

# 基于共轴双旋翼无人机的目标检测与薄弱位置定位系统设计\*

冯航涛<sup>1,2</sup>, 曾少锋<sup>1,2</sup>, 张璐<sup>1,2</sup>, 杨旭<sup>1,2,3</sup>, 刘智勇<sup>1,2,3†</sup>

(1 中国科学院大学人工智能学院, 北京 100049; 2 中国科学院自动化研究所复杂系统管理与控制国家重点实验室, 北京 100190; 3 中国科学院脑科学与智能技术卓越创新中心, 上海 200031)

(2021年11月1日收稿; 2022年3月9日收修改稿)

冯航涛, 曾少锋, 张璐, 等. 基于共轴双旋翼无人机的目标检测与薄弱位置定位系统设计[J]. 中国科学院大学学报, DOI: 10.7523/j.ucas.2022.017.

**摘要** 由自主控制算法控制的非载人无人飞行器在执行定向打击等危险任务中往往比有人机具有更大的优势。然而, 在执行爆破任务的时候, 无人机在不同场景下的检测算法鲁棒性往往无法得到保证, 这极大地影响了无人机对目标的定位效果, 导致执行任务的成功率大幅降低。为了解决上述问题, 我们提出利用基于跨域的目标检测算法来提高无人机在不同场景下检测算法的鲁棒性, 并通过在线 GPS 聚类算法提高无人机对目标定位的稳定性。同时, 鉴于目标爆破位置对爆破结果的影响, 该系统通过提出一种薄弱部位定位算法, 提高爆破的精确度和成功率。

**关键词** 无人机; 目标检测; 知识蒸馏; 图像分割

中图分类号: TP319.4 文献标志码: A DOI: 10.7523/j.ucas.2022.017

## Design of object detection and weak position location system base on coaxial dual-rotor drone

FENG Hangtao<sup>1,2</sup>, ZENG Shaofeng<sup>1,2</sup>, ZHANG Lu<sup>1,2</sup>, YANG Xu<sup>1,2,3</sup>, LIU Zhiyong<sup>1,2,3</sup>

(1 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China; 2 The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China; 3 Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, 200031, China)

\* 科技创新 2030 -“新一代人工智能”重大项目 (2020AAA0108902)、中国科学院战略性先导科技专项 (XDB32050100)、东莞市核心技术攻关项目 (2019622101001) 和国家自然科学基金 (61627808) 资助

† 通信作者, E-mail: zhiyong.liu@ia.ac.cn

**Abstract** Unmanned aerial vehicles controlled by autonomous control algorithms often have greater advantages than manned aircraft in the execution of directional strikes and other dangerous tasks. When performing blasting tasks, the robustness of the UAV detection algorithm in different scenarios is often not guaranteed, which greatly affects the UAV's positioning of the target, thus reducing the success rate of the mission. In order to solve the above problems, the cross-domain-based object detection algorithm is used to improve the robustness of the UAV detection algorithm in different scenarios, and the online GPS clustering algorithm is used to improve the robustness of object positioning. At the same time, in view of the impact of the object blasting position on the blasting result, the system uses an algorithm for locating weak parts to improve the accuracy and success rate of blasting.

**Keywords** unmanned aerial vehicles; object detection; knowledge distillation; image segmentation

无人机是一类通过自主程序控制或者利用无线电遥控设备和控制装置操纵的非载人飞机。与载人飞机相比,它具有体积小、造价低、使用方便、对作战环境要求低、战场生存能力较强等特点,且具有自主飞行能力的无人机更适合到危险的环境中执行任务。在军事环境中,无人机越来越多地应用到不同的场景中,比如:情报侦查,定点打击,电子干扰等<sup>[1-5]</sup>。

然而在当前的研究和实际应用中,缺乏能够在不同战场环境中进行指定目标智能打击的无人机系统,因而这种任务往往由无人机指挥员或者载人飞机执行。这种任务需要飞行员时刻保持高度警觉,同时拥有对不同战场环境的适应能力和快速的反应能力,这对飞行员来说是个巨大的挑战。因此,研究能够在不同战场环境中进行指定目标自主打击的无人机系统是非常必要的。

目标检测是无人机执行自主巡航打击任务过程中最基础的问题之一,其目的是对图像中的物体进行定位并识别出类别<sup>[6]</sup>。随着深度学习技术的发展,基于深度学习的目标检测算法在标准数据集上取得了明显的进步,但是在真实、开放的环境中,由于光线,背景,视角等因素时刻发生变化,使得训练数据(源域)和测试数据(目标域)分布产生较大差异,导致网络的泛化性能明显下降;同时,在目标域上收集、标注大量用于训练的数据也需消耗大量人力和物力,更遑论在多数军事对抗下,根本无法提前获取大量敌方目标样本。针

对上述问题,我们提出利用一种风格迁移<sup>[7]</sup>的网络,将数据从源域迁移到目标域,用于网络的学习,从而提高网络的泛化性能。

当无人机进行目标打击时,需要不断下降靠近目标,但是无人机本身的抖动或者外界干扰产生的抖动导致固定在无人机上的相机拍摄的画面会产生巨大的偏移,从而影响下降阶段定位的稳定性。为了解决这个问题,本文提出一种像素补偿的方法来进行消抖,提高无人系统下降阶段定位的稳定。

