Peer-to-Peer Networking and Applications (2024) 17:1667–1681
https://doi.org/10.1007/s12083-024-01659-1

A UAV aided lightweight target information collection and detection approach

Meng Huang[1] · Hanming Li[1] · Yina Zhou[2] · Ting Ma[2] · Jinshan Su[1] · Haibo Zhou[2]

Received: 26 September 2023 / Accepted: 1 February 2024 / Published online: 13 March 2024

Abstract
With the resumption of the World Cup and various concerts, the number of heavily crowded scenarios grows intensely. In these cases, it poses new challenges to massive information collection and lightweight target detection. Fortunately, the booming development of unmanned aerial vehicle (UAV) technology provides a highly flexible and cost-effective solution in many scenarios. This paper proposes a UAV aided lightweight target information collection and detection approach, where the target information is carried to a terrestrial distributed platform by a UAV and then a fast target detection is implemented. Firstly, we implement a 3D trajectory optimization for the UAV by minimizing the information collection time. Secondly, we design a lightweight target detection algorithm based on UAV loadable ARM (Advanced RISC Machines) architecture edge computing device. Finally, a terrestrial distributed processing platform is established. To ensure the stability and reliability of the target detection system, each module is tested separately. Numerical simulations show that, with the target detection module deployed on the Jetson Xavier NX edge computing platform for testing, the proposed target detection approach can achieve a detection accuracy of 89.5% and 71FPS detection speed using GPU acceleration compared with state-of-the-art methods.

Keywords UAV · Trajectory optimization · Edge computing · Distributed computation

1 Introduction

An Unmanned Aerial Vehicle (UAV, also known as a drone) is a vehicle that can fly without a pilot and is remotely controlled by a human or computer [1]. Following the advancement of industrial technology, significant progress has been made in the field of UAVs. In many fields such as aerial photography, agricultural mapping and traffic regulation, drones are playing a huge advantage in terms of cost and flexibility. In a world where video surveillance has become of central importance in a variety of scenarios, drones are of strategic importance. Capturing footage from the air for film and news reporting, UAVs produce high quality and low cost video [2]. In the field of agricultural mapping and remote sensing, UAVs fill the gap of traditional satellite remote sensing with high accuracy, low cost, fast production cycles and great flexibility, and with more detail in the images captured by UAVs, making it possible to detect small targets. UAVs can monitor and control traffic flow in real-time in today's urban intelligent traffic network management to lessen traffic congestion [3].

As the global epidemic comes to an end, many concerts and sporting events are taking place as scheduled. At large events, safety and security are critical. With the advancement of technology, UAV and face recognition technologies have revolutionised event security. For example, the god of song, Jacky Cheung, who is known as the "fugitive's nemesis",

✉ Jinshan Su
sqsjs1968@aliyun.com

✉ Haibo Zhou
haibozhou@nju.edu.cn

Meng Huang
hm1771860025@163.com

Hanming Li
ylnulhm@163.com

Yina Zhou
zyn@nju.edu.cn

Ting Ma
majiawan27@163.com

[1] Key Laboratory of Vibration Signal Capture and Intelligent Processing, Yili Normal University, Yining 835000, China

[2] School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China

successfully assisted the police in arresting five fugitive suspects in four concerts, with the strong support of face recognition technology. At the concerts, face detection and comparison technology can quickly and accurately identify the people present to ensure the safety of the event. The introduction of UAV has greatly enhanced the efficiency and scope of face recognition, and UAV has the advantage of aerial surveillance, which can actively, multi-angle and multi-directional tracking of targets, greatly expanding the tracking range and detection angle [4]. Combining UAV and face recognition system can more easily complete the tasks of personnel search and remote surveillance. Compared to traditional security cameras capturing face data, UAV with face recognition system is more flexible, shorter deployment time, and lower cost. This innovation not only reduces security risks by quickly identifying suspicious persons, but also reduces labour costs and human error.

In order to ensure a certain degree of autonomy for UAVs, the first problem to be solved is path planning. Finding the best flight route for many UAVs from the starting point to the destination in a safe, effective, and economical way without running into any obstacles in the environment or entering a no-fly zone is the major goal of UAV path planning [5]. Distributed methods, population-based evolutionary algorithms, and graph-based algorithms are the primary path planning techniques created to date [6]. In [7], a route planning algorithm based on an ant colony algorithm and an artificial potential field was suggested. The method creates an artificial field that represents the environment for collision-free path planning while taking into account both dynamic threats and static impediments.For the purpose of designing UAV flight paths in static situations, [8] researched an advanced swarm optimization technique based on the bat algorithm (BA). The multi-objective social learning pigeon-inspired optimization method (MSLPIO) was proposed by [9] and used for UAV formation obstacle avoidance. However, in fixed circumstances, these techniques did not effectively address the issue of acquisition route planning for various crowd distributions.

At the same time, target detection in aerial video images from UAVs is challenging: the size of the target observed from an overhead platform is small, the target occupies only a small number of pixels in the image and lacks many of the features typically present when dealing with ground-based surveillance scenes; the degrees of freedom of the UAV cause changes in camera position and angle, resulting in instability in the UAV image. The movement of the drone changes the visual shape of the target, i.e., the size and position of the target within the image changes. This visual change alters the appearance of the target and hinders the detection task, a phenomenon referred to as multi-viewpoint [10]. Multi-view points are affected by roll, pitch and yaw movements, i.e. the rotation of the UAV on the x-, y- and z-axes. Traditional methods are mainly based on algorithms such as feature extraction and classifiers for target recognition, but they are limited by computational complexity and algorithm accuracy, and often suffer from slow target recognition, which cannot meet the demand for real-time detection in high-speed motion and dense scenes. These reasons have given rise to many difficulties in UAV lightweight target detection techniques.

Due to the limited load capacity, computational power and storage space of UAVs, lightweighting is one of the challenges of face detection in UAVs. The current top-ranked methods all use large pre-trained backbone networks, complex feature enhancement modules and heavy test time augmentations (TTAs) to achieve better rankings [11]. For example, Mogface [12], one of the best detectors available today, achieves state-of-the-art accuracy standards with 808 GFLOPs and 711M parameters. But its surprising accuracy rates all stem from the consumption of large amounts of storage and computational resources. In this scenario, there is a need for a highly accurate and fast running face detector. A Light and Fast Face Detector (LFFD) for edge devices is introduced in [13]. It combines RF "anchor points" and proper RF pacing to detect a wide range of continuous face scales. Zhang et al. [14], Chi et al. [15], and Liu et al. [16] have investigated various anchor sampling/matching strategies to balance the ratio of positive and negative samples, match the outer surface with high-quality anchors, and accelerate model convergence. BlazeFace is a lightweight, high-performing face detector with increased inference speed that was suggested by [17] for use in mobile GPU inference. [18] created a face detector called YOLO5Face that was based on the YOLOv5 target detector. Kim et al. [19] based on the YOLOv5 model, adds a prediction header and a novel multi-scale fusion layer to the backbone to improve detection performance. Wu et al. [20] build the network using a small but efficient feature extraction backbone and a simplified pyramidal feature fusion neck that greatly reduces the number of parameters. Different model sizes, from big models for best performance to extremely small models for real-time detection on embedded or mobile devices, were taken into account when designing detectors. However, these approaches were either designed for very small input sizes that could not meet the subsequent feature extraction requirements, or the network was not designed to operate on edge computing devices based on the arm architecture, such that the detection rate could not meet the requirements when actually deployed.

Based on the above problems and challenges, this paper proposes a UAV-assisted lightweight target information acquisition and detection method, in which a UAV transmits target information to a ground-based distributed platform to

achieve fast target detection. The proposed work consists of following major contributions which includes.

(1) Implemented a 3D trajectory optimization for the UAV to minimize the information collection time.
(2) Considering the load limitation and algorithmic capability of UAVs, we use MobileNetV3-Small network as a backbone, build an anchorless face detection network framework using the idea of FPN and SSH, and design a lightweight target detection algorithm that can be used for UAVs loaded with ARM (Advanced RISC Machines) architecture edge computing devices use.
(3) It is noted that a large number of face matching jobs place a huge demand on the computational power and memory of the UAV. Therefore, a Hadoop-based ground-based distributed processing platform is established to process the target data collected by the UAV.

