**无人机飞行障碍精准检测与实时跟踪研究**

张继*+1*，时宏伟*1*

(1.四川大学1，四川 成都 610064)

**摘要：高低空飞行的无人机，时常面临飞机、鸟类、行人、汽车、交通灯等障碍物的碰撞威胁。如何准确地检测出这些飞行障碍，并在规划飞行路径时躲避它们，已经成为无人机研究方向的一个重要课题。本文专注于无人机障碍物检测与跟踪环节，基于YOLO目标检测与KCF目标跟踪，对高低空常见的5种典型障碍物进行精准识别与实时跟踪。探索了高准确率YOLO目标检测与高速度KCF目标跟踪之间的最佳平衡点，从而同时保证了无人机障碍物检测的准确率与实时性**。

**关键词：飞行障碍、目标检测、目标跟踪**

**中图分类号：**TP3 **文献标识码：**A

**Research on UAV Obstacle Detection and Real-time Tracking**

Zhang Ji*+1*，Shi Hong-wei*1*

(1. Sichuan University, Sichuan Province Chengdu 610064)

**Abstract:** Flying UAV (Unmanned Aerial Vehicle) often threatens by plane, birds, pedestrian, car, traffic light and so on. How to detect those obstacles, and avoid them while planning flight route, has become an important research topic of UAV. This paper focuses on UAV obstacle detection and obstacle tracking. Based on YOLO object detection and KCF object tracking, we carried out precise obstacle detection and real-time tracking on 5-typical obstacles. We explored the optimal balance between high accuracy YOLO object detection and high speed KCF object tracking, thus ensuring the accuracy and real-time performance of UAV obstacle detection simultaneously.

**Keywords:** Flying Obstacle, Object Detection, Object Tracking

随着技术的发展，无人机在军民两大领域逐渐得到广泛运用。近几年的阅兵式上，军用无人机愈发崭露头角，翼龙、暗剑、彩虹等无人机在情报侦察、军事打击、信息对抗、通信中继和后勤保障等领域能发挥重大作用；民用无人机最具代表性的当属大疆出品的系列无人机，其在农业、基建、电力巡检、摄影和公共安全等领域能发挥重大作用。

高空飞行的无人机，时常会面临飞鸟以及其他飞机的威胁；低空飞行的无人机，与行人、汽车、交通灯、飞鸟相碰撞的几率较大。这不仅会对无人机自身造成损伤，更有可能给其他飞机、行人带来严重伤害。



图1 典型高空障碍



图2 典型低空障碍

# 主要贡献

本文以无人机高低空作业场景中常见的典型障碍物作为研究对象，使用YOLO目标检测模型进行典型障碍物的精准识别检测，同时借助KCF目标跟踪算法提升实时性。我们测试了不同检测间隔对准确度与速度的影响，以找到精准检测与快速跟踪之间的最优平衡点。

优点：

1. 使用YOLO目标检测模型进行典型障碍物的精准识别检测；
2. YOLO仅检测飞机、行人、汽车、交通灯、飞鸟5种类别障碍物，节省（80-5）/80=93.75%的后处理时间；
3. 每隔10帧使用一次YOLO目标检测，其余帧使用KCF目标跟踪，在保证80%准确度的同时，提升了5倍的运行速度；
4. 可扩展性：将自定义类别数据（如房屋、树木）加入YOLO训练数据集，重新训练YOLO目标检测模型，可使无人机获得其他类别障碍物的检测避让能力。

