Workshop on Sensor-based Activity Recognition and Interaction, 2018, pp. 1–10.

[35] "Intel lidar camera l515," https://store.intelrealsense.com/buy-intel-realsense-lidar-camera-l515.html, 2022.

[36] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie, "Prompting visual-language models for efficient video understanding," in *European Conference on Computer Vision*. Springer, 2022, pp. 105–124.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[38] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2085–2094.

[39] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3835–3844.

[40] S. Shen, W. Li, X. Wang, D. Zhang, Z. Jin, J. Zhou, and J. Lu, "Clip-cluster: Clip-guided attribute hallucination for face clustering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 786–20 795.

[41] "Nvidia jetson xavier nx," https://developer.nvidia.com/embedded/jetson-xavier-nx, 2020.

[42] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 5202–5211.

[43] "Dji mavic," https://www.dji.com, 2022.

[44] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, Aug. 2021.

[45] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "Ssh: Single stage headless face detector," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 4885–4894.

[46] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *Proceedings of the European conference on computer vision (ECCV)*, Sept. 2018, pp. 797–813.

[47] P. Hu and D. Ramanan, "Finding tiny faces," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1522–1530.

[48] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[49] S. Kim, J. Kim, J. Park, and D. Lee, "Vision-based pose estimation of fixed-wing aircraft using you only look once and perspective-n-points," *Journal of Aerospace Information Systems*, vol. 18, no. 9, pp. 659–664, May. 2021.

[50] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the seventh ieee international conference on computer vision*, vol. 1. Ieee, Sept. 1999, pp. 666–673.

[51] H.-J. Hsu and K.-T. Chen, "Droneface: an open dataset for drone research," in *Proceedings of the 8th ACM on multimedia systems conference*, Jun. 2017, pp. 187–192.

[52] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jul. 2016, pp. 1–9.

[53] Z. Cheng, X. Zhu, and S. Gong, "Low-resolution face recognition," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 605–621.

[54] ——, "Surveillance face recognition challenge," *arXiv preprint arXiv:1804.09691*, 2018.

[55] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, May. 2011, pp. 2992–2997.

[56] U-blox, "China reports expansion in 5g network coverage," 2021. [Online]. Available: https://www.u-blox.com/en

[57] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "Agedb: the first manually collected, in-the-wild age database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, vol. 2, no. 3, 2017, p. 5.

[58] S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2020, pp. 23–27. [Online]. Available: https://doi.org/10.1109/ASYU50717.2020.9259802

[59] O. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015.

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

[61] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[62] Y. Zhong, W. Deng, J. Hu, D. Zhao, X. Li, and D. Wen, "Sface: Sigmoid-constrained hypersphere loss for robust face recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 2587–2598, 2021.

[63] Y. Tan, Q. Li, J. Peng, Z. Yuan, and Y. Jiang, "Air-cad: Edge-assisted multi-drone network for real-time crowd anomaly detection," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 2817–2825.

[64] R. S. De Moraes and E. P. De Freitas, "Multi-uav based crowd monitoring system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 2, pp. 1332–1345, 2019.

[65] J. Peng, L. Qing, Y. Tan, D. Zhao, Z. Yuan, J. Chen, H. Wang, and Y. Jiang, "Skynet: Multi-drone cooperation for real-time identification and localization," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2023, pp. 1–10.

[66] N. Patrizi, G. Fragkos, K. Ortiz, M. Oishi, and E. E. Tsiropoulou, "A uav-enabled dynamic multi-target tracking and sensing framework," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[67] M. Shahbazi, J. Theau, and P. Menard, "Recent applications of unmanned aerial imagery in natural resource management," *Mapping Sciences & Remote Sensing*, vol. 51, no. 4, pp. 339–365, Jun. 2014.

[68] M. C. Fysh and M. Bindemann, "Person identification from drones by humans: insights from cognitive psychology," *Drones*, vol. 2, no. 4, p. 32, 2018.

[69] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[70] E. Ferro, C. Gennaro, A. Nordio, F. Paonessa, C. Vairo, G. Virone, A. Argentieri, A. Berton, and A. Bragagnini, "5g-enabled security scenarios for unmanned aircraft: Experimentation in urban environment," May. 2020.

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

[72] D. Triantafyllidou, P. Nousi, and A. Tefas, "Fast deep convolutional face detection in the wild exploiting hard sample mining," *Big data research*, vol. 11, pp. 65–76, Mar. 2018.

[73] G. Amato, F. Falchi, C. Gennaro, F. V. Massoli, and C. Vairo, "Multi-resolution face recognition with drones," in *2020 3rd International Conference on Sensors, Signal and Image Processing*, Oct. 2020, pp. 13–18.

[74] "Iarpa-janus," https://www.iarpa.gov/research-programs/janus.

[75] T. Kasidakis, G. Polychronis, M. Koutsoubelias, and S. Lalis, "Reducing the mission time of drone applications through location-aware edge computing," in *2021 IEEE 5th International Conference on Fog and Edge Computing (ICFEC)*, Feb. 2021, pp. 45–52.

[76] J. Dick, C. Phillips, S. H. Mortazavi, and E. de Lara, "High speed object tracking using edge computing: Poster abstract," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, ser. SEC '17. New York, NY, USA: Association for Computing Machinery, 2017.

[77] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, "Distributed fog computing for latency and reliability guaranteed swarm of drones," *IEEE Access*, vol. 8, pp. 7117–7130, 2020.