2 Model and methods

2.1 System model

New opportunities for employing UAVs as an inspection platform are emerging thanks to the development and use of deep learning. Through a rational adaptive path planning design, the UAV adaptively and quickly optimization trajectory to cover the whole field of targets for scanning and information collection. By using a lightweight deep convolutional network as the backbone, combined with a targeted network structure design, achieves a lightweight, accurate and real-time target detection model that can be carried by the UAV's own computing power. By encoding and packaging the detected target images, sending them to the ground platform and using Hadoop to achieve fast feature comparison. The project believes that the combination of UAV aerial information collection and ground server recognition processing to form an "Airground Collaboration" model may be a better way to quickly find a specific target among a large number of targets. Not only solving the problem of the lack of flexibility of traditional security cameras and the lack of UAV computing power, but also making the target detection algorithm more accurate. Figure 1 displays the system's flowchart.

2.2 Venue modelling

Most indoor venues are laid out in a ladder-surround layout, with the venue in the middle and the audience in a circle around it. For modelling purposes, the arena can be simplified into two parts: the blue part is the spectator stand, where data collection is required, and the green part is the stadium, where no data collection is required, as shown in Fig. 2. the range of data to be collected by the UAV can be defined using the following Algorithm 1. The main goal of Algorithm 1 is to make the UAV fly over the area where people are sitting, i.e., the blue area in the figure. This is especially important in practical applications, such as in sports stadium surveillance, where we only need to focus on the spectator area, not the whole stadium. This also means that when we plan the flight path of the UAV, we should minimise unnecessary flights and cover only the spectator area to save time. In the simulation experiments in Figs. 6 and 7, it can be clearly seen that the UAV's flight path is planned based on some specific data collection points and it takes only 235.2 s to pass through these collection points, which are usually the key locations in the spectator area, and the UAV flies through these points and collects the required data. Subsequent UAV path planning was also optimised based on these selected points to ensure that the UAV was able to complete the data collection task efficiently.

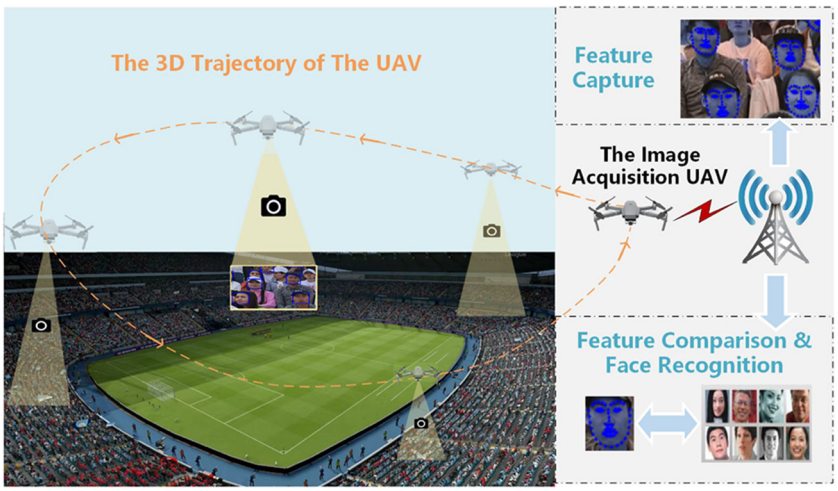

Fig. 1 The system flow chart. After acquiring the 3D information of the target site, the UAV models and determines the collection points and adaptively plans the optimal path. The UAV then lifts off to collect target data at each acquisition point, encodes and packages the acquired face images and sends them to the ground platform, where the ground-based distributed Hadoop platform performs feature comparison and face recognition

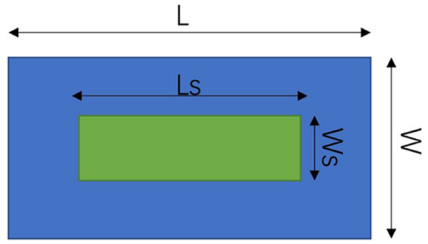

Fig. 2 Simplified venue map. L and W denote the length and width of the spectator stands respectively.Ls and Ws are the length and width of the athletics track respectively

2.3 Trajectory optimization

Assume that there are K shooting points obtained by Algorithm 1. Let the location of the i-th shooting point be $S_i = X_i, Y_i, Z_i, i = 1, \cdots, K$ and the current position of the UAV, at the t-th time slot be $q(t)$. The distance between the UAV and shooting point i at time t is represented by Eq. 1:

$$d_i\,(\mathrm{t}) = \|q\,(\mathrm{t}) - S_i\|\,, 1 \le i \le K\,, \quad 0 \le t \le T\,. \tag{1}$$

Letting the maximum rate of the UAV be V_{max}, we obtain constraint 1 as in Eq. 2. Define $\dot{q}$(t) as the first-order derivative of a time-varying function q(t) with respect to time t.:

$$||\dot{q}\,(t)|| \;\le V_{\max}, \quad 0 \le t \le T\,. \tag{2}$$

When communication is carried out between the UAV and the i-th shooting point, due to the high altitude of the UAV, a line-of-sight communication model is more appropriate constraint 1 and the LoS channel gain model consisting of the signals received at the i-th shooting point can be expressed as Eq. 3:

$$h_i\,(\mathrm{t}) = \lambda_0 d_i\,(t)^{-2}, 1 \le i \le K\,. \tag{3}$$

where λ_0 represents the channel power gain of the line-of-sight link at a unit distance of 1 m, and d(t) denotes the Euclidean distance between the UAV and the i-th shooting point in 3D space.Defining the transmission power at each shooting point as P, the received power of the signal at the i-th shooting point can be expressed as in Eq. 4:

$$\begin{aligned} P_i\,(\mathrm{t}) &= P \times h_i\,(t)\,, \\ &= P \times \lambda_0 d_i\,(t)^{-2}. \end{aligned} \tag{4}$$

Also, defining the bandwidth as B and the Gaussian white noise as N_0, the instantaneous normalized achievable rate can be defined as in Eq. 5:

$$R_i\,(t) = \log_2(1+\frac{P_i\,(t)}{BN_0}) = \log_2(1+\frac{P_i \times \lambda_0 d_i\,(t)^{-2}}{BN_0}). \tag{5}$$

Then the signal-to-noise ratio at the reference distance d = 1m is:

$$\gamma_0 \triangleq \frac{\lambda_0}{BN_0}. \tag{6}$$

Thus, Eq. 5 can be rewritten as:

$$R_i\,(t) = \log_2(1+\frac{\gamma_0 P}{||q\,(\mathrm{t}) - S||^2}). \tag{7}$$

Our goal is to minimize the job completion time T by optimizing the trajectory designof the UAV. Define:

$$Q \triangleq \{q\,(t)\}\,. \tag{8}$$

Algorithm 1 Shooting points choose.

input : Length(**L**), width(**W**) and height(**H**) of the building; Length(**Ls**), width(**Ws**) of the site; The distance moved horizontally is **Xi**, vertically is **Yi**, and horizontally is **Zi**; Set the list **POINT** to store each collection point data, **POINTS** to store all collection point data.
output: Set the UAV departure point as (**0**, **0**, **H**) and the UAV real-time position as (**X**, **Y**, **Z**).

