

经费编号

B-9145-18-001181

上海师范大学

Shanghai Normal University

大学生创新创业训练计划 项目结题报告

项目名称 基于深度学习的鸟类识别系统监测系统

负责人姓名 刘璐

项目组成员 胡威、马梓珊、谢鸣、于孙甫

指导教师姓名 汪春梅

项目管理学院(部门) 信息与机电工程学院

上海师范大学教务处

二〇一八年十月制

说 明

报告内容：

每一申请结题项目，须提交项目结题报告。报告内容主要包括项目研究内容、实施过程中采取的方法和所做工作、项目成果，还应包括课题组成员在能力培养和素质提高，特别是在创新思维和创新实践方面的体验和收获。

结题报告可由成员提交的研究报告综合、整理而成，但各部分内容须有机结合，思路清晰，条理清楚。全文字数不限，语言应凝炼，引用或介绍已有成果的篇幅不宜过长。

凡有论文发表、申请专利、竞赛获奖等成果的项目，提交报告时请将所发表的论文电子版、录用通知(电子版或复印件)、专利申请书、获奖证书等证明材料一并提交。

报告要素：封面、摘要、关键词、正文、参考文献。

页面设置：WORD 编辑，A4 大小，页边距使用默认值。全文行距均为固定值 20 磅，段前、段后间距 0 行。报告中的图表要求有标题且清晰、单倍行距。各部分(除封面)排版格式见表 1。

表 1 各部分排版格式

文章要素	字号	字体	对齐方式等说明
“摘要：”	五号	宋体、加粗	两端对齐，左顶格
摘要内容	五号	宋体	两端对齐，无悬挂缩进
“关键词：”	五号	宋体、加粗	两端对齐，左顶格
关键词内容	五号	宋体	两端对齐，无悬挂缩进
正文	五号	宋体	两端对齐，段落首行缩进两字符
一级标题	四号	宋体、加粗	两端对齐，左顶格
二级标题	小四号	宋体、加粗	两端对齐，左顶格或适当缩进
三级及以下标题	五号	宋体	两端对齐，左顶格或适当缩进
插图及其标题	五号	宋体	居中，标题在图片下方
表格及其标题	五号	宋体	两端对齐，左顶格，标题在表格上方
“参考文献”	五号	宋体、加粗	两端对齐，左顶格
参考文献内容	五号	宋体	另起行，两端对齐，左顶格

项目负责人签名：

项目其他成员签名：

日期：_____年____月____日

基于深度学习的鸟类识别监测系统

刘璐，胡威，马梓珊，谢鸣，于孙甫

指导教师：汪春梅

(上海师范大学，信息与机电工程学院)

摘要：提出了一种基于深度学习的鸟类识别监测系统的设计。采用载有 OV4689 芯片的高速 USB 摄像头作为鸟类原始数据的采集器，结合 KCF 目标鉴别式跟踪算法，基于主流的深度学习工具 Tensorflow 的 object detection API，以及采用 SQL 语言建立鸟类数据库，实现对特定区域内鸟类物种的识别以及数量的统计。测试结果显示，该鸟类识别监测系统在测试环境下可以较准确对目标区域内鸟类进行物种识别和分类统计。

关键词：KCF、Tensorflow、物种识别、数据库

1. 引言

随着图像识别技术与物联网技术的迅速发展,对于动植物进行联网物种识别成为一个备受关注的研究热点。多年来，红外触发相机一直被广泛的应用于我国生物多样性的检测和野生动物的研究。然而因为硬件设备的固定以及鸟类活动的区域广泛等原因，地面和林下层鸟类在红外相机技术识别中占有绝对优势。所以说鸟类物种的检测仍需要更完善的监测系统，因此提出了一种基于深度学习的鸟类识别监测系统。

2. 研究方法过程

在研究过程中，首先我们采用了资料检索和实践训练的方式，充分利用网络资源，学习项目所需的各种算法和语言。在研究生的帮助下，我们在此期间不断地学习有关深度学习的基本算法知识和技能。在学习的过程中，我们把了解到的前沿技术和项目初期的设计方案不断融合改进，最后确定了比较完善的系统设计方案。项目中期，我们开始着手训练识别系统，除此之外我们还了解到了一些国外主流的深度学习方法，图像识别算法，目标跟踪算法等，从国内一些算法学习社区的经验分享，学习心得等，加深了我们对所设计的系统的理解。项目后期，我们在大量的测试中，不断地提高物种识别率，数据分析，以及误差改进，使系统功能更加完善。