当下降到一定高度时,目标可能充斥整个相机视野,使得仅仅使用彩色图像信息定位的精度受到较大的影响。因此,本文提出了一种融合深度信息和彩色图像信息的薄弱部位精准定位算法,提高单个模态定位的精度,提高打击的准确性和高效性。同时通过深度信息的判断,保证了对投弹时机的精准把握。

综上,本文设计了一种基于无人机平台的自主巡航、检测、定位、打击的系统。通过结合跨域目标检测,相机消抖和薄弱部位定位等模块,所提出的算法有效提高了整个系统的可靠性和稳定性。该系统在实际搭建的环境中进行了充分的测试,能够圆满完成预定的打击任务。

## 1 系统设计

### 1.1 系统总体架构

本文设计了一个基于共轴双旋翼无人机平台的自

主巡航, 打击系统。整个系统由硬件系统和软件系统构成, 其中硬件系统如图 1, 主要划分为: (1) 动力总成模块; (2) 电源/BMS 模块; (3) 飞行控制模块; (4) 任务载荷模块; (5) 目标探测感知模块。其中主要的智能感知系统搭载目标探测感知模块上。智能感知系统主要包括: 目标检测模块, 下降靠近模块, 薄弱部位定位模块三部分。系统执行任务的总体流程图如图 2 所示: 首先, 无人机需要到达指定位置, 并在指定区域内巡航, 同时进行目标检测任务; 当巡航完成, 若发现目标, 则靠近目标并下降打击目标; 若未发现目标, 则飞抵下一个巡航区域开展相同的任务。

图 2 系统流程图

Fig.2 Systems flowchart

## 1.2 系统硬件设计

针对指定目标打击任务, 微小型共轴双旋翼无人飞行器结构紧凑、体积小、灵活性好、隐蔽性强、具备较长时间空中悬停、垂直起降以及多方向飞行的能力, 能够在复杂或者狭小环境中飞行, 在执行攻击任务时, 具有固定翼飞行器无法比拟的优势。因此, 本文使用共轴双旋翼无人机平台进行实验。

### 1.2.1 共轴双旋翼无人平台硬件设计方案

共轴双旋翼无人飞行平台硬件系统基于小型化、高可靠性、易维修性和易扩展性等原则进行优化设计。硬件系统根据系统结构设计方案进行总体布局, 模块间的信号用屏蔽线连接。各主要电路组件采用 SMD 封装, 自发热器件通过有利于电屏蔽和热源隔离, 并且提高系统的可维修性, 电气总体布局如图所示。根据系统要求以及系统结构, 硬件系统主要划分为: (1) 动力总成模块; (2) 电源/BMS 模块; (3) 飞行控制模块; (4) 任务载荷模块; (5) 目标探测感知模块, 如图 1。其中目标探测感知模块是无人系统智能核心, 其主要功能是在可见光和弱光情况下, 检测并识别既定目标, 完成对目标的定位, 引导飞行平台瞄准目标薄弱部位, 实现对薄弱部位的打击爆破。

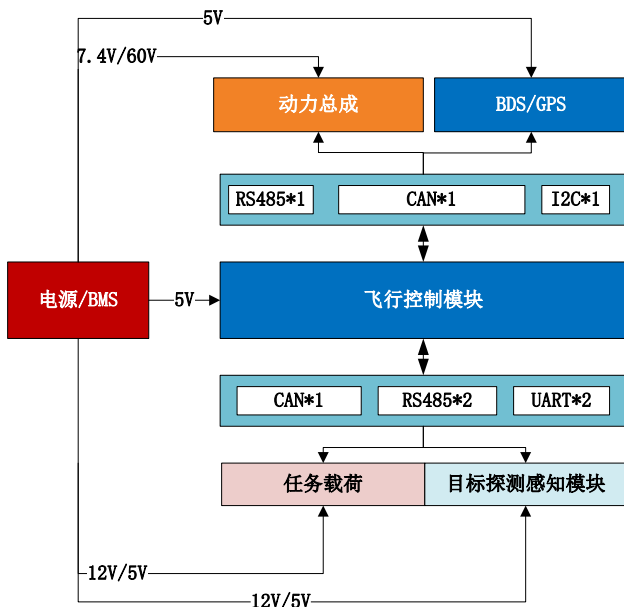
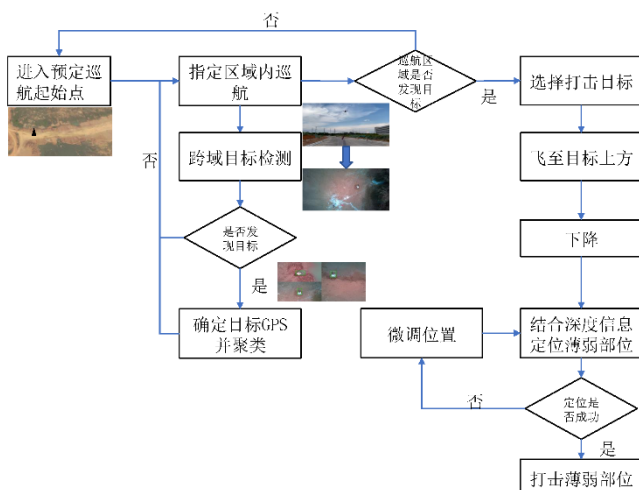