So, UAV needs to collect the following ranges:
if *Yn=(0,...,nYi,...,(W/2 - Ws/2)) &&((W/2 + Ws/2),...,nYi,...,W)* **then**
  Xn=0,…,n**Xi**,…,**L**, **Z**=0,…,n**Zi**,…,**H**;
else
  Xn=(0,…,n**Xi**,…,(**L**/2-**Ls**/2)) && ((**L**/2-**Ls**/2),…,n**Xi**,…,**L**);
end
Collection point definition:
while *0<Y<(W/2 - Ws/2) || (W/2 + Ws/2)<Y<W* **do**
  for *X=0,...,Xi,...L* **do**
    POINT = [**X**, **Y**, **Z**];
    POINTS.append(**POINT**);
    Z = **Z** - **Zi**;
    X = **X** + **Xi**;
  end
end
while *(W/2 - Ws/2) < Y < (W/2 + Ws/2)* **do**
  if *0<X<(L/2-Ls/2) || (L/2+Ls/2)<X<L* **then**
    for *Y =(W/2 - Ws/2),...,Yi,...(W/2 + Ws/2)* **do**
      POINT = [**X**, **Y**, **Z**];
      POINTS.append(**POINT**);
      Z = **Z** - **Zi**;
      X = **X** + **Xi**;
    end
  end
end

Then theoptimization problem can be formulated as:

$$(P1) \quad \min_{T,Q} \quad T$$

$$B\int_0^T R_i(t)\,dt \geq C_i, \qquad 1 \leq i \leq K, \tag{9}$$

$$||\dot{q}(t)|| \leq V_{\max}, \qquad 0 \leq t \leq T, \tag{10}$$

$$q(0) = q(t). \tag{11}$$

where Eq. 9 is the amount of data C_i that the UAV can receive per shooting point, Eq. 10 is a limit on the maximum speed of the UAV, and Eq. 11 is a limit on the initial and final locations of the UAV, i.e., the UAV should return to its initial location after finishing the data collection mission.Since time t is a continuous function, the variable Q in problem $(P1)$ is infinite-dimensional. Also, the issue is non-convex because $R_i(t)$ in Eq. 9 is not concave with respect to $Q(t)$. In addition, the variable T acts as an upper constraint on the integration interval, and it is sometimes challenging to solve the issue directly in the absence of closed form expressions. In the subsequent section, we will therefore refer to the analysis in [21] to approximate the problem.

Given the close coupling between the variables T and $q(t)$, we may add an intermediary variable η to the problem (P1) so that the variables T and $q(t)$ can be optimized in turn. Consider the following optimization issue for a certain T:

$$(P1.1) \quad \max_{\eta,Q} \quad \eta$$

$$\frac{B}{C}\int_0^T R_i(t)\,dt \geq \eta, \qquad 1 \leq i \leq K, \tag{12}$$

$$||\dot{q}(t)|| \leq V_{\max}, \qquad 0 \leq t \leq T, \tag{13}$$

$$q(0) = q(t). \tag{14}$$

Assume that $\eta^*(T)$ is problem $(P1.1)$'s ideal solution. Therefore, $\eta^*(T) \geq 1$ is naturally comparable to the constraint (9) of problem $(P1)$. Problem $(P1)$ may be equivalently expressed as:

$$(P1.2) \quad \min_{T} \quad T$$

$$\eta^*(T) \geq 1, \qquad 0 \leq t \leq T. \tag{15}$$

where $\eta^*(T)$ is a function that increases monotonically with respect to T. An optimal solution exists for T^* in problem $(P1.2)$ when $\eta^*(T^*) = 1$ is satisfied. Also, we can search for T satisfying the condition $\eta^*(T) = 1$ for any given T using the dichotomy method. Therefore, we are committed to solving the problem $(P1.1)$ by obtaining $\eta^*(T)$ for a given T. To simplify the analysis process, we discretize the continuous time T into N time lengths $\delta = T/N$, then T can be expressed as $[t1, ..., tN]$, where $tn = n, n = 1, ..., N$. As a result, the UAV's trajectory at time T may be discretized as:

$$\mathrm{q}[n] = q(t_n), n = 1, \cdots, N. \tag{16}$$

Its velocity can be discreted as:

$$\|\dot{q}(n)\| = \frac{\|q[n+1] - q[n]\|}{\delta}, n = 1, \cdots, N-1. \tag{17}$$

Similarly, Eq. 5 can be discretized as:

$$R_i[n] = \log_2\left(1 + \frac{\gamma_0 P_i}{\|q[n] - S_i\|^2}\right). \tag{18}$$

Finally, after defining $Q = \{q[n]\}$, problem $(P1.1)$. can be formulated as both a discretization of problem $(P1)$:

$$(P1.3) \quad \max_{\eta,Q} \quad \eta$$

$$\frac{B\delta}{C_i}\sum_{n=1}^{N} R_i[n] \geq \eta, \qquad 1 \leq i \leq K, \tag{19}$$

$$\frac{\|q[n+1] - q[n]\|^2}{\delta^2} \leq {V_{\max}}^2, \qquad n = 1, \cdots N-1, \tag{20}$$

$$q[1] = q[n]. \tag{21}$$

Notably, constraint (19) is nonconvex. However, $R_i[n]$ is convex with respect $q[n]$. Then, according to the successive convex approximation (SCA) technology, we transform constraint (19) to a convex one by introducing a concave lower-bound of $R_i[n]$. For a given trajectory $Q^l \triangleq \{q^l[n]\}$, $R_i[n]$ can be lower bounded by $\hat{R}_i[n]$ with:

$$\hat{R}_i[n] \triangleq \log_2\left(1 + \frac{P\gamma_0}{z_i^l[n]}\right) - \phi_i^l[n](z_i[n] - z_i^l[n]), \tag{22}$$

where $z_i[n] = \|q[n] - S_i\|^2$, $z_i^l[n] = \|q^l[n] - \S_i\|^2$, and $\phi_i^l[n] = \frac{P\gamma_0 \log_2 \mathrm{e}}{z_i^l[n](z_i^l[n] + P\gamma_0)}$.

Subsequently, we get a lower bound of the optimal value of problem (P1.3) via solving the following approximate problem

$$(P1.4) \quad \max_{\eta,Q} \quad \eta$$

$$\text{s.t.} \quad \frac{B\delta}{C}\sum_{n=1}^{N} \hat{R}_i[n] \geq \eta, \ 1 \leq i \leq K, \tag{23}$$

$$\frac{\|q[n+1] - q[n]\|^2}{\delta^2} \leq V_{\max}^2, n = 1, \cdots, N-1, \ \ q[1] = q[N],$$

where the objective function and all constraints are convex. Hence, problem (P1.4) is a convex problem, which can

be solved by standard convex optimization techniques such as the CVX toolbox. Notice that, the optimal solution of problem (P1.4) is a lower bound of that of problem (P1.3). Therefore, we iteratively solve problem (P1.3) multiple times to improve the quality of the solution. In summary, we provide the algorithm for solving problem (P1.3) in Algorithm 2.

Algorithm 2 Algorithm for (P1.3) based on SCA.

input : Given T, initial trajectory of the Ferry UAV Q^0, prescribed thresholds $\varepsilon_1 > 0, \varepsilon_2 > 0, l = 0$.
output:
while $\frac{\eta^{l+1}-\eta^l}{\eta^l} \geq \varepsilon_2$ **do**
  Set the inner initial trajectory $Q^{l,0}$=Q^l and the inner iterative index $r = 0$ to their starting values;
  while $\frac{\eta^{r+1}-\eta^r}{\eta^r} \geq \varepsilon_1$ **do**
    For given $Q^{l,r}$, obtain $Q^{l,r+1}$ and η^{r+1} by solving problem (P1.3);
    r = r + 1;
  end
  $Q^{l+1} = Q^{l,r}$, $\eta^{l+1} = \eta^r$;
  $l = l$+1;
end

Then, for each collecting point, we provide a spherical area with a radius r that is centered on that collection point in order to create a workable beginning trajectory. Then, via designing the UAV trajectory and radius r properly, we hope to minimize the UAV traveling distance while the UAV is guaranteed to reach each spherical region. S_i represents the location of the UAV. You may express the problem $(P2)$ as follows.

$$(P2) \quad \min_{r,q(t),T_{tr}} \quad T_{tr}$$

$$\min \ ||q(t) - S_i|| \leq r, \quad 0 \leq t \leq T_{tr}, \tag{24}$$

$$||q(t)|| \leq V_{\max}, \quad 0 \leq t \leq T_{tr}, \tag{25}$$

$$q(0) = q(T_{tr}). \tag{26}$$

As demonstrated in Fig. 3, the UAV may go around all spheres made up of collecting points with a radius of r since (22) has at least one point in time t that will limit the distance between the UAV and the collection point to be no greater than $(P2)$.