[78] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 308–318.

[79] F. Dittrich, L. E. de Oliveira, A. S. Britto Jr, and A. L. Koerich, "People counting in crowded and outdoor scenes using a hybrid multi-camera approach," *arXiv preprint arXiv:1704.00326*, 2017.

[80] P. Baqué, F. Fleuret, and P. Fua, "Deep occlusion reasoning for multi-camera multi-target detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 271–279.

[81] K. G. Quach, P. Nguyen, H. Le, T.-D. Truong, C. N. Duong, M.-T. Tran, and K. Luu, "Dyglip: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 784–13 793.

[82] D. Wierzbicki, "Multi-camera imaging system for uav photogrammetry," *Sensors*, vol. 18, no. 8, Jul. 2018.

[83] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[84] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

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

[86] V. Vapnik and A. Y. Lerner, "Recognition of patterns with help of generalized portraits," *Avtomat. i Telemekh*, vol. 24, no. 6, pp. 774–780, 1963.

[87] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[88] K. R. Sapkota, S. Roelofsen, A. Rozantsev, V. Lepetit, D. Gillet, P. Fua, and A. Martinoli, "Vision-based unmanned aerial vehicle detection and tracking for sense and avoid systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Ieee, Oct. 2016, pp. 1556–1561.

[89] E. Honkavaara, H. Saari, J. Kaivosoja, I. Pölönen, T. Hakala, P. Litkey, J. Mäkynen, and L. Pesonen, "Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight uav spectral camera for precision agriculture," *Remote Sensing*, vol. 5, no. 10, pp. 5006–5039, Oct. 2013.

[90] W. Guo, J. Chen, W. Wang, H. Luo, and S. Wang, "Three-dimensional object co-localization from mobile lidar point clouds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 1996–2007, 2021.

[91] X. Wang, L. T. Yang, D. Meng, M. Dong, K. Ota, and H. Wang, "Multi-uav cooperative localization for marine targets based on weighted subspace fitting in sagin environment," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5708–5718, Apr. 2022.

[92] H. Zhang, G. Wang, Z. Lei, and J.-N. Hwang, "Eye in the sky: Drone-based object tracking and 3d localization," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 899–907.

[93] S. H. Silva, P. Rad, N. Beebe, K.-K. R. Choo, and M. Umapathy, "Cooperative unmanned aerial vehicles with privacy preserving deep vision for real-time object identification and tracking," *Journal of parallel and distributed computing*, vol. 131, pp. 147–160, Apr. 2019.

[94] X. Zhang, Y. Yang, and J. Feng, "Ml-locnet: Improving object localization with multi-view learning network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[95] X. Zhang, Y. Yang, H. Xiong, and J. Feng, "Pml-locnet: Improving object localization with prior-induced multi-view learning network," *IEEE Transactions on Image Processing*, vol. 29, pp. 2568–2582, 2019.

[96] K. Srivatsan, M. Naseer, and K. Nandakumar, "Flip: Cross-domain face anti-spoofing with language guidance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 685–19 696.

[97] M. Shahid, "Unsupervised dual modality prompt learning for facial expression recognition," in *2023 8th International Conference on Computer and Communication Systems (ICCCS)*, 2023, pp. 1056–1061.

[98] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[99] G. Joshi, R. Walambe, and K. Kotecha, "A review on explainability in multimodal deep neural nets," *IEEE Access*, vol. 9, pp. 59 800–59 821, 2021.

Junkun Peng received his B.S. degree in Information Management and Information Systems from Shanghai University, Shanghai, China, in 2015. He is currently pursuing a Ph.D. in Computer Science at Tsinghua University. His research focuses on in-network caching/computing, real-time video transmission and analysis, and robot learning.

Qing Li (Senior Member, IEEE) received the B.S. degree in computer science and technology from the Dalian University of Technology, Dalian, China, in 2008, and the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2013. He is currently an Associate Researcher with the Peng Cheng Laboratory, Shenzhen, China. His research focuses on network function virtualization, in-network caching/computing, intelligent self-running networks, and edge computing.

Yuanzheng Tan received his B.S. degree in intelligent science and technology from Sun Yat-Sen University, ShenZhen, China, in 2023. He is currently pursuing a Ph.D. in computer science and technology with Tsinghua University (joint program with Peng Cheng Laboratory). His research interests include edge computing, multimedia transmission and analysis, and UAV swarm intelligence.

Dan Zhao is currently an Assistant Researcher with Peng Cheng Laboratory, Shenzhen, China. She was a Postdoctoral Researcher with School of Electronic Engineering, Dublin City University. Dr Dan Zhao obtained her bachelor's degree from Beijing University of Posts and Telecommunications in 2011, majored in Telecommunications. In 2015, she graduated from Queen Mary University of London with a Ph.D. degree in Electronic Engineering.

Jinhua Chen is currently a senior majoring in Intelligent Science and Technology at Sun Yat-sen University in Shenzhen, China. His research interests include cluster scheduling, large-scale reinforcement learning, and drone video analysis.

Yong Jiang (Member, IEEE) received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1998 and 2002, respectively. He is currently a Full Professor with the Division of Information Science and Technology, Tsinghua Shenzhen International Graduate School, Shenzhen, China, and the Peng Cheng Laboratory, Department of Mathematics and Theories, Shenzhen. He mainly focuses on the future internet architecture, the Internet of Things, edge computing, and AI for networks.