图 1 无人机平台系统框图

Fig.1 Block diagram of UAV platform system



## 1.3 系统软件设计

系统软件部分主要分为三个模块, 即: 目标检测模块, 下降靠近模块和薄弱部位定位模块 (附录部分提供了各个算法符号的含义)。

### 1.3.1 目标检测模块

现代战争包含的实际场景包括城市公路, 野外草地, 沼泽池塘等复杂多变的情况, 直接使用基于深度学习的算法, 往往在新环境中无法取得与训练环境中相当的结果, 比如, 在室内环境中训练的模型在室外环境中的检测效果就会显著下降。因此, 本文搭建了一种基

于风格迁移的跨域目标检测算法。整个算法包括离线风格迁移网络和目标检测网络两个部分。

该算法使用 Contrastive Learning for Unpaired Image-to-Image Translation (CUT)<sup>[8]</sup>网络模型完成对源域数据的风格迁移。其基于对比学习<sup>[9]</sup>的方法，目的是使编码器可以专注于源域和目标域之间共性的部分如形状，忽略两个域之间的差异部分如颜色，从而实现对源域数据的风格迁移。

在实际使用过程中，本文直接使用 CUT 网络在源域和目标域网络上进行训练。训练完成后，将目标域风格迁移到源域数据上，使用目标检测网络进行训练。

基于深度学习的通用目标检测算法主要分为二阶段算法和一阶段算法。其中，二阶段算法首先生成可能包含目标的候选区域，再对候选区域内的目标进行分类，主要包括 RCNN<sup>[10]</sup>，Fast-RCNN<sup>[11]</sup>，Faster-RCNN<sup>[12]</sup>等方法。二阶段算法比一阶段算法精度高，但是速度慢。一阶段算法的网络通过学习直接输出包含目标的区域坐标以及类别信息，主要包括 YOLO<sup>[13]</sup>，YOLO9000<sup>[14]</sup>，YOLOv3<sup>[15]</sup>，YOLOv4<sup>[16]</sup>，YOLOv5，SSD<sup>[17]</sup>，CenterNet<sup>[18]</sup>等方法。其速度快，但精度略低于二阶段算法。

在无人机上部署算法需要考虑硬件计算能力和功耗的限制，同时需要保证检测的速度和精确率的平衡，本文选用 YOLOv5 算法作为基础的目标检测算法，并使用蒸馏模型对网络进行压缩。

蒸馏，即知识蒸馏，Hinton 等<sup>[9]</sup>人在 2015 年首次提出神经网络中的知识蒸馏技术/概念。较前者的一些工作，这是一个通用而简单的、不同的模型压缩技术。蒸馏是将教师网络的知识迁移到学生网络上，使学生网络具备教师网络相当的性能，并且具有更小的模型和更快的速度，便于部署到算力较弱的设备上。无人机平台算力有限，而且为了保持更久地续航，硬件平台所拥有的功率也非常有限，这些都限制了大模型在无人机平台上的部署，因此，我们使用更小的模型 (YOLOv5s) 为基础，并结合蒸馏的方法，提高模型的性能和速度。

首先使用 YOLOv5l 作为教师网络在训练数据集上进行训练，并保留模型权重；然后使用 YOLOv5s 作为学生网络，进行蒸馏训练。

蒸馏网络结构如图 3 所示，首先使用精度更高 YOLOv5l 作为教师网络在数据集上进行训练，然后使用速度更快的 YOLOv5s 作为学生网络进行蒸馏训练，其中教师网络在进行蒸馏训练时，权重固定，学生网络的权重由教师网络的权重进行初始化，并且使用随机梯度下降算法进行更新，网络损失函数为：

$$Loss_{obj}(\mathbf{o}_i^{gt}, \mathbf{o}_i, \mathbf{o}_i^t) = f_{obj}(\mathbf{o}_i^{gt}, \mathbf{o}_i) + \lambda f_{obj}(\mathbf{o}_i, \mathbf{o}_i^t), \quad (1)$$

$$Loss_{cls}(\mathbf{p}_i^{gt}, \mathbf{p}_i, \mathbf{p}_i^t, \mathbf{o}_i^t) = f_{cls}(\mathbf{p}_i^{gt}, \mathbf{p}_i) + \mathbf{o}_i^t \lambda f_{cls}(\mathbf{p}_i, \mathbf{p}_i^t), \quad (2)$$

$$Loss_{reg}(\mathbf{bb}_i^{gt}, \mathbf{bb}_i, \mathbf{bb}_i^t, \mathbf{o}_i^t) = f_{reg}(\mathbf{bb}_i^{gt}, \mathbf{bb}_i) + \mathbf{o}_i^t \lambda f_{reg}(\mathbf{bb}_i, \mathbf{bb}_i^t), \quad (3)$$