Using the optimum value of the specified problem $(P2)$, it can be inferred that is a decreasing function with respect to r for a fixed radius $(P2)$. In order to solve $(P2)$ using a given r and acquire the corresponding, one can do so. Then, one can use a dichotomy to choose the best r^* =T. Since the ideal trajectory q(t) only consists of linked line segments, it is possible to reduce $(P2)$ in order to determine the ideal ordering by optimizing the path points inside the sphere for any radius r. These path points are actually the beginning

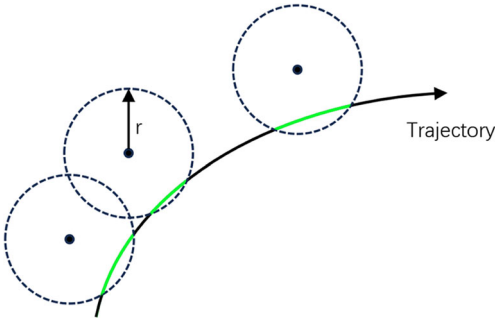

Fig. 3 UAV path diagram. The black arcs in the figure indicate the trajectory of the UAV, where the green line segments indicate the portion of the UAV that intersects the acquisition point. The black circular dashed line indicates the acquisition range of the acquisition point

and ending locations of the line segments. Define as the path points within the sphere, then the flight time of the UAV is:

$$T_{tr}(\{g_i\}, \pi) = \frac{\sum_{i=1}^{K-1} ||g_{\pi(i+1)} - g_{\pi(i)}|| + ||g_{\pi(K)} - g_{\pi(1)}||}{V_{\max}}. \tag{27}$$

Thus, the problem (P2) can be reduced to:

$$(P2.1) \quad \min_{g_i,\pi} \quad T_{tr}(\{g_i\}, \pi) \tag{28}$$

$$||g_i - Si|| \leq r. \tag{29}$$

A suboptimal solution of $(P2)$ can be efficiently obtained by the standard convex optimisation technique with $\pi = \hat{\pi}$. For a given flight time T, the initial trajectory of the UAV can be designed as Algorithm 3:

By effectively using the typical convex optimisation approach with $\pi = \hat{\pi}$, a suboptimal solution to $(P2)$ may be achieved. The starting trajectory of the UAV can be planned as follows for a specified flight period T:

Algorithm 3 Design of the initial trajectory for a given T.

input : A given T, locations of UAV $\{S_i\}$.
output:
Solve TSP to get the shortest trip time T_{tsp} and the best visiting order. $\hat{\pi}$ based on $\{S_i\}$.
let $r_l = 0$, r_u be sufficiently large and tolerance $\varepsilon > 0$.
while $|T_{tr} - T| \leq \varepsilon$ **do**
  $r = (r_l + r_u)/2$;
  For issue (P2.1), use the visiting order $\hat{\pi}$ to calculate the journey time T_{tr} and the waypoints $\{g_i\}$;
  if $T_{tr} > T$ **then**
    $r_l = r$;
  else
    $r_u = r$;
  end
end
Construct the initial trajectory based on $\{g_i\}$.

2.4 Airborne information gathering

The goal of the well-known computer vision problem of face identification is to locate each face precisely in an image or video. Early face detectors could find faces in the form of sliding windows using cascade classifiers. These approaches then used more effective manual features [22–24] and potent classifiers [25, 26], yet they were still unable to produce results that were accurate enough. In line with CNN's progress, two-stage detectors, which are often based on anchors, now create region recommendations from the picture in addition to the final bounding box. Faster R-CNN [27] is a typical two-stage detector, and [28] when applied to faces, it produces positive results in object localization but at the expense of sluggish detection time and extremely high processing resource needs. Real-time detection is challenging to implement on CPU or ARM systems. We believe that the slight increase in accuracy at the expense of running speed is unsuitable for real-world applications. As a result, first-class face detectors dominate today. We have analysed the first four methods on the WIDER face validation set and show their mAP, model size and number of parameters in Table 1. These networks use either Resnet50 or VGG16 as backbone and combine operations such as long-edge pull-up and the use of multi-scale transforms, resulting in large model sizes, large number of parameters and low computational efficiency. While these methods can achieve extremely high APs, they require extremely high arithmetic power for the device, as well as low detection rates, and may be able to achieve better results when used in large GPU servers. However, due to the energy, load and other limitations of UAVs, the computational power and memory storage are at a low level, in which case a lightweight face detector with acceptable accuracy and speed is required.

2.4.1 Backbone

The factor that has the greatest impact on the detection network is the choice of backbone. The most popular lightweight CNNs are MobileNets, which concentrate on mobile or embedded devices. They have fewer parameters and operations while maintaining a similar level of accuracy as conventional convolutional neural networks [32].

The most recent official installment in the MobileNets series is MobileNetV3 [33]. Based on MobileNetV2, the authors have made many efforts in recent years to optimise the network speed, reduce the network complexity and improve the detection accuracy, in addition to making some improvements to the backbone network structure and activation function. The network has now achieved a higher degree of detection speed and accuracy thanks to these optimizations. The following are enhancements made to MobileNetV3 over the previous version.

The authors add an attention mechanism to the core architecture, known as SENet (Squeeze-and-Excitation) [34]. This neural network's main goal is to enhance the quality of the representation it generates by explicitly modeling the relationships between its convolutional feature channels. It specifically learns to automatically determine the value of each feature channel and utilizes the outcome to boost features that are beneficial and suppress features that are less beneficial for the job at hand. Through this method, the network learns to utilize global information to suppress less valuable traits and selectively emphasize informative ones. Additionally, the authors reduce the channel of the expansion layer in the structure that contains SE to 1/4 of its original size, which increases accuracy without increasing time consumption. In addition, the authors also reduce the number of convolutions in the first convolutional layer without losing accuracy, so that MobileNetV3 can save about 2ms of computing time with the same accuracy.

In addition, [33] found that the newly proposed s-wish activation function can effectively improve the network accuracy compared to the traditional ReLU activation function, as follows:

$$s - wish = x \cdot \sigma\left(x\right). \tag{30}$$

When the sigmoid activation function is σ and is as follows:

$$\sigma\left(x\right) = \frac{1}{1 + e^{-x}}. \tag{31}$$

The S-function is more computationally expensive on mobile devices and incurs non-zero costs in embedded environments, which has a greater impact on computational

Table 1 Example of a lengthy table which is set to full textwidth

Model	Model size	No. Parameter	mAP (%)		
			Easy	Medium	Hard
RetinaFace [29]	458M	141.38M	0.955	0.940	0.844
Yolov5n [18]	356.4M	46.627M	0.958	0.943	0.861
PyramidBox [30]	218M	57.18M	0.961	0.950	0.899
DSFD [31]	458M	141.38M	0.966	0.957	0.909

efficiency and also some unexpected problems in the quantization process when applied to the UAV platform, despite the fact that this nonlinear activation function can effectively improve the network's accuracy. Based on the aforementioned factors, the authors suggest the h-swish activation function, which uses the h-sigmoid activation function rather than the sigmoid activation function to mainly resolve the issues of computing cost and unfavorable quantization. H-swish activation function is as follows:

$$h - switch\,[x] = x \cdot \frac{\mathrm{Re}\,LU6\,(x+3)}{6}. \tag{32}$$

2.4.2 Network architecture

In convolutional networks, the deep network is easy to characterise semantic features and the shallow network is easy to characterise image features. However, in target detection, although deep networks can characterize semantic features, they do not possess much geometric information due to the small size of Feature Map, which is not conducive to target detection, while shallow networks contain more geometric information but not many semantic features of images, which is not conducive to image classification. In [35], the idea of FPN (feature pyramid networks) was proposed to achieve the effect of feature fusion by increasing the up-sampling process between different network layers and then connecting them with the features extracted from the shallower layers. This operation is conducive to extracting features at different scales and achieving increased detection accuracy of the network without sacrificing memory and speed. Additionally, classic two-stage detectors increase the scope of the proposal to gather contextual information at the price of model size and speed in order to achieve greater detection accuracy. The idea of SSH (Single Stage Headless) was proposed in [36], which can achieve similar results by simply stacking the conv layers and does not have a large impact on the size and accuracy of the model.