# YOLO目标检测模型

YOLO是一种基于深度神经网络的目标检测模型，以darknet为基本骨架，定义了YOLO自己的损失函数，然后在COCO、VOC或其他数据集上进行训练。

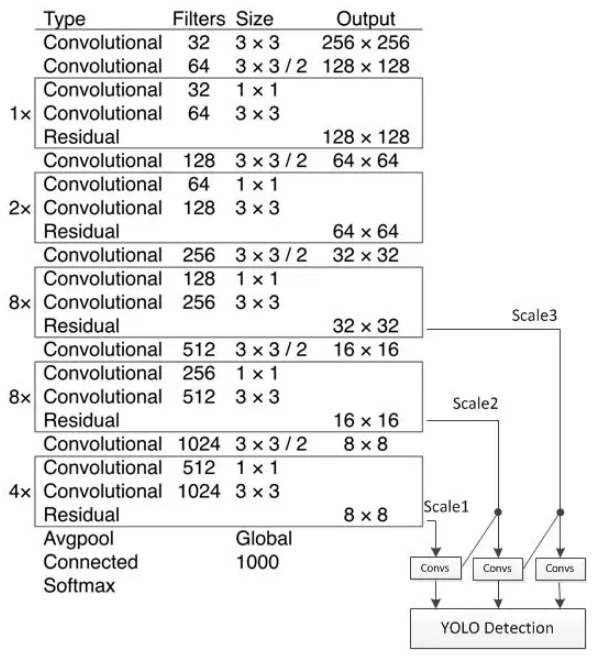


图3 YOLO神经网络架构

常用的YOLO模型如表1所示，它们有着不同的准确度mAP和检测速度FPS。大体而言，准确度越高，则检测速度相对越慢。综合考虑准确度和检测速度的影响，我们选择YOLOv3-416作为目标检测模型，它使用COCO作为训练数据集，具备飞机、行人、汽车、交通灯、鸟类等80个类别障碍物的检测识别能力。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 常用YOLO模型 | | | | | |
| 序号 | 模型名称 | 训练数据 | mAP | FLOPS | FPS |
| 1 | YOLOv3-320 | COCO | 51.5 | 38.97 Bn | 45 |
| 2 | YOLOv3-416 | COCO | 55.3 | 65.86 Bn | 35 |
| 3 | YOLOv3-608 | COCO | 57.9 | 140.69 Bn | 20 |
| 4 | YOLOv3-tiny | COCO | 33.1 | 5.56 Bn | 220 |
| 5 | YOLOv3-spp | COCO | 60.6 | 141.45 Bn | 20 |

# KCF目标跟踪模型

KCF即Kernel Correlation Filter（核相关滤波），它通过物体在当前帧的位置，来预测物体在下一帧的位置，是一种快速的目标跟踪算法，运行速度可达到150fps。KCF主要有循环矩阵和相关滤波这两个特点：

## 循环矩阵

KCF算法将图像沿着上下、左右的方向进行平移，以产生更多的样本用于训练。同时这种平移可以通过循环矩阵来表示，循环矩阵可以对角化，计算时仅需关注对角线上的非零元素，因此能够大幅加快矩阵与矩阵的计算速度。

## 相关滤波

相关是衡量两个信号相似值的度量，相关值越高，表示这两个信号越相似。KCF的输入往往是通过其他检测手段得到的物体在某一帧图像上的准确位置，该准确位置记为patch\_0。在后续的图像帧上，有patch\_1~patch\_n共计n个patch，那么KCF预测的位置即为与patch\_0相关值最高的patch。

Patch\_predict = max(correlation(patch\_0，patch\_k))，（1≤k≤n）

在计算patch\_0与patch\_k（1≤k≤n）的相关值时，转换到了傅里叶域进行。因为两个patch的相关卷积相当于傅里叶域中的元素乘积（时域卷积 = 频域点积），而乘积计算大大快于卷积运算。

# 最佳检测间隔

我们以YOLO目标检测的bbox作为KCF跟踪算法的初始输入，并在随后所有视频帧使用KCF替代YOLO检测出物体在每帧图像中的位置。初始输入时无人机在汽车的正后方，YOLO准确地找到了汽车的位置（见图3，红色矩形框完整包围了汽车）。随着时间的推移，无人机飞到了汽车的右侧面，视角相对初始位置发生较大变化，我们发现KCF逐渐无法跟踪到正确的目标（见图3，红色矩形框仅能包围汽车的小部分车尾）。



为了矫正这种跟踪错误，每隔一定的帧数，使用一次YOLO目标检测重新获得准确的目标位置，并更新KCF跟踪算法的输入。本小节以所有帧均进行YOLO检测得到的bbox和运行时间作为对照基准，研究不同YOLO检测间隔时的准确率和运行速度情况。

## 对照基准benchmark

使用无人机录制的一段包含汽车的视频作为数据研究对象，该视频时长54s，帧率30fps，共计54\*30=1620帧。对视频的每帧均进行YOLO目标检测，然后输出bbox，并输出该段视频总共的检测时间，作为优化对照的基准。部分结果如表1所示：

表1 每帧均进行YOLO目标检测

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| framNo | x1\_b | y1\_b | width\_b | height\_b |
| 7 | 821 | 630 | 235 | 197 |
| 8 | 821 | 633 | 235 | 194 |
| ··· | ··· | ··· | ··· | ··· |
| 1608 | 913 | 682 | 87 | 70 |
| elapseTime\_b | 76.831s |  |  |  |

其中，(x1\_b, y1\_b)表示benchmark bbox的左上角点，width\_b和height\_b分别表示benchmark bbox的宽和高。易知，benchmark bbox右上角点坐标为(x1\_b + width\_b, y1\_b)，benchmark bbox右下角点坐标为(x1\_b + width\_b, y1\_b + height\_b)，benchmark bbox左下角点坐标为(x1\_b, y1\_b + height\_b)。