三个损失函数分别为 YOLO 的目标损失，分类损失以及回归损失。其中， $\mathbf{o}_i^{gt}, \mathbf{o}_i, \mathbf{o}_i^t$  分别表示第  $i$  个位置对应的目标真实标签，学生网络预测值以及教师网络预测值； $\mathbf{p}_i^{gt}, \mathbf{p}_i, \mathbf{p}_i^t$  分别表示第  $i$  个位置对应的类别标签，学生网络预测值以及教师网络预测值； $\mathbf{bb}_i^{gt}, \mathbf{bb}_i, \mathbf{bb}_i^t$  分别表示第  $i$  个位置对应的回归标签，学生网络预测值以及教师网络预测值。每一行等式右边第一项为网络的原始损失，即学生网络的输出与真实的标签的损失；第二项为蒸馏损失，其使用教师网络的输出作为标签，计算学生网络的输出与该标签的损失，其中  $\lambda$  为平衡常数，在实际训练时，我们设置为 0.05。网络的总体损失为三个损失的和。

$$Loss = Loss_{obj}(\mathbf{o}_i^{gt}, \mathbf{o}_i, \mathbf{o}_i^t) + Loss_{cls}(\mathbf{p}_i^{gt}, \mathbf{p}_i, \mathbf{p}_i^t, \mathbf{o}_i^t) + Loss_{reg}(\mathbf{bb}_i^{gt}, \mathbf{bb}_i, \mathbf{bb}_i^t, \mathbf{o}_i^t), \quad (4)$$

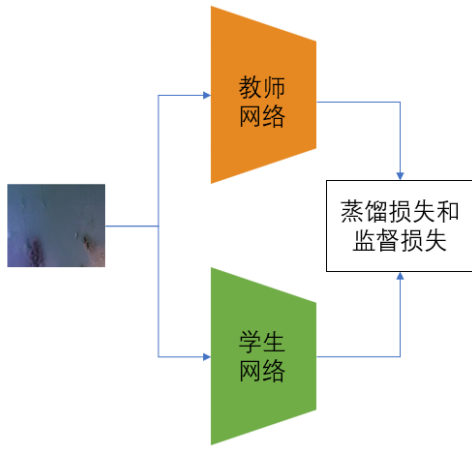


图3 知识蒸馏网络结构

Fig.3 Knowledge distillation network architecture

为了提高目标检测算法对于遮挡物体检测的效果，本文在数据标注时将每个目标都标注为  $N + 1$  个类，即局部 1，局部 2...局部  $N$  以及整体，前向推理时，通过投票的方法来确定是否检测到正确的目标，提高检测的稳定性（如图 4）。

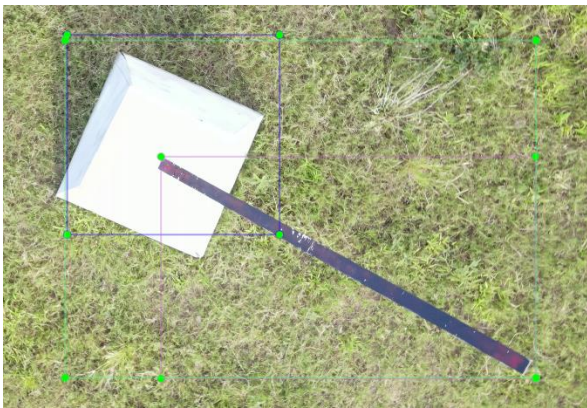


图4 标注样例

Fig.4 Annotated examples

当检出目标时，为了实现对目标的精准定位和打击，需要知道高精度的目标 GPS 坐标，但是单次计算目标的 GPS 会产生较大的误差，因此，使用一种 GPS 在线聚类的算法来提高定位的精度，其通过记录历史 GPS 坐标进行聚类，以减少单次计算误差。

### 1.3.2 下降迫近模块

在下降阶段，由于外界干扰或者无人机本身飞行

时产生的抖动会使相邻帧中的相同目标的像素坐标发生较大的漂移，影响定位的稳定性。因此，本文使用基于像素补偿的防抖算法和跟踪算法来提高定位的稳定性。

像素补偿防抖算法具体流程如下：首先根据无人机的姿态，即俯仰角，横滚角和方位角，计算出相机光轴与初始位置（即竖直向下）的夹角  $\theta$ ；其次将成像平面旋转  $\theta$ ，使得相机光轴与初始位置重合；最后对旋转后的图片进行裁剪和填补，然后再送入检测跟踪算法，进行目标识别和定位。

单次下降需要稳定地跟随同一个目标进行下降，此时需要跟踪器进行稳定的定位，但是仅仅使用跟踪器会出现错跟和漏跟的情况，将会严重影响到跟踪的效果。将跟踪器和检测器叠加使用，以减少漏检和误检情况，以提高系统的稳定性，具体算法流程见算法 2。

---

#### 算法 2 视觉防抖跟踪与鲁棒定位

---

输入：彩色图像  $I$ ，未检测到目标次数  $n$ ，未检测到目标次数阈值  $N$

1. **dets** = Detector( $I$ ), **track** = Tracker( $I$ ).
  2. **if** **dets** =  $\emptyset$  and **track**  $\neq \emptyset$  **then**
  3.      $n = n + 1$ .
  4.     **if**  $n \geq N$  **then**
  5.         Stop(Tracker).
  6.     **else**
  7.         **bb** = **track**.
  8.     **end if**
  9. **end if**
  10. **if** **dets** =  $\emptyset$  and **track** =  $\emptyset$  **then**
  11.      $n = n + 1$ .
  12.     **if**  $n \geq N$  **then**
  13.         Stop(Tracker).
  14.     **end if**
-