According to the above analysis, we designed an anchorless face detection network considering the restricted load and arithmetic power of UAV, and the network architecture is shown in Fig. 4. We used a 9-block MobileNetV3-Small network as the backbone. Considering that H-swish can obtain better results only in deeper layers of the network, we used the ReLU6 activation function for the first 3 layers of the network, and the H-swish activation function with SE blocks for layers 6-9. Regarding the neck module, we extracted the output data (S4, S8, S16, S32) at layers 1,3,8,9 of the network respectively and performed FPN operations by upsampling S32 and processing it through 3×3 convolution and BN layers and activation functions (i.e. Conv, BN, Activation) to obtain the feature of the layer The map is then added to the feature map of S16 by the same CBA process, and the result is upsampled again and added to S8. The result is then

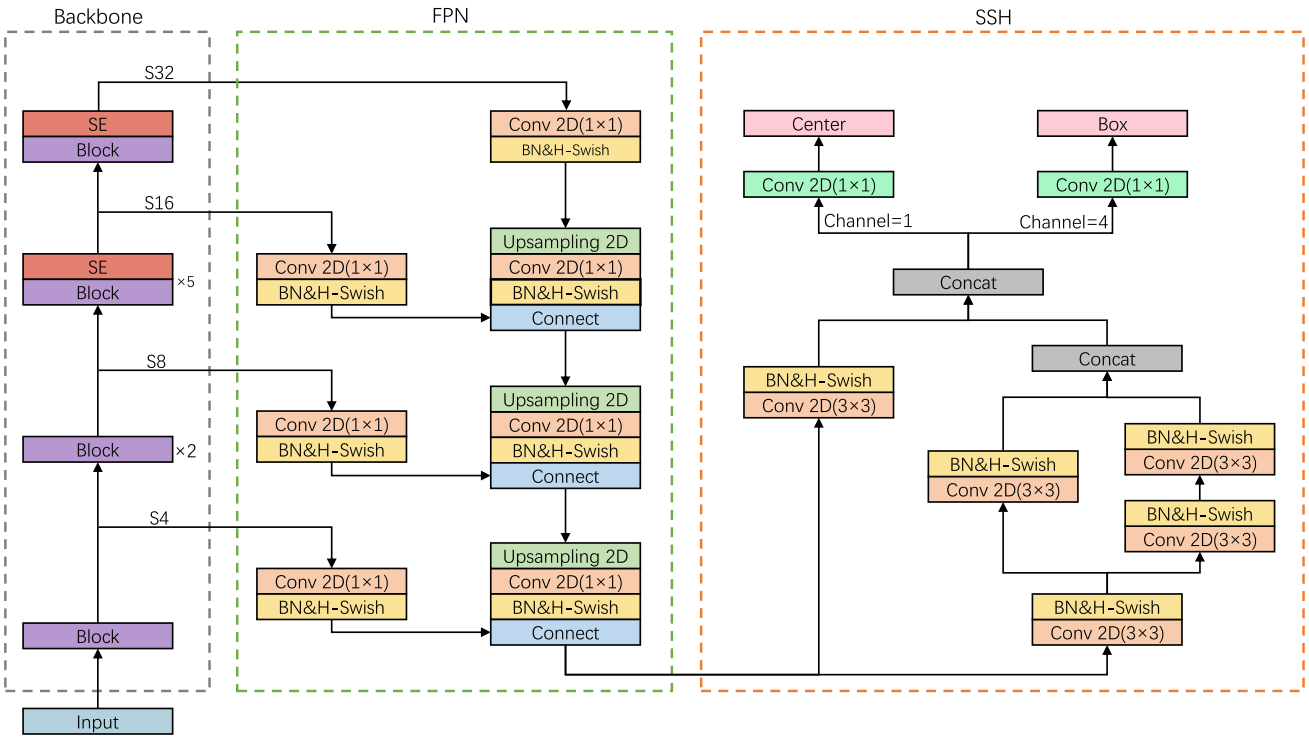

Fig. 4 Network Architecture. The backbone is a 9-block MobileNetV3-Small, the FPN neck module superimposes the feature maps of the 4 layers, the SSH module superimposes the final feature maps, and finally the head module obtains the facial box part by logistic regression

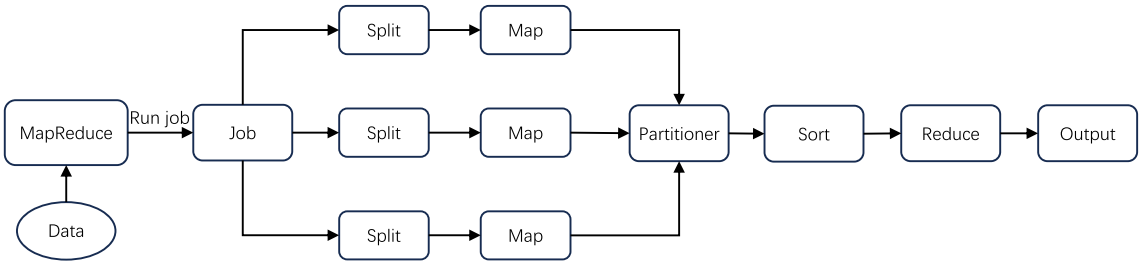

Fig. 5 MapReduce Running Flowchart. MapReduce is a distributed data processing model that breaks large datasets into smaller, equal-length chunks of data for parallel processing.Map tasks process each chunk of data independently and generate the output data (intermediate data). Partitioner decides in which partition to place each chunk of intermediate data, and the Reduce task combines these computations into a final output. Sorting can be done before or after the Reduce step

up-sampled again and summed with S8. Finally, the four layers of feature maps are superimposed to maximise the extraction of features at different scales. The resulting feature map is then fed into the SSH module, and after extracting the features at different scales using Conv, BN and Activation, the features at different scales are then concatted to obtain the final feature map, in order to increase the effective perceptual field and obtain a feature map with stronger characterisation capability. For the head module, a Gaussian heat map of the face box centroid is extracted using different convolutions and the face box part is obtained by logistic regression.

2.5 Ground data processing

Due to the huge computational power and memory requirements of the large batch of face comparison work, it would be more effective to have the work processed by a ground station because of the high energy consumption and slow processing speed of UAV. The design of the MapReduce framework is based on the excellent Hadoop platform, which has powerful parallel processing capabilities, as well as its easy expansion, high computational power, high fault tolerance and low price, etc. The outstanding Hadoop platform, which has strong parallel processing capabilities, simple extension, high computing power, high fault tolerance, and cheap cost, etc., forms the foundation for the MapReduce framework's architecture. The MapReduce framework is well suited for the case in this study because of its cheap hardware needs, high fault tolerance, ease of writing, ease of expansion, and high throughput. Figure 5 depicts the MapReduce framework's processing flow.

In the MapReduce framework, the image feature library read from HDFS is first sliced, not in the actual sense of slicing, but in the sense of recording the path, ID number and the corresponding encoding of the image data to be processed, where the encoding is the feature value of the image. The key value Key represents the position of the image feature in the sliced segmentation, and the key value is the feature value in the form of the image code. The map function then calls these Key/Value values, determines whether the image features meet the matching criteria, and stores the intermediate results in the local system as Key/Value key-value pairs, where the Key value represents the similarity and the Value represents the corresponding built-in image in the image feature library. Next, the intermediate results from the Map task are collated and passed to the Reduce task. Finally, the Reduce task receives the intermediate results from the Map and sorts them by similarity and stores the sorted results in HDFS.

3 Experimental simulation

3.1 Trajectory optimization simulation

Taking the Russell Stadium as an example, a 500m long by 400m wide by 80m high arena, all the collection points were visualised after simulation calculations using MATLAB tools, as shown in Fig. 6:

Taking the acquisition point shown in Fig. 7 as an example, with P1 and P2 as constraints, and setting the maximum flight speed of the UAV at 50m/s, the path trajectory was simulated using MATLAB and the acquisition took a total of 235.2s as shown in Fig. 7.