3. 研究结论与成果

该项目所设计的基于深度学习的鸟类识别监测系统由摄像头数据采集系统，KCF 鉴别式目标追踪算法，图像识别算法以及 SQL 数据库四部分组成。摄像头采用载有 OV4689 芯片的 200 万像素高性能摄像头传感器，拍摄特定区域内活动中的鸟类，该部分主要完成目标区

域的鸟类原始数据采集。原始数据(即视频)先分离成一张张图片,之后图片经 object detection 图像识别进行标签,识别,除此之外还需要经 KCF 鉴别式目标跟踪算法处理,确认原始数据中 鸟类的数量,随后识别出的鸟类物种名称和数量会被统计在数据库中,为该区域的鸟类物种多样性分析作数据基础。本系统的总体设计框图如下图 1 所示:

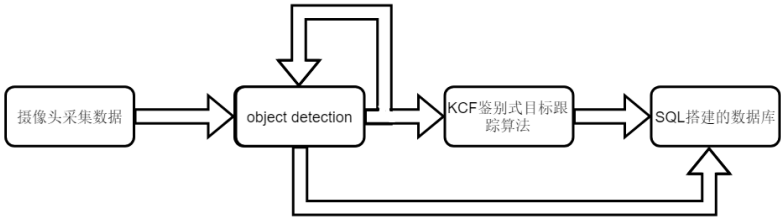


图 1 基于深度学习的鸟类识别监测系统

3.1硬件系统设计部分

图像采集模块:OV4689 芯片采用 OmniBSITM-2 技术的彩色 CMOS 400 万像素 (2688x1520) 图像传感器,它能够提供最同类的低光灵敏度和高动态范围 (HDR) 。OV4689 可以以每秒 90 帧 (fps) 的速度捕获全分辨率的 400 万像素高清 (HD) 视频,以 120 fps 的速度捕获 1080p 高清,以及以 180 fps 的速度捕获 720p 高清。传感器的高帧速率可实现快速移动物体的清晰,干净的图像和视频捕捉。图 2 为图像采集模块的功能图表。

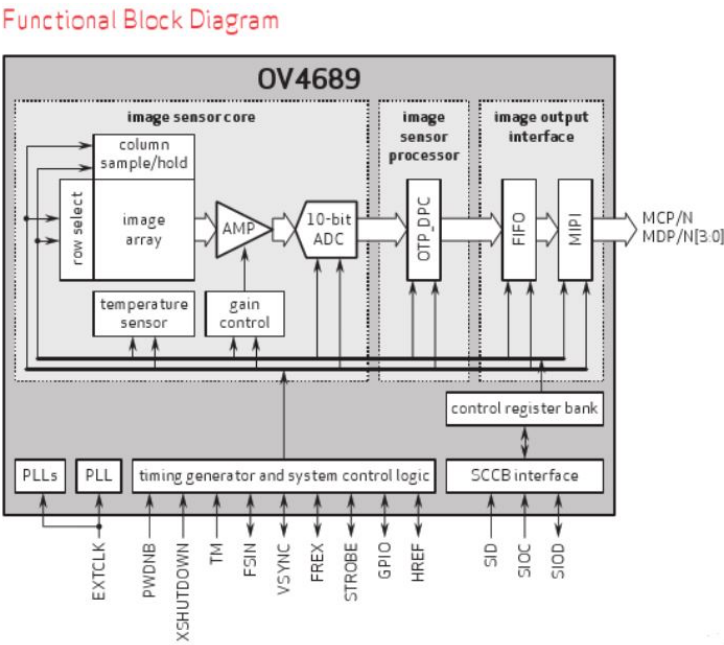


图 2 图像采集模块的功能图表

该摄像头模块采用 USB 直接与笔记本电脑通信,极大地方便了数据的传输和处理。

3.2图像识别模块的设计

Tensorflow 是一个基于数据流编程的符号数学系统,被广泛应用于各类机器学习算法的

编程实现。本项目使用 Tensorflow object detection API 与 openCV 实现动态目标的实时检测。Object detection 将目标图中的目标使用框体标注，并识别出框体中对应物体的类别。Tensorflow 官方已经给出一些在不同数据集上训练好的模型，在经过对训练速度和误差比较后，我们选用了 ssd_mobilenet_v1_coco 这个模型来训练自己的数据集。对于 Tensorflow，需要训练的图像必须事先转化为 TFrecord Format 格式。收集图片以 5: 1 的比例分为训练集和测试集，采用 LabelImg 这个软件手动对图片进行标注标签，将所有图片转化为 xml 格式。再通过编写 Python 的脚本文件，将 xml 文件转化为 csv 格式并生成 TFrecord 文件，此时数据已经能被 Tensorflow 所识别。