---

```

15. bb = None.
16. end if
17. if dets !=  $\emptyset$  and track !=  $\emptyset$  then
18.   if  $\max_{IoU}(dets, track) = 0$  then
19.      $n = n + 1$ .
20.   end if
21.   bb = track.
22. end if
23. if dets !=  $\emptyset$  and track =  $\emptyset$  then
24.   Init(Tracker).
25.   bb = Select(dets).
26. end if
输出: bb

```

---

### 1.3.3 薄弱定位模块

通过深度相机可以获取的深度图像包含了视野中物体的深度（距离）信息，而目标一般要高于其周围物体，因此利用这些深度信息同样有助于目标的检测以及薄弱部位定位任务。图 2 所示是两幅包含目标的深度图像数据（添加伪彩色），图像中的每个点的像素值代表此处的距离信息。图 5 中两幅图分别表示陆地和水面两种环境下获取的数据。图中，颜色越接近蓝色表示距离相机拍摄位置越近，而越接近红色表示距离越远。分析图像可知，目标的深度特征较明显，但同样存在较大的噪声，且当在水域环境下，噪声程度会极具增加。

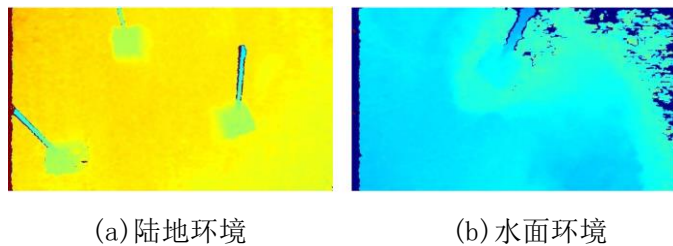


图 5 深度图像数据实例

Fig.5 Examples of depth images

基于上述分析，本文将基于深度图像的目标检测及薄弱位置定位问题的求解过程总结为如下几个步骤：1) 图像预处理；2) 目标提取；3) 薄弱位置定位。图像预处理主要用于消除噪声，一种简单有效的方法是通过深度相机测距的有效范围进行噪声去除，例如，深度相机测距有效范围为  $0.5m \sim 10m$ ，则将深度数据中大于等于  $10m$  小于  $0.5m$  的数据视为噪声去除，噪声处像素可重新赋值为非噪声像素的均值。

目标提取主要完成前景与背景的分割，本文采用图像处理中的二值化技术对深度图像进行分割提取前景，二值化的阈值通过二分查找获得。具体过程如算法 3。算法输入深度图像以及估算的目标在图像中的大小，其中可通过相机内参以及目标实际大小计算得到。算法 3 步骤 1~2 估算图像中的最大最小值，分别作为二分查找的上限和下限；步骤 4~18 为二分查找的具体过程，查找过程中保存分割后前景面积最接近的二值化结果，当前景面积和足够接近或当查找的上限下限足够接近时停止。

---

#### 算法 3 深度图像二值化

---

输入：预处理后的深度图像  $I$ ，目标在图像中的预估面积  $a$  (通过相机内参估算)

```

1.  $A = \text{AvgPool}(I, \text{kernel} = k, \text{step} = k)$ .
2.  $u = \min(A), v = \max(A)$ .
3.  $B = 0$ .
4. while true
5.    $t = (u + v)/2$ .
6.    $T = I > t$ .
7.   if  $|\text{sum}(T) - a| < \epsilon$  or  $v - u < \gamma$  then
8.     break.
9.   end if

```

---

---

```

10.  if abs(sum(T) - a) < abs(sum(B) -
      a) then
11.      B = T.
12.  end if
13.  if sum(T) > a then
14.      v = t.
15.  else
16.      u = t.
17.  end if
18. end while
输出: B

```

---

通过算法 3 获取二值化图像 **B** 后, 二值图像中前景很可能并非目标, 因此还需对目标做进一步判别。具体的, 首先获取二值图像中的连通域, 对各连通域分别进行判别, 判别方法包括但不限于面积分析、轮廓分析、形状分析以及神经网络等。面积分析即去除面积过小或过大的区域; 轮廓分析即目标的轮廓长度与其外接旋转矩形框的周长相近; 形状分析主要通过包含 2 个子分布的混合高斯模型估计前景分布, 并由此计算两个分布的长短轴比例, 长短轴两侧目标中像素坐标分布的对称性等; 神经网络方法主要用于对前景所在区域的原始深度图像提取特征并分类。

目标薄弱位置定位主要通过计算二值图像连通域质心获取。但受桩柱的影响, 质心会偏离薄弱部位而偏向桩柱一侧 (如图 6 实心圆所示)。为消除这种偏移, 可先求解目标连通域外界矩形框 (如图 6 虚线矩形框所示), 然后通过目标区域深度信息选择矩形框接近基座的短边, 并求解其中心位置, 薄弱位置即处于质心与该中心位置连线的中点处 (实心矩形)。