3.2 Face detection simulation

3.2.1 Dataset and data augmentation

The proposed method is trained by using the WIDER FACE benchmark training set.32,203 pictures and 393,703 labeled faces make up the WIDER FACE. There is a lot of size variation in these faces. The most often used benchmark for face detection up to this point has been WIDER FACE. Furthermore, images in each subset are graded to three Since many faces in Hard have only 5*5 pixels, over-seeking hard-set

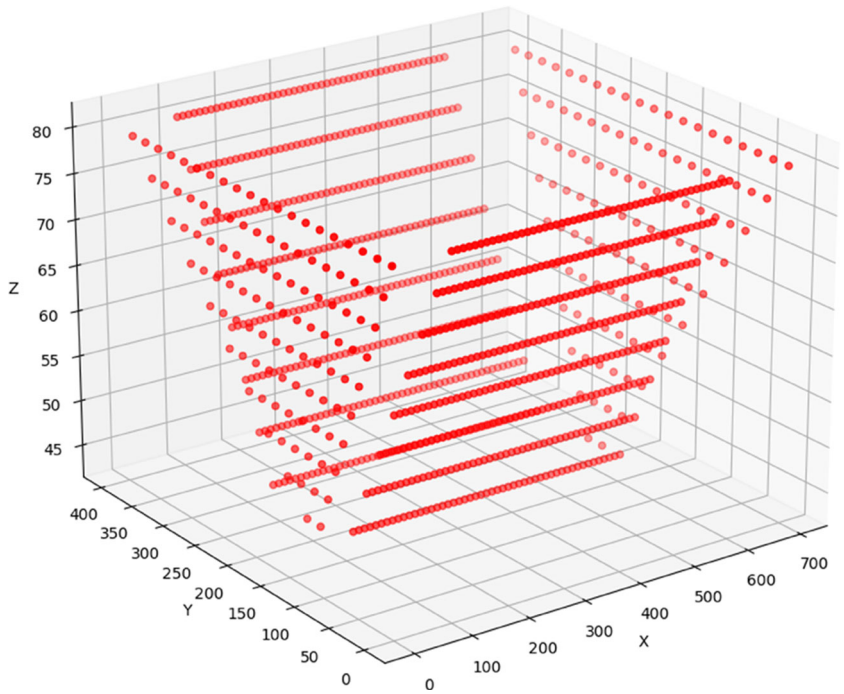

Fig. 6 Collection point simulation. The red dots in the figure indicate the target information that the UAV needs to collect

results can lead to a more sensitive detector, resulting in more false detections. Faces with less than 12 pixels will be ignored and have no effect on Loss, improving the accuracy of hard small targets while ensuring the detection of large targets. Data augmentation plays a crucial role in improving the accuracy and robustness of neural networks. We mainly use color jittering, flip, random cropping, and scale transformation expansion to achieve the expansion of the number and distribution pattern of the original data.

3.2.2 Loss function

For face classification and box regression, the loss function is divided into two parts. For face classification, due to the imbalance of sample classes, only a small portion of an image may contain the target, the negative samples are over-represented and take up most of the total loss, and the negative samples are mostly easily classified, thus making the optimization direction of the model not as we would like it to

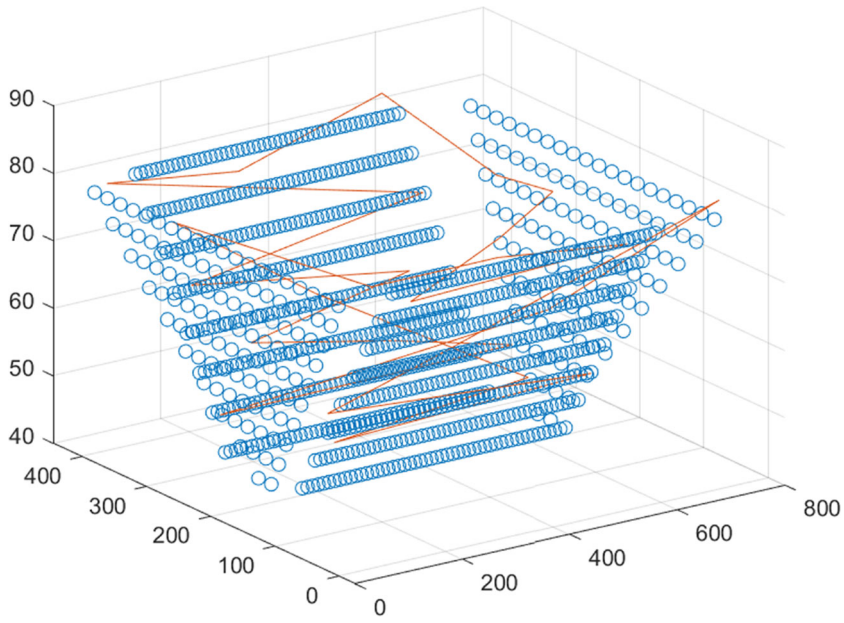

Fig. 7 UAV flight path simulation. The orange curve in the figure shows the path trajectory of the UAV's operation

be. In [37], by altering the normal cross-entropy loss, Focal Loss was created to solve this issue. It enables the model to concentrate more on the difficult-to-classify data during training by decreasing the weights of the easy-to-classify samples. The loss function is as follows.

$$FL\left(p_t\right) = -(1-p_t)^{\lambda}\log\left(p_t\right). \tag{33}$$

To reduce the impact of small faces on Loss, we add a parameter mask to Focal Loss. mask will be set to 0 when the face is less than 12 pixels, otherwise it will be 1 to achieve ignoring small faces, the function is as follows.

$$FL\left(p_t\right) = -(1-p_t)^{\lambda}\log\left(p_t\right) * mask. \tag{34}$$

For box regression, IoU Loss has the best optimisation and we have chosen its latest version, CIoU Loss [38], to optimise the position of the predicted box in relation to the actual box with the following equation.

$$L_{CIOU} = 1 - IoU\left(A, B\right) + \frac{\rho^2(A_{ctr}B_{ctr})}{c^2} + a\cdot v, \tag{35}$$

$$v = \frac{4}{\pi^2}\left(\tan^{-1}\frac{w^{gt}}{h^{gt}} - \tan^{-1}\frac{w}{h}\right), \tag{36}$$

$$a = \frac{v}{(1-IoU)+v}. \tag{37}$$

Included intersection over union (IoU) is calculated as follows.

$$IoU\left(A, B\right) = \frac{A\cap B}{A\cup B}. \tag{38}$$

3.2.3 Training parameters

We used a pre-trained model of MoileNetV3-Small, the Adam optimiser, with a dynamic learning rate scheduling strategy, with an initial learning rate of 1e-3 and gradually decreasing to 1e-5. The Pytorch 1.8 framework was used for training on an Ubuntu 18.04 server platform, using a 3090 graphics card. All training data were obtained from the training set in WIDER FACE. The batch size for each training round was 18, and a total of 120 iterations were performed, taking about 10 h.

3.3 Experimental results

UAV target detection was performed using the trained target detection model, which was not tested using training data for this experimental setup, as shown in Fig. 8. For the test targets studied there was a dense distribution of people, a large number of people, different focal lengths of the targets at different locations, the presence of partial occlusion and a large degree of light variation, which are a few of the important issues that have emerged in UAV target detection.

Our FPN design gives the network a larger perceptual field by superimposing the feature maps extracted from various layers, as is evident in the test images, where partial occlusion and focal defocus are clearly resolved. Additionally, the network has a low false detection rate, which prevents a large number of false images from clogging up system resources.

3.4 Distributed face recognition

After receiving a certain amount of face encoding data on the Hadoop platform, we chose to conduct test experiments with the same amount of data, the same recognition method, and the same specific target, and to compare them. In Fig. 9, the comparative findings are displayed. Compared to traditional single-threaded face recognition using the face_recognition library, the improvement in efficiency using Hadoop can reach over 90%, and the improvement becomes more pronounced the more images there are. Even with multi-threading, face recognition using the face_recognition library with 16 threads still lags behind the Hadoop platform by about 50%, but 16 threads places extremely high demands on the performance of the device hardware, whereas Hadoop, as a distributed platform, places very low demands on the performance of the device hardware. During the experiments, we found that Hadoop spends most of its time in the process of task division and reorganisation when performing distributed computation, for example, for 10,000 images, the computation time is only 6.3s, and the remaining 103 seconds are spent in task allocation, which can be further improved by improving the device hardware of each node.