制作完需要训练的数据后，需要设置配置文件，将 ssd_mobilenet_v1_coco 模型内的学习率、识别类别数、识别种类、batch size 等参数根据实际情况进行更改。制作 Label map 也是训练前所必须要的一环，在 .pbtxt 文件中写入所有的标签类别，标签 id 顺序与生成 TFrecord 脚本文件中的顺序相同。至此，训练的预准备已经完成。

在对应 Tensorflow 的虚拟环境终端中，输入训练指令，训练就会依赖 GPU 或 CPU（本项目采用 GPU 训练模型）在预先设定的步数内不断进行迭代，从终端可以直观看到，随着训练步数的增加，模型的 loss 不断下降。完成设定步数后，可生成 frozen_inference_graph.pb 文件，即在自己数据集上训练所得到的模型。至此模型的训练部分已经完成，只需进行最后的测试。

该模块流程框图如图 3 所示：

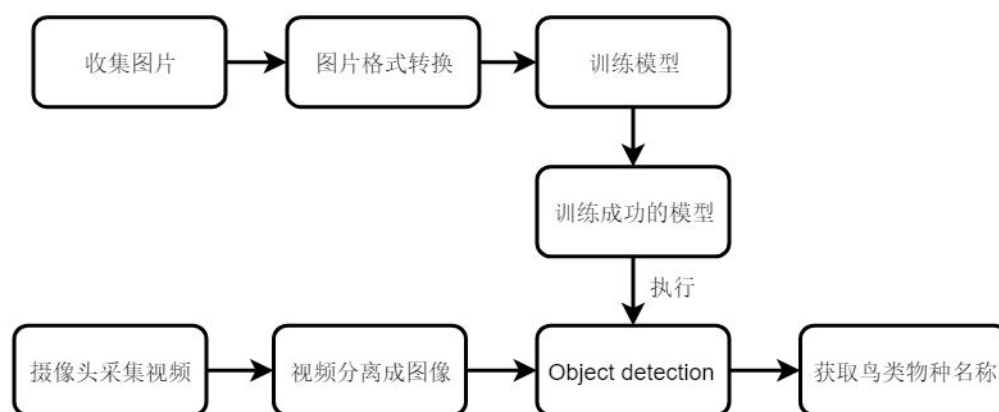


图 3 图像识别模块的功能图表

通过 Jupyter Notebook 打开 Object detection 文件夹下 object_detection_tutorial 的 ipynb 文件，将识别模型改为如上自己的模型，对代码进行修改并将所需要测试的文件放入对应文件夹。运行代码后可以看到图片中所需要识别的物体被框出且识别，并标注了识别该物体可信用度。

3.3 基于 KCF 核相关滤波算法实现目标跟踪

该部分的软件设计主要基于 opencv 库中的 8 种目标跟踪工具, 包括 BOOSTING Tracker, MIL Tracker, KCF Tracker, CSRT Tracker, MedianFlow Tracker, TLD Tracker, MOSSE Tracker 和 GOTURN Tracker。在经过准确度和速度的比较后, 我们最终选用 KCF 算法来实现对鸟类的跟踪。

下图为部分代码的实现, 图 4 设置不同类型的目标跟踪工具, 图 5 利用 KCF 算法更新物体框坐标。

```
7 # Set up tracker.
8 # Instead of MIL, you can also use
9
10 tracker_types = ['BOOSTING', 'MIL', 'KCF', 'TLD', 'MEDIANFLOW', 'GOTURN']
11 tracker_type = tracker_types[2]
12
13
14 if tracker_type == 'BOOSTING':
15     tracker = cv2.TrackerBoosting_create()
16 if tracker_type == 'MIL':
17     tracker = cv2.TrackerMIL_create()
18 if tracker_type == 'KCF':
19     tracker = cv2.TrackerKCF_create()
20 if tracker_type == 'TLD':
21     tracker = cv2.TrackerTLD_create()
22 if tracker_type == 'MEDIANFLOW':
23     tracker = cv2.TrackerMedianFlow_create()
24 if tracker_type == 'GOTURN':
25     tracker = cv2.TrackerGOTURN_create()
26
```