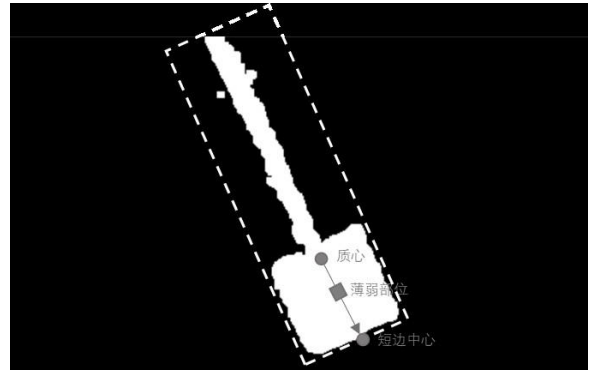


图 6 薄弱位置定位示意图

Fig.6 Schematic diagram of weak location positioning

## 2 实验与分析

### 2.1 数据集与评价指标

实验采用在两个不同的场景, 不同天气情况 (比如晴天, 阴天), 光照条件下采集到的数据进行跨域目标检测的验证。训练数据包括 739 张在郊区土地场景下尺寸为 1280\*720 的彩色图像, 共包括 3621 个实例, 分为“局部 1”, “局部 2”和“整体”三类。测试数据采用 485 张池塘场景下尺寸为 1280\*720 的彩色图像, 包括 1245 个样例 (部分样例如图 7 所示)。

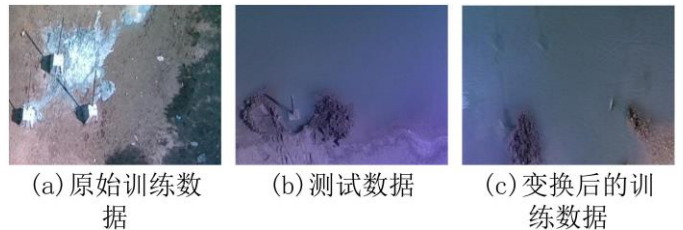


图 7 训练图像与测试图像

Fig.7 Training images and testing images

对于薄弱位置定位算法的测试, 共选取了 1250 张距离目标高度在 0.4m 至 5m 高度不等, 水边陆地不同场景的尺寸为 1280\*720 的深度数据。

除了在静态的数据集上的测试, 我们也在实际情况下进行了测试。实际测试是在 4 个不同的场景下进行的 (如图 8 所示), 同一个场景也包含不同的天气条

件,如晴天阴天,不同的光线条件,如强光弱光,以及场景中出现的不同的干扰,如制造假的目标,车辆干扰,行人干扰等等。

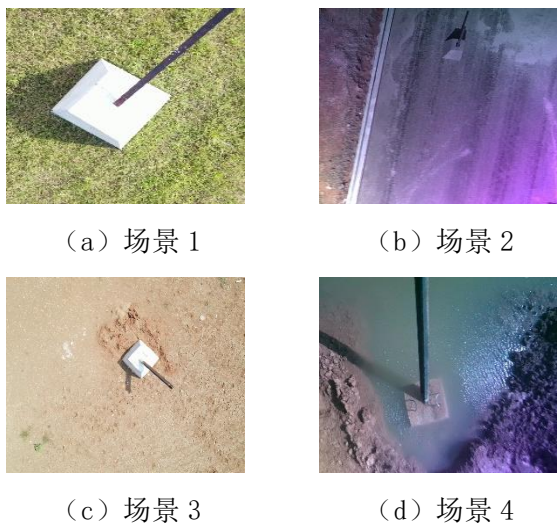


图 8 测试场景

Fig.8 Testing scenarios

## 2.2 实验设置

目标探测感知模块由于飞行平台重量和体积的限制,其由嵌入式 AI 计算平台 NVIDIA Jetson TX2 和深度相机 Intel RealSense D455 构成。其中 NVIDIA Jetson TX2 配备 NVIDIA Pascal GPU,具有 8GB 内存和 256 个计算核心,可以实现快速的深度学习计算和推理; Intel RealSense D455 相机具备 RGB 传感器和深度传感器,RGB 传感器可以为无人机提供 1280\*720 像素的彩色图片,为目标检测模型提供高清有效的输入数据;深度传感器在 0.4m 至 6m 范围内可以提供高精度的深度数据,对于实现对目标的薄弱位置高精度定位提供有效的支持。

目标检测与蒸馏算法的具体参数信息如表 1 所示。

表 1 算法参数表

Table 1 Algorithm parameters

参数	数值
学习率 lr	0.01

权值衰减 wd	0.0005
动量系数 momentum	0.937
每批数量 batch_size	64
最大迭代次数 epochs	300
平衡常数 $\lambda$	0.05

目标检测模块的训练中,使用 4 张显存大小是 11GB 的 NVIDIA Geforce 2080Ti 显卡训练,并在无人机平台上进行测试。在训练过程中,采用带有 Nesterov 动量的 SGD 训练网络,初始学习率初始化设置为 0.01,权值衰减设置为 0.0005,动量系数设置为 0.937, batch size 设置为 64,最大迭代次数设置为 300 轮,并用余弦退火训练的前 3 轮。网络的输入图像大小为 640\*640,并采用随机上下翻转,左右翻转,颜色增强,随机 mixup 和 mosaic 操作对数据进行增广。蒸馏训练时,我们设置平衡常数  $\lambda$  为 0.05。