3.5 Analysis of model

Considering the lack of load and arithmetic power of the UAV and the fast-flying nature of the face detection module, it should be lightweight, efficient, and accurate. In subsequent experimental verification, we will use mAP as an indicator to confirm the accuracy of face detection technology, model size and parameter count as indicators to confirm the complexity of the model, and FLOPs and FPS as indicators to confirm the effectiveness of the model.

Regarding accuracy and lightweight validation, considering the actual usage, we did not perform multi-scale transformations and long-edge pull-ups on the images in the tests, but directly used the original images from the validation set in WIDER FACE for single-scale input to do inference tests, and used the official MATLAB script provided for PR plotting, as shown in Fig. 10. Second, comparisons were done in terms of model size, FLOPs, and the number of parameters.

Fig. 8 UAV target detection effects. Our network successfully detects the faces of the vast majority of viewers in the figure, and the undetected faces lack features. They can be directly ignored in real applications

Due to the UAV's limited memory capacity, it is important to take the face detector's memory utilization into account. The size of the model and the number of parameters are closely connected. Fewer parameters do not, however, imply less processing. According to [39], we used FLOPs to measure computations with a resolution of 640 × 480. The data are displayed in Table 2. It is clear that our technique uses less resources while yet achieving a good degree of identification accuracy. We determined the comparative values for different approaches in terms of model size, number of parameters, Floating Point Operations per Second (FLOPs), and mean average accuracy (mAP), using the metrics from the BlazeFace [17] method as the baseline. Compared to methods such as LFFD [13], Yolov5n [18], and our approach considerably cuts down on the size and number of parameters in the model. After reducing the model size, our method can achieve performance levels similar to or even better than BlazeFace and other methods in terms of computational complexity, and in some cases, it even surpasses these methods.

We have tested the speed of model detection. We used Jetson Xavier NX, an AI edge computing device from NVIDIA, as the test platform, which is widely used in

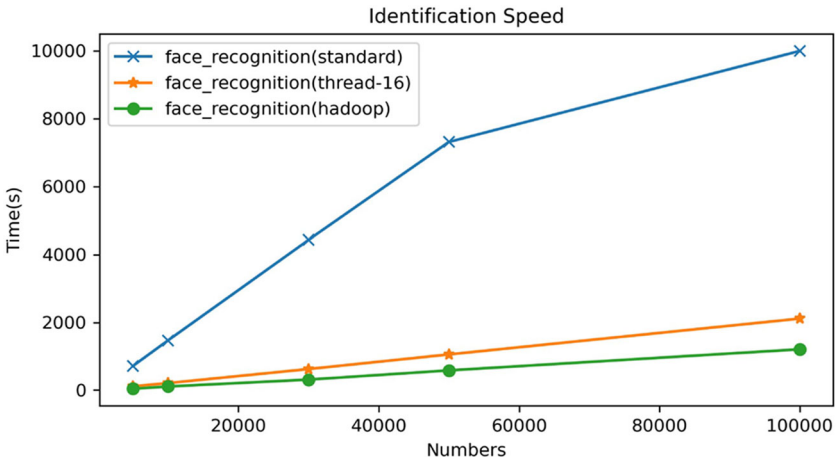

Fig. 9 Face recognition time comparison graph based on Hadoop. The horizontal axis represents the amount of data and the vertical axis represents the processing time. The blue line in the figure indicates the recognition rate using a traditional face recognition library with a single thread. The orange line indicates the recognition rate with 16 threads, and the green line indicates the recognition rate using the Hadoop platform. The advantage of the Hadoop platform in terms of recognition rate can be clearly seen

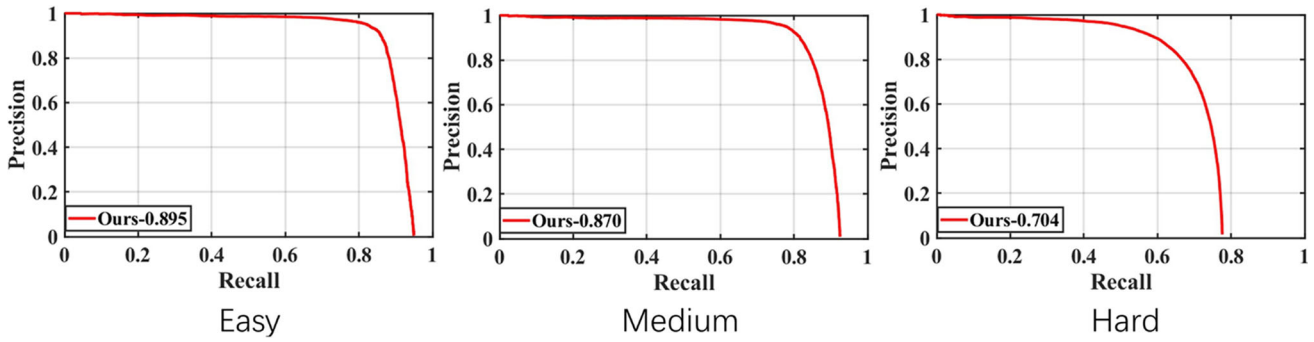

Fig. 10 Evaluation results on WIDER FACE. Easy, Medium and Hard are the three different detection difficulties in the WIDER FACE validation set.Our method exhibits superior detection performance in all three difficulties

high-performance AI systems such as drones, small commercial robots, smart cameras and other IoT embedded systems, thanks to the very low power consumption of the arm architecture (20W) and the high performance of the volta GPU (384 CUDA cores). We tested using batchsize 1 and some common resolutions, using Tensorrt for inference acceleration, for all models using the same parameters transformed into onnx models for a fair comparison, and 100 rounds of inference speed tests with the same settings using the same images, taking the average speed as the comparison standard. The final results are shown in Table 3.

In Table 3, we have chosen to compare three lightweight models that are more friendly to edge devices with lower computational power. Considering that our practical task requires feature extraction of the detected face images, too low a resolution would lead to difficulties in extracting valid recognition feature points in the face, we have chosen to use 640×640 as the minimum detection resolution. Clearly, LFFD and YoloV5n-Face are more difficult to process for large input sizes, and the detection speed is difficult to meet the needs of practical applications. Compared to BlazeFace, the efficiency of our network decreases more slowly as the input image size increases, and our network has the best efficiency at input sizes larger than 2048*1152. To further demonstrate the efficiency advantage of our proposed network, the network efficiency calculation proposed in accordance with [13] is as follows:

$$E_{net} = {}^{FLOPs}/_{t}. \tag{39}$$

where the running time is indicated by t. The network's computational efficiency, E_{net}, is reflected by its size (the greater, the more efficient), and it may be calculated at a certain resolution on a particular platform. We calculated this metric separately for YoloV5n-Face, BlazeFace and Ours networks with an input of 1152 x 2048 (BlazeFace vs. YoloV5n-Face vs. Ours):

0.06G/ms vs. 0.15G/ms vs. 0.21G/ms on Jetson Xavier NX

Clearly, the network we designed has more efficient computation, demonstrating the superiority of concise network design.

Combining the results in Tables 2 and 3 shows that the simpler network design helps to improve the inference speed of the model on the less powerful edge devices, with the Jetson Xavier NX as the piggyback platform, the speed increase is about 50% compared to the most advanced YoloV5n-Face at common resolutions, with a very small performance loss of

Table 2 Model size, number of parameters, FLOPs and mAP

Model	Model Size	No. Parameter	FLOPs (640*480)	mAP (%)		
				Easy	Medium	Hard
BlazeFace [17]	0.78M (1)	0.175M (1)	1.39G (1)	0.877 (1)	0.816 (1)	0.639 (1)
LFFD [13]	9M (11.538)	2.16M (12.343)	9.25G (6.655)	0.910 (1.038)	0.881 (1.079)	0.780 (1.221)
Yolov5n [18]	1.726M (2.213)	13.7M (78.286)	2.11G (1.518)	0.936 (1.067)	0.915 (1.121)	0.805 (1.260)
Ours	1.3M (1.667)	0.325M (1.857)	1.94G (1.396)	0.895 (1.020)	0.870 (1.066)	0.704 (1.102)

Table 3 Model size, number of parameters, FLOPs and mAP

Model	640*640	1024*1024	2048*1152	2048*2048
LFFD	29.07FPS	12.16FPS	5.43FPS	2.03FPS
YoloV5n-Face	45.09FPS	19.69FPS	9.26FPS	4.50FPS
BlazeFace	**76.92FPS**	**31.25FPS**	13.16FPS	3.49FPS
Ours	71.43FPS	30.12FPS	**14.08FPS**	**4.90FPS**

The bold numbers represent the optimal detection performance, further confirming the efficiency advantage of our proposed network

about 5%. Compared to the most miniaturised BlazeFace, the detection speed is almost the same at common resolutions, with a modest performance gain of around 6%. Our network strategy results in a relatively excellent balance of detection speed and accuracy for the model, with optimal utility in the application scenario of UAV for dense target detection.