图 4 设置不同类型的目标跟踪工具

```
6 while True:
7     # Read a new frame
8     ok, frame = video.read()
9     if not ok:
10         #sys.exit()
11         break
12
13     # Start timer
14     timer = cv2.getTickCount()
15
16     # Update tracker
17     ok, bbox = tracker.update(frame)
18
19     # Calculate Frames per second (FPS)
20     fps = cv2.getTickFrequency() / (cv2.getTickCount() - timer);
21
22     # Draw bounding box
23     if ok:
24         # Tracking success
25         p1 = (int(bbox[0]), int(bbox[1]))
26         p2 = (int(bbox[0] + bbox[2]), int(bbox[1] + bbox[3]))
27         cv2.rectangle(frame, p1, p2, (255,0,0), 2, 1)
28     else:
29         # Tracking failure
30         cv2.putText(frame, "Tracking failure detected", (100,80), cv2.FONT_HERSHEY_SIMPLEX, 0.75,(0,0,255),2)
31
32     # Display tracker type on frame
33     cv2.putText(frame, tracker_type + " Tracker", (100,20), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50,170,50),2);
34
35     # Display FPS on frame
36     cv2.putText(frame, "FPS : " + str(int(fps)), (100,50), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50,170,50), 2);
37
38     print(bbox)
39
40     # Display result
41     cv2.imshow("Tracking", frame)
42     # Exit if ESC pressed
43     k = cv2.waitKey(1) & 0xff
44     if k == 27 :
45         break
46
```

图 5 利用 KCF 算法更新物体框坐标

KCF 全称为 Kernel Correlation Filter 核相关滤波算法。在 2014 年由 Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista 提出, 主要工作是以跟踪对象为正样本, 以周围环境为负样本, 训练一个判别分类器。

KCF 算法模型的建立主要有以下几个步骤：

1.线性回归：这里采用岭回归方法，或者叫做正则化最小二乘法，训练的目标是找到一个函数 $f(z) = W^T Z$ ，使函数值与其回归目标 y^i 的平方误差最小，其中 W^T 是 W 的转置矩阵，

$$\min \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2$$

求得函数中的 w 权值为，

$$W = (X^H X + \lambda I)^{-1} X^H y$$

2.循环矩阵：这个步骤是将上述式子中的样本矩阵 x 变为循环矩阵，假设 x 为一维矩阵，那么循环矩阵 X 为，

$$X = C(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}.$$

循环矩阵拥有的一个性质 $X = F \text{diag}(\bar{x}) F^H$ ，其中， \bar{x} 代表 x 的傅里叶变换， F 是离散傅里叶变换矩阵。因此得到

$$\begin{aligned} w^* &= (F^H \text{diag}(\bar{x}^*) F F^H \text{diag}(\bar{x}) F + \lambda I - 1 F^H \text{diag}(\bar{x}^*) F_y = \\ &F - 1 (\text{diag}(\bar{x}^* \otimes \bar{x}) + \lambda I) - 1 \text{diag}(\bar{x}^*) F_y = \\ &F - 1 \text{diag}(\bar{x}^* \bar{x} \otimes \bar{x}^\lambda) F_y \end{aligned}$$

其中， \otimes 代表向量对应元素相乘，然后两边同时左乘 F 得

$\bar{W}^* = \bar{x}^* \otimes \bar{y} \bar{x}^* \otimes \bar{x}^* + \lambda$ 至此，我们可以看出通过上述变换后，权重向量 w 的求解变换到了傅里叶变换域，而且计算量大大降低。

若将 x 扩展为二维图像，则循环矩阵如下图 6 所示，

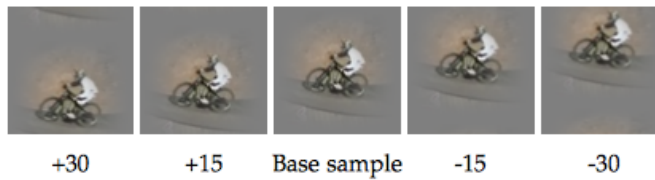


图 6 循环矩阵

3.4SQL 创建表

SQL 语言是一种数据库查询和程序设计语言,用于存取数据以及查询、更新和管理关系数据库系统。结构化查询语言 SQL (STRUCTURED QUERY LANGUAGE) 是最重要的关系数据库操作语言,并且它的影响已经超出数据库领域,得到其他领域的重视和采用,如人工智能领域的数据库检索。该部分采用 SQL 语言设计数据库,实现上海师范大学徐汇校区一个月内鸟类的目科属种和数量统计。以下是数据库设计及搭建原理:

1.E-R 模型图的建立

在建立数据库之前, E-R 模型的建立有利于加深对系统的完整认知,特别是数据库中各表的具体联系,使得在各表间需要添加的关系更加明了清晰。

如图 7, 在 OrderInfo、FamilyInfo、GeneraInfo、SpeciesInfo 四个需要建立的表中,两两相互约束。以目 (OrderInfo) 和科 (FamilyInfo) 为例: 一个目中含有多个科,属于一对多的关系。

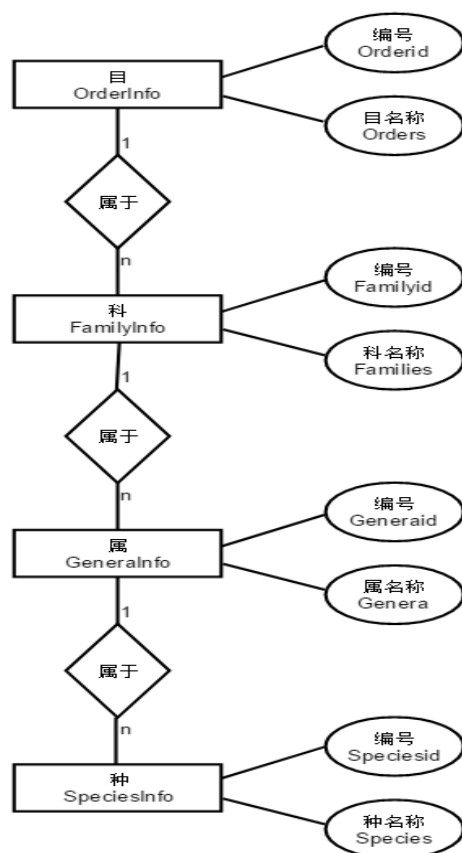


图 7 E-R 模型图

2. SQL Server 表的建立

列名	数据类型	允许 Null 值
Record_id	int	<input type="checkbox"/>
Orders	nvarchar(40)	<input type="checkbox"/>
Families	nvarchar(40)	<input type="checkbox"/>
Genera	nvarchar(40)	<input type="checkbox"/>
Species	nvarchar(40)	<input type="checkbox"/>
Record_Time	smalldatetime	<input type="checkbox"/>

图 8 建立 SQL Server 表

在创建好表后，根据 E-R 图添加外键并保存。

3. ADO.NET 操作数据库

Windows 窗体设计界面如下图 9：

分为数据增加模块，数据显示模块，数据修改模块和数据查询模块。

图 9 Windows 窗体设计界面

在数据增加模块和数据查询模块，鸟类的数据均是由目科属种的方式增加或查询的如下图 10，鸟类目科属种联动采用 TreeView 方法。

图 10 鸟类目科属种数据窗口

数据显示模块和数据修改如下图 11, 在选中 DataGridView 中的数据后, 在下方的“编辑修改”GroupBox 中也会显示对应的目科属种数据。



图 11 数据修改窗口

数据查询模块显示效果如下图 12:

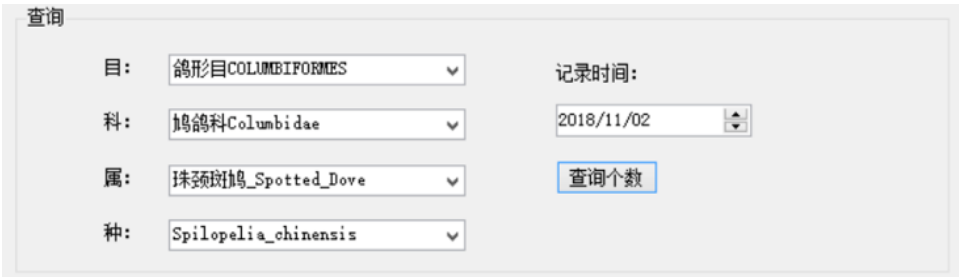


图 12 数据查询窗口

在确认记录时间后, 点击查询可以查询到对应日期出现的具体鸟类数据条数:

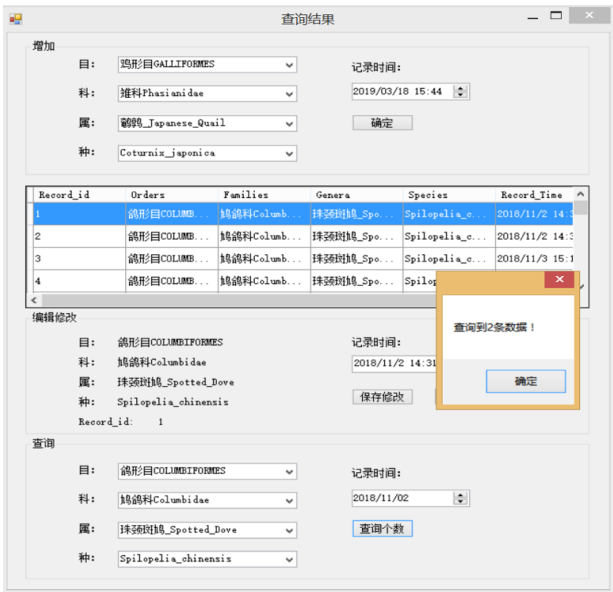


图 13 数据查询结果窗口

在通过鸟类识别后，将一个月内的上海师范大学徐汇校区鸟类数据通过 Windows 窗体界面插入数据，共有 41 条鸟类数据，在 SQL Server 中查询结果如下：