视觉防抖跟踪与鲁棒定位算法中,将未检测到目标次数阈值 N 设置为 5。

深度图像二值化算法中,经过池化后图片大小为 640\*360,均匀池化中卷积核大小为 2,步长 2,其中  $\epsilon$  和  $\gamma$  分别设置为  $a$  的 10% 和 30。

## 2.3 评价方式

对于目标检测算法,采用的评价指标为查准率 P (Precision),以及在不同 IOU (Intersection over Union) 阈值下的平均类别精确率 mAP (mean Average Precision) 和召回率 R (Recall)。

对于薄弱位置定位算法的评估,使用传统的 Niblack<sup>[20]</sup>, Otsu<sup>[21]</sup>, Sauvola<sup>[22]</sup> 进行对比,采用像素的平均绝对误差 MAE (Mean Absolute Error) 和平均计算所需时间作为评价指标。

## 2.4 实验结果

目标检测的实验结果如表 2 所示,由图中可以看出,使用 CUT 风格迁移网络进行数据增广可以极大地



提高网络在测试数据集上的性能 (mAP@0.50从12%提高到41%), 从而使网络具备在不同环境下的检测能力; 通过知识蒸馏, 在保持在 NVIDIA Jetson TX2 上实现平均 42ms/帧的处理速度的前提下, 进一步提高模型的性能 (mAP@0.50从41%提高至43%)。并且在实际搭建的硬件平台上, 在实际的不同光照条件和随机增加一些外界干扰的场景中测试可以高效地模拟完整的爆破流程。

表 2 实验结果

Table 2 Experimental results

模型	查准率 (P)	召回率 (R)	mAP@0.50	mAP@0.50:
baseline	0.28	0.27	0.12	0.03
baseline+C UT	0.41	0.61	0.41	<b>0.14</b>
baseline+C UT+DK	<b>0.43</b>	<b>0.63</b>	<b>0.43</b>	<b>0.14</b>

将薄弱位置定位算法, 分别与 Niblack 算法, Otsu 算法和 Sauvola 算法进行对比, 实际对比结果如图 9 所示, 可以很明显地看出本文所提出的算法分割的效果优于其他三种算法。

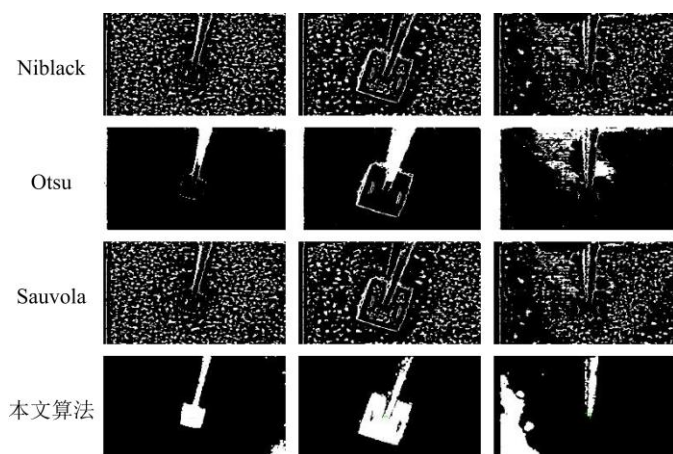


图 9 分割结果

Fig.9 Segmentation results

如表 3 所示, 在定量结果上, 本文提出的算法显然优于对比的三种方法, 其中 “--” 表示根据分割结果和计算薄弱位置的规则, 无法获得薄弱位置的像素坐标, 本文提出的算法在测试数据上取得平均绝对误差为 34.6 个像素的结果, 远远优于 Otsu 的 337.4 和 Sauvola 的 131.8; 在计算速度上, 所提出的算法在硬件平台上可以达到 10.3ms 每帧的速度, 也远远优于对比的三种算法。两个指标都证明了提出的分割算法的有效性。

表 3 实验结果

Table 3 Experimental results

算法	MAE	Speed (ms)
Niblack	--	17.9
Otsu	337.4	18.0
Sauvola	131.8	24.2
本文算法	<b>34.6</b>	<b>10.3</b>

同时, 我们在实际不同场景下的测试也较好的完成了整个任务流程, 最终正确下降至距离薄弱位置。

### 3 结论

本文提出了一种基于无人机平台的目标检测, 下降靠近及薄弱部位定位的系统架构。该系统使用风格迁移模型将源域数据迁移至目标域, 提高模型在目标域的性能, 并通过知识蒸馏在保持模型推理速度的前提下, 提高模型的检测性能; 下降过程中, 将检测结果和跟踪结果融合判断, 提高对目标定位的准确率和鲁棒性; 最后使用基于图像分割的薄弱部位定位算法, 进一步提高系统对目标薄弱位置的精准定位, 提高爆破的准确性和有效性。并通过在实际采集的数据上进行测试, 验证了本文提出系统的可行性和有效性。

### 参考文献

- [1] Zhao J H, Xiao G, Zhang X C, et al. A survey on object tracking in aerial surveillance[C]//Proceedings of

