



南京农业大学

NANJING AGRICULTURAL UNIVERSITY

实验报告册

课程名称: 大数据与农业应用

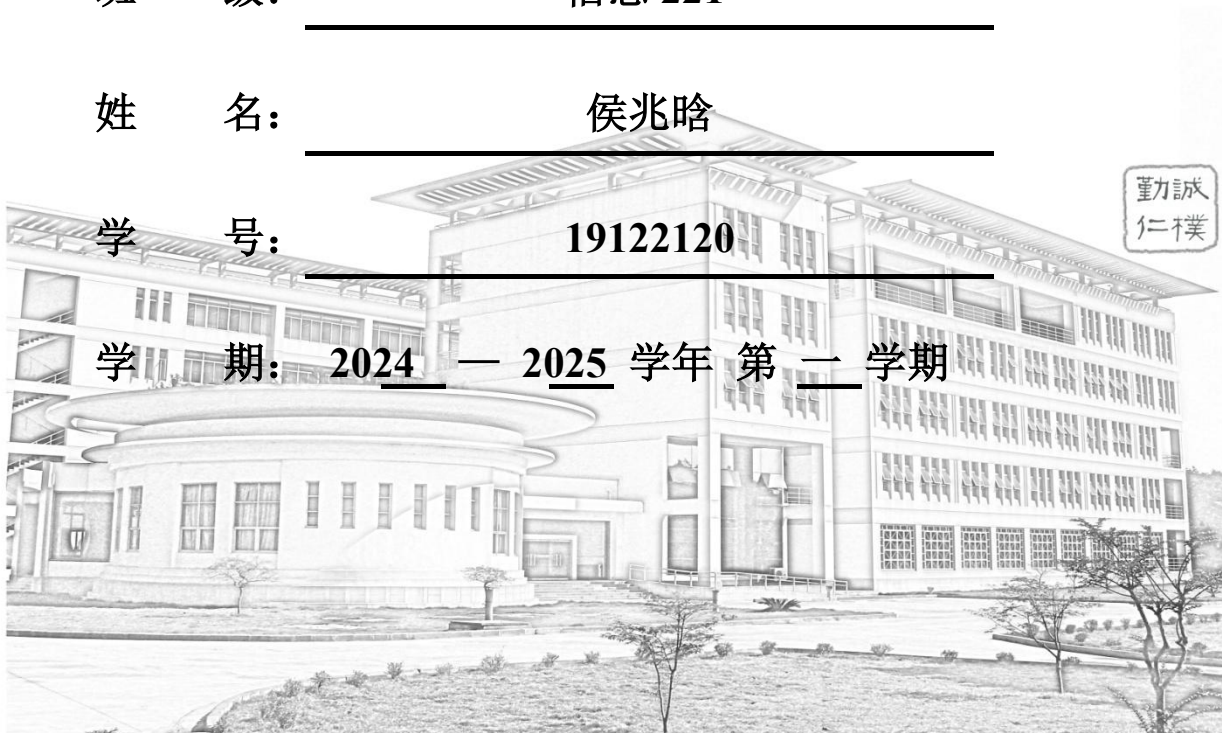
指导老师: 邹修国

班级: 信息 221

姓名: 侯兆晗

学号: 19122120

学期: 2024 — 2025 学年 第 一 学期



勤誠
仁樸

南京农业大学教务处印



实 验 目 录

实验一：发酵床网养种鹅冬季氨气分析预测

实验二：基于 EEMD-GRU 的猪舍二氧化碳浓度预测模型

研究

实验三：基于 VGG 的 17 类花卉分类

实验四：基于 YOLOv5 的鸡热应激行为识别



姓名：侯兆晗

学号：19122120

班级：信息 221

成绩：

实验一

一、实验目的

发酵床网养种鹅冬季氨气分析预测

根据联合国粮农组织（FAO）统计，禽肉消费比例逐年增长，从全球范围内看，禽肉将逐步取代猪肉的地位。鹅业作为我国传统养殖业，自 21 世纪以来，我国肉鹅出栏量一直位居世界首位，约占全球产量 90% 左右。

原因：氨气主要是由粪便以及潮湿垫料的发酵产生。

危害：(1)使种鹅处于亚健康状态，影响种鹅体重的增重和生产性能。

(2)机体的免疫力降低。

(3)氨气含量高，氧的含量相对较低；表现为精神萎靡，食欲减退。

(4)严重的会发生疾病：猝死症、腹水症、鹅的慢性呼吸道疾病。

(5)氨气会损害工作人员的健康，影响周围居民的环境质量。

因此，对发酵床网养种鹅冬季的氨气进行准确的分析预测具有重要意义。

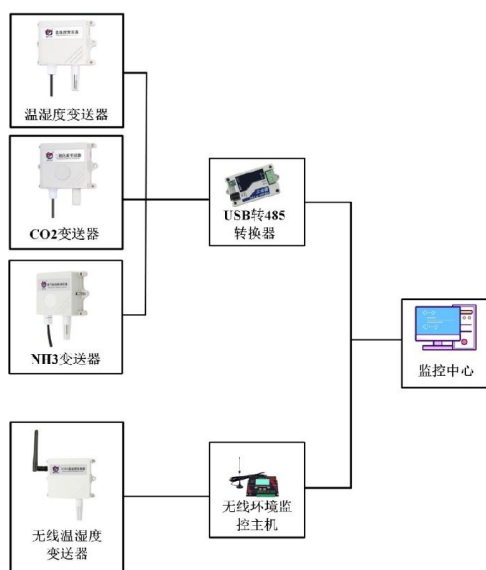
二、实验原理

(1) 数据采集

试验变送器布点

舍外：鹅舍外屋檐下安装温湿度变送器和 HOBO 温湿度记录仪来记录鹅舍外的温湿度情况。

舍内：Fan2 和 Fan3 中心位置安装温湿度变送器，离地高度 1.65m；在种鹅生活区中心线等间距安装 5 组温湿度、NH₃ 和 CO₂ 变送器，离网床高度 1m；在湿帘处安装