4 Conclusions

In this paper, we have investigated the target detection for a crowded and vast region. By optimizing the 3D trajectory of the UAV and leveraging the ARM-based edge computing device, we have proposed a UAV aided lightweight target information collection and detection approach. Numerical simulation results have shown that the proposed framework admits a faster detection speed with almost the same detection accuracy, which can effectively achieve fast target detection in crowded scenarios. This work makes a workable and practical suggestion for how to locate a specific target object in a crowded and vast region. In the future, we will study multiple UAVs aided target detection in crowded scenarios.

Acknowledgements The National Natural Science Foundation of China is funding JianShan Su under Grant No. 62165015. The Autonomous Region Graduate Student Innovation Program is funding Meng Huang under Grant No. XJ2023G258

Author Contributions Conceptualization, H.L. and H.Z.; methodology, J.S. and M.H.; software, T.M.; validation, J.S., Y.Z. and T.M.; formal analysis, M.H.; investigation, H.Z.; resources, J.S.; data curation, Y.Z.; writing–original draft preparation, M.H.; writing–review and editing, H.Z.; visualization, M.H.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S.

Funding National Natural Science Foundation of China, 62165015. Autonomous Region Graduate Student Innovation Program, XJ2023G258.

Availability of data and materials Not applicable.

Code Availability Not applicable.

Declarations

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. And we confirm informed consent of study participants was obtained to publish the information/image(s) in an online open access publication.

Competing interests The authors declare no competing interests.

References

1. Chohan UW, Van Kerckhoven S (2023) Activist retail investors and the future of financial markets: understanding YOLO capitalism. Taylor & Francis, ???
2. Chen K, Li H, Li C, Zhao X, Wu S, Duan Y, Wang J (2022) An automatic defect detection system for petrochemical pipeline based on cycle-gan and yolo v5. Sensors 22(20):7907
3. Zeng Y, Zhang R (2017) Energy-efficient uav communication with trajectory optimization. IEEE Trans Wirel Commun 16(6):3747–3760
4. Zhu H, Qi Y, Shi H, Li N, Zhou H (2018) Human detection under uav: an improved faster r-cnn approach. In: 2018 5th International conference on systems and informatics (ICSAI), pp 367–372. IEEE
5. Shen Y, Zhu Y, Kang H, Sun X, Chen Q, Wang D (2021) Uav path planning based on multi-stage constraint optimization. Drones 5(4):144
6. Liu Y, Zhang X, Guan X, Delahaye D (2016) Potential odor intensity grid based uav path planning algorithm with particle swarm optimization approach. Math Probl Eng 2016
7. Huang C, Lan Y, Liu Y, Zhou W, Pei H, Yang L, Cheng Y, Hao Y, Peng Y (2018) A new dynamic path planning approach for unmanned aerial vehicles. Complexity 2018:1–17
8. Zhou X, Gao F, Fang X, Lan Z (2021) Improved bat algorithm for uav path planning in three-dimensional space. IEEE Access 9:20100–20116
9. Ruan W-y, Duan H-b (2020) Multi-uav obstacle avoidance control via multi-objective social learning pigeon-inspired optimization. Front Inf Technol Electron 21(5):740–748
10. Blondel P, Potelle A, Pégard C, Lozano R (2014) Human detection in uncluttered environments: from ground to uav view. In: 2014 13th International conference on control automation robotics & vision (ICARCV), pp 76–81. IEEE
11. Feng Y, Yu S, Peng H, Li Y-R, Zhang J (2021) Detect faces efficiently: a survey and evaluations. IEEE Trans Biom Behav Identity Sci 4(1):1–18
12. Liu Y, Wang F, Deng J, Zhou Z, Sun B, Li H (2022) Mogface: Towards a deeper appreciation on face detection. In: Proceedings

of the IEEE/CVF conference on computer vision and pattern recognition, pp 4093–4102
13. He Y, Xu D, Wu L, Jian M, Xiang S, Pan C (2019) Lffd: a light and fast face detector for edge devices. arXiv:1904.10633
14. Zhang S, Zhu X, Lei Z, Shi H, Wang X, Li SZ (2017) S3fd: Single shot scale-invariant face detector. In: Proceedings of the IEEE international conference on computer vision, pp 192–201
15. Chi C, Zhang S, Xing J, Lei Z, Li SZ, Zou X (2019) Selective refinement network for high performance face detection. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 8231–8238
16. Liu Y, Tang X, Han J, Liu J, Rui D, Wu X (2020) Hambox: Delving into mining high-quality anchors on face detection. In: 2020 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 13043–13051 . IEEE
17. Bazarevsky V, Kartynnik Y, Vakunov A, Raveendran K, Grundmann M (2019) Blazeface: sub-millisecond neural face detection on mobile gpus. arXiv:1907.05047
18. Qi D, Tan W, Yao Q, Liu J (2023) Yolo5face: Why reinventing a face detector. In: Computer vision–ECCV 2022 workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V, Springer, pp 228–244
19. Kim N, Kim J-H, Won CS (2022) Fafd: fast and accurate face detector. Electronics 11(6):875
20. Wu W, Peng H, Yu S (2023) Yunet: a tiny millisecond-level face detector. Mach Intell Res 1–10
21. Sun R, Matolak DW (2016) Air-ground channel characterization for unmanned aircraft systems part ii: hilly and mountainous settings. IEEE Trans Veh Technol 66(3):1913–1925
22. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
23. Zhu Q, Yeh M-C, Cheng K-T, Avidan S (2006) Fast human detection using a cascade of histograms of oriented gradients. In: 2006 IEEE Computer society conference on computer vision and pattern recognition (CVPR'06), vol. 2, pp 1491–1498. IEEE
24. Yang B, Yan J, Lei Z, Li SZ (2014) Aggregate channel features for multi-view face detection. In: IEEE International joint conference on biometrics, pp 1–8. IEEE
25. Brubaker SC, Wu J, Sun J, Mullin MD, Rehg JM (2008) On the design of cascades of boosted ensembles for face detection. Int J Comput Vis 77:65–86
26. Pham M-T, Cham T-J (2007) Fast training and selection of haar features using statistics in boosting-based face detection. In: 2007 IEEE 11th International conference on computer vision, pp 1–7. IEEE
27. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. Neural Inf Process Syst 28
28. Jiang H, Learned-Miller E (2017) Face detection with the faster r-cnn. In: 2017 12th IEEE International conference on automatic face & gesture recognition (FG 2017), pp 650–657. IEEE
29. Deng J, Guo J, Ververas E, Kotsia I, Zafeiriou S (2020) Retinaface: single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5203–5212
30. Tang X, Du DK, He Z, Liu J (2018) Pyramidbox: a context-assisted single shot face detector. In: Proceedings of the european conference on computer vision (ECCV), pp 797–813
31. Li J, Wang Y, Wang C, Tai Y, Qian J, Yang J, Wang C, Li J, Huang F (2019) Dsfd: dual shot face detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5060–5069
32. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. Preprint arXiv:1704.04861
33. Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, et al (2019) Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324
34. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
35. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
36. Najibi M, Samangouei P, Chellappa R, Davis LS (2017) Ssh: single stage headless face detector. In: Proceedings of the IEEE international conference on computer vision, pp 4875–4884
37. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
38. Zheng Z, Wang P, Ren D, Liu W, Ye R, Hu Q, Zuo W (2021) Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE Trans Cybern 52(8):8574–8586
39. Khan A, Aftab F, Zhang Z (2019) Bicsf: Bio-inspired clustering scheme for fanets. IEEE Access 7:31446–31456