- International Conference on Aerospace System Science and Engineering 2018, 2019: 53-68. DOI:10.1007/978-981-13-6061-9\_4.
- [2] Vanegas F, Campbell D, Roy N, et al. UAV tracking and following a ground target under motion and localisation uncertainty[C]//2017 IEEE Aerospace Conference. March 4-11, 2017, Big Sky, MT, USA. IEEE, 2017: 1-10. DOI:10.1109/AERO.2017.7943775.
- [3] Tijtgat N, Van Ranst W, Volckaert B, et al. Embedded real-time object detection for a UAV warning system[C]//2017 IEEE International Conference on Computer Vision Workshops. October 22-29, 2017, Venice, Italy. IEEE, 2017: 2110-2118. DOI:10.1109/ICCVW.2017.247.
- [4] Du D W, Qi Y K, Yu H Y, et al. The unmanned aerial vehicle benchmark: Object detection and tracking[M]//Computer Vision – ECCV 2018. Cham: Springer International Publishing, 2018: 375-391. DOI:10.1007/978-3-030-01249-6\_23.
- [5] Mittal P, Singh R, Sharma A. Deep learning-based object detection in low-altitude UAV datasets: A survey[J]. Image and Vision Computing, 2020, 104: 104046. DOI:10.1016/j.imavis.2020.104046.
- [6] Zou Z X, Shi Z W, Guo Y H, et al. Object detection in 20 years: A survey[EB/OL]. arXiv:1905.05055. (2019-05-16) [2021-11-01]. <https://arxiv.org/abs/1905.05055v1>
- [7] Parmar N, Vaswani A, Uszkoreit J, et al. Image transformer[EB/OL]. arXiv:1802.05751. (2018-02-15) [2021-11-01]. <https://arxiv.org/abs/1802.05751>
- [8] Park T, Efros A A, Zhang R, et al. Contrastive learning for unpaired image-to-image translation[C]//Computer Vision – ECCV 2020, 2020: 319-345. DOI:10.1007/978-3-030-58545-7\_19.
- [9] Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[EB/OL]. arXiv:2002.05709. (2020-02-13) [2021-11-01]. <https://arxiv.org/abs/2002.05709>
- [10] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//2014 IEEE Conference on Computer Vision and Pattern Recognition. June 23-28, 2014, Columbus, OH, USA. IEEE, 2014: 580-587. DOI:10.1109/CVPR.2014.81.
- [11] Girshick R. Fast R-CNN[C]//2015 IEEE International Conference on Computer Vision. December 7-13, 2015, Santiago, Chile. IEEE, 2015: 1440-1448. DOI:10.1109/ICCV.2015.169.
- [12] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149. DOI:10.1109/TPAMI.2016.2577031.
- [13] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. June 27-30, 2016, Las Vegas, NV, USA. IEEE, 2016: 779-788. DOI:10.1109/CVPR.2016.91.
- [14] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition. July 21-26, 2017, Honolulu, HI, USA. IEEE, 2017: 6517-6525. DOI:10.1109/CVPR.2017.690.
- [15] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL]. arXiv:1804.02767. (2018-04-08) [2021-11-01]. <https://arxiv.org/abs/1804.02767v1>
- [16] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: optimal speed and accuracy of object detection[EB/OL]. arXiv:2004.10934. (2020-04-23) [2021-11-01]. <https://arxiv.org/abs/2004.10934v1>
- [17] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[C]//Computer Vision – ECCV 2016, 2016: 21-37. DOI:10.1007/978-3-319-46448-0\_2.
- [18] Duan K W, Bai S, Xie L X, et al. CenterNet: keypoint

triplets for object detection[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). October 27 - November 2, 2019, Seoul, Korea (South). IEEE, 2019: 6568-6577. DOI:10.1109/ICCV.2019.00667.

- [19] Hinton G E, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2015, abs/1503.02531
- [20] McKinley G A. An introduction to digital image processing[J]. Clinical Microbiology Newsletter, 1990, 12(12): 89-92. DOI:10.1016/0196-4399(90)90015-4.
- [21] Otsu N. A threshold selection method from gray-level histograms[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1979, 9(1): 62-66. DOI:10.1109/TSMC.1979.4310076.
- [22] Sauvola J, Pietikäinen M. Adaptive document image binarization[J]. Pattern Recognition, 2000, 33(2): 225-236. DOI:10.1016/S0031-3203(99)00055-2.

## 附录：符号列表

$o_i^{gt}$	第i个位置对应的目标真实标签
$o_i$	第i个位置对应的目标学生网络预测值
$o_i^t$	第i个位置对应的目标教师网络预测值
$p_i^{gt}$	第i个位置对应的类别标签
$p_i$	第i个位置对应的类别学生网络预测值
$p_i^t$	第i个位置对应的类别教师网络预测值
$bb_i^{gt}$	第i个位置对应的回归标签
$bb_i$	第i个位置对应的学生网络预测值
$bb_i^t$	第i个位置对应的教师网络预测值
$I$	输入图片
Detector	检测器
Tracker	跟踪器
$dets$	检测结果
$track$	跟踪结果
$\emptyset$	空集
Stop( $\cdot$ )	停止函数
Init( $\cdot$ )	初始化函数
Select( $\cdot$ )	筛选检测框函数
AvgPool( $\cdot$ )	均匀池化函数
$T$	算法内部用于表示二值化图像的变量
$B$	二值化图像