由于实验条件限制，使用实验室的测试机对无人机拍摄的视频进行处理，测试机配置如表2所示。

表 2 测试机配置

|  |  |
| --- | --- |
| 系统 | Windows 10 企业版 64位操作系统 |
| CPU | CPU: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz 32GB |
| GPU | GPU: NVIDIA GeForce GTX 1060 6GB |

## 优化对照

对汽车视频每间隔interval帧抽取一帧进行YOLO目标检测，其余帧使用KCF输出bbox位置坐标和运行时间。interval为1时的检测跟踪结果如表3所示：

表3 interval为1时的检测跟踪结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| interval = 1 | | | | |
| framNo | x1\_r | y1\_r | width\_r | height\_r |
| 7 | 821 | 630 | 235 | 197 |
| 8 | 821 | 630 | 235 | 194 |
| ··· | ··· | ··· | ··· | ··· |
| 1608 | 913 | 682 | 87 | 72 |
| elapseTime | 46.145s |  |  |  |

其中，(x1\_r, y1\_r)表示real bbox的左上角点，width\_r和height\_r分别表示real bbox的宽和高。易知，real bbox右上角点坐标为(x1\_r + width\_r, y1\_r)，real bbox右下角点坐标为(x1\_r + width\_r, y1\_r + height\_r)，real bbox左下角点坐标为(x1\_r, y1\_r + height\_r)。

### 位置偏差

位置偏差采用交并比IOU（Intersection over Union）进行评价，顾名思义，IOU为两个集合的交集与并集之比，即：。

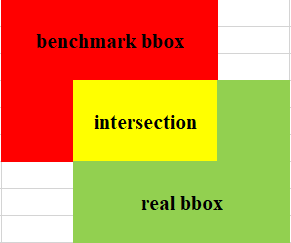


图4 单帧图像的IOU计算

IOU在相同图像帧之间进行计算，使用基准方法得到benchmark bbox，使用优化方法得到real bbox，则该帧图像的IOU=area\_intersection / (area\_benchmark\_bbox + area\_real\_bbox – area\_intersection)。对所有1620帧均计算IOU，最后得到该段视频的平均IOU。

### 速度提升

位置偏差采用交并比IOU（Intersection over Union）进行评价，顾名思义，IOU为两个集合的交集与并集之

*运行时间降低率计算公式为：*

## 优化结果分析

将interval值依次设置为2~100，并计算位置偏差和运行时间降低率，使用matlab分别绘制如下：

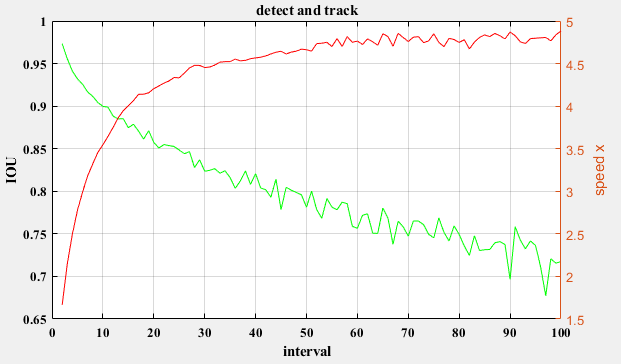


图4 不同Interval的位置偏差

图5 不同Interval的运行时间降低率

可以看到，当interval设置为5时，有最小位置偏差xxxx，运行时间降低率为xxxx。

# 结束语

本文探索使用自学习卷积神经网络分类器对LCD屏的生产缺陷进行检测，采用自学习的方法快速建立标注的缺陷库，通过大量训练图片，不断迭代更新CNN分类器，最终实现对LCD屏产品缺陷的实时检测分类。自学习CNN分类器对LCD屏缺陷的分类精度较高，且使用GPU完全能够满足LCD屏生产线的实时检测要求。本文的亮点在于自学习方法，它解决了高达数十万张训练图片标注缓慢的问题。通过速度更快的“数据筛选”替代“数据标注”，能够在较短时间内迅速扩充缺陷库的规模，而大量的训练数据又能够带来更高的分类精度，从而形成了一个良性循环的缺陷检测机制，研究结果具有较强的理论探索意义和工程应用价值。

# 参考文献

[1]S. S. Farfade, M. J. Saberian, and L. J. Li, “Multi-view face detection using deep convolutional neural networks,” in ACM Int. Conf. Multimedia Retrieval, 2015, pp. 643–650.

[2]J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017. 1, 2, 3

[3]陈崔军.TFT-LCD ARRAY 光刻制程 CD 均一性分析和实现[D].电子科技大学,2018.

[4]袁林. TFT混切技术的干蚀刻制程工艺优化研究[D].华南理工大学,2018.

[5]A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[6]K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[7]C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. In CVPR, 2015.

[9]谢希仁.计算机网络.电子工业出版社.2008.