Record_id	Orders	Families	Genera	Species	Record_Time
1	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-02 14:31:00
2	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-02 14:31:00
3	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-03 15:11:00
4	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-06 16:01:00
5	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-11 09:11:00
6	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-15 13:34:00
7	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-16 13:45:00
8	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-20 11:18:00
9	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-23 16:21:00
10	鸽形目COLUMBIFORMES	鸠鸽科Columbidae	珠颈斑鸠_Spotted_Dove	Spilopelia_chinensis	2018-11-28 13:22:00
11	雀形目PASSERIFORMES	伯劳科Laniidae	棕背伯劳_Long-tailed_Shrike	Lanius_schach	2018-11-05 13:23:00
12	雀形目PASSERIFORMES	伯劳科Laniidae	棕背伯劳_Long-tailed_Shrike	Lanius_schach	2018-11-11 12:44:00
13	雀形目PASSERIFORMES	伯劳科Laniidae	棕背伯劳_Long-tailed_Shrike	Lanius_schach	2018-11-13 11:51:00
14	雀形目PASSERIFORMES	伯劳科Laniidae	棕背伯劳_Long-tailed_Shrike	Lanius_schach	2018-11-17 15:56:00
15	雀形目PASSERIFORMES	伯劳科Laniidae	棕背伯劳_Long-tailed_Shrike	Lanius_schach	2018-11-18 14:22:00
16	雀形目PASSERIFORMES	伯劳科Laniidae	棕背伯劳_Long-tailed_Shrike	Lanius_schach	2018-11-22 08:51:00
17	雀形目PASSERIFORMES	伯劳科Laniidae	棕背伯劳_Long-tailed_Shrike	Lanius_schach	2018-11-22 15:46:00
18	雀形目PASSERIFORMES	鸚科Eberizidae	小鸚_Little_Bunting	Eberiza_pusilla	2018-11-03 15:03:00
19	雀形目PASSERIFORMES	鸚科Eberizidae	小鸚_Little_Bunting	Eberiza_pusilla	2018-11-18 15:34:00
20	雀形目PASSERIFORMES	鸚科Eberizidae	小鸚_Little_Bunting	Eberiza_pusilla	2018-11-20 12:55:00
21	雀形目PASSERIFORMES	鸚科Eberizidae	小鸚_Little_Bunting	Eberiza_pusilla	2018-11-21 14:52:00
22	雀形目PASSERIFORMES	鸚科Eberizidae	小鸚_Little_Bunting	Eberiza_pusilla	2018-11-25 16:23:00
23	雀形目PASSERIFORMES	鸚科Eberizidae	黄喉鸚_Yellow-throated_B...	Eberiza_elegans	2018-11-06 13:06:00
24	雀形目PASSERIFORMES	鸚科Eberizidae	黄喉鸚_Yellow-throated_B...	Eberiza_elegans	2018-11-17 15:36:00
25	雀形目PASSERIFORMES	鸚科Eberizidae	黄喉鸚_Yellow-throated_B...	Eberiza_elegans	2018-11-18 14:13:00
26	雀形目PASSERIFORMES	鸚科Eberizidae	黄喉鸚_Yellow-throated_B...	Eberiza_elegans	2018-11-20 11:25:00
27	雀形目PASSERIFORMES	鸚科Turdidae	乌鸫_Common_Blackbird	Turdus_aerula	2018-11-09 16:22:00
28	雀形目PASSERIFORMES	鸚科Turdidae	乌鸫_Common_Blackbird	Turdus_aerula	2018-11-11 12:22:00
29	雀形目PASSERIFORMES	鸚科Turdidae	乌鸫_Common_Blackbird	Turdus_aerula	2018-11-19 13:56:00
30	雀形目PASSERIFORMES	鸚科Pycnonotidae	白头鸫_Light-vented_Bulbul	Pycnonotus_sinensis	2018-11-01 13:36:00
31	雀形目PASSERIFORMES	鸚科Pycnonotidae	白头鸫_Light-vented_Bulbul	Pycnonotus_sinensis	2018-11-19 16:06:00
32	雀形目PASSERIFORMES	鸚科Hirundinidae	家燕_Barn_Swallow	Hirundo_rustica	2018-11-12 13:06:00
33	雀形目PASSERIFORMES	鸚科Hirundinidae	家燕_Barn_Swallow	Hirundo_rustica	2018-11-12 14:26:00
34	雀形目PASSERIFORMES	鸚科Hirundinidae	家燕_Barn_Swallow	Hirundo_rustica	2018-11-12 14:56:00
35	雀形目PASSERIFORMES	鸚科Hirundinidae	家燕_Barn_Swallow	Hirundo_rustica	2018-11-24 13:45:00
36	雀形目PASSERIFORMES	鸚科Hirundinidae	家燕_Barn_Swallow	Hirundo_rustica	2018-11-30 12:30:00
37	鸚目PODICIPEDIFORMES	鸚科Podicipedidae	小鸛_Little_Grebe	Tachybaptus_rufic...	2018-11-13 15:22:00
38	鸚目PODICIPEDIFORMES	鸚科Podicipedidae	小鸛_Little_Grebe	Tachybaptus_rufic...	2018-11-13 18:06:00
39	雁形目ANSERIFORMES	鸚科Anatidae	绿翅鸚_Eurasian_Teal	Anas_crecca	2018-11-13 15:11:00
40	雁形目ANSERIFORMES	鸚科Anatidae	斑嘴鸚_Chinese_Spot-bill...	Anas_zonorhyncha	2018-11-02 09:10:00
41	雁形目ANSERIFORMES	鸚科Anatidae	斑嘴鸚_Chinese_Spot-bill...	Anas_zonorhyncha	2018-11-15 18:06:00

图 14 月统计数据显示窗口

3.5卷积神经网络 CNN 的图像分割

卷积神经网络（CNN）是一种深度学习模式，在图像分类识别，目标检测方面有着较好的表现。由于 CNN 在输入多维图像处理上的优势，可以将图像直接输入网络，较大程度的保留了原始样本的数据真实性，避免传统识别算法中的复杂的特征提取和数据重建过程。

CNN 作为一种多层的神经网络结构，其中每层由多个二维平面网络结构组成，包含了若干独立神经元。CNN 一般由卷积层，池化层以及全连接层构成。卷积层的每个节点连接的是上层输入的领域子图，用于表达图像的局部特征。池化层的处理是对卷积之后得到的特征进行聚类整合，包括平均池化和最大池化两种方式，其作用是统计各个区域相似特征的维数，通过降低特征维数，避免实验数据结果出现过拟合的情况，在局部感受域范围内，计算最大值或平均值，作为网络节点的输出值。

全连接层的作用是表示输出特征，通常由特征映射构成。特征输出向量的维数一般与最后一个特征映射输出的网络节点数量相同，利用输出的特征向量训练分类器模型实

现对目标类别的辨识。

边缘类聚方法的核心思想是依据 CNN 分割的初步结果，确定边缘像素的位置和颜色特征，然后边缘集合聚类为位置邻近和颜色相似的各个边缘子集，最后进行边界扩充，以获得完整连续的图像分割效果，提升分割结果的完整性和连续性。

CNN 模型结构如下图 15 所示

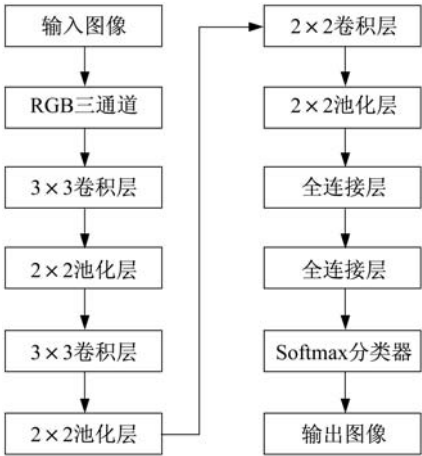
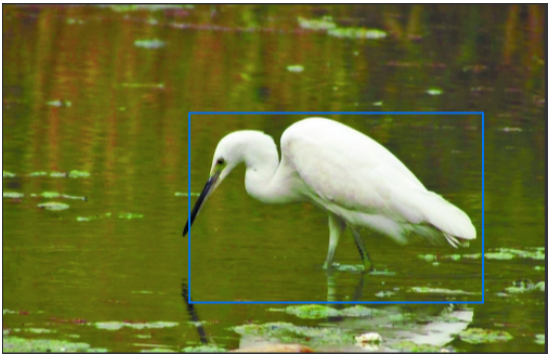


图 15 CNN 结构设计图表

3.6研究成果

通过系统调试我们进行了多种鸟类运动视频的拍摄，视频图像的数据处理，以及数据的统计。根据大量实践结果分析得出以下结论：在拍摄视频质量较高的情况下（即没有明显障碍物遮挡），可以取得不错的识别率。如果因为树枝等其他遮挡物大面积遮挡目标的情况下，识别效果不尽如人意。以下是徐汇校区常见两种鸟类识别结果：

对于白鹭的识别：



鸟类名称	白鹭	小白鹭	鹭鸟	白鹤	黄嘴白鹭
可能性	75.74%	18.39%	0.0454%	0.0051%	0.0021%

对于山麻雀的识别：



鸟类名称	山麻雀	旋木雀	黑顶麻雀	绿啄木鸟	夜莺
可能性	97.10%	0.0016%	0.0011%	0.0009%	0.0001%

目前该系统能够完成已搭建的情景框架下鸟类的识别和监测,但是在日后更为复杂的环境条件下,数据的准确性和稳定性还需要得到进一步的保证,所以系统还有很多改进的地方。

4. 成员分工及收获体会

刘璐	系统调试
马梓珊	目标跟踪
胡威	CNN 识别算法
谢鸣	模型训练
于孙甫	数据库搭建

在项目的实施过程中,每个人分工根据自己的兴趣爱好和专业特长,承担项目的各部分设计和实现,分工十分明确,但到后来发现,其实从刚开始的鸟类图像捕捉到训练模型以及数据库的搭建都是需要整个团体去合作实现。经过这几个月的项目设计和实现上的分析讨论,团队内队员之间也更有默契了。遇到难以理解的问题时,研究生也积极为我们解答,并且不断监督。在整个项目试验过程中,团队队员的学习能力以及科学综合实践能力明显提升,理论联系实际,从项目需求出发,在实验探究中,牵涉到专业中没有学习过的内容,队员们都纷纷选择自主学习,通过在网上查资料,购买相关学习资料,以及上网课等方式,深入学习项目所需知识技能。碰到困难,大家一起解决,整个研究过程中,团队队员们的合作以及思考能力有很大提高。总而言之,此次项目的完成对于每位组员而言都收获颇丰。

参考文献

- [1] 徐良,何毅.上海师范大学徐汇校区校园鸟类调查[J].绿色科技,2018(18):11-13.
- [2] 李红江.基于高斯核相关滤波的多特征融合的目标跟踪[J].电子技术与软件工程,2019(03):99-100.
- [3] 杨志方,陈曦.优化搜索策略的KCF目标跟踪算法[J].武汉工程大学学报,2019,41(01):98-102.
- [4] 陈翠平.基于支持向量机的目标检测算法综述 [J] . 控制与决策 2014 (02) 101-104 .
- [5]房雪键. 基于深度学习的图像分类算法研究[D].辽宁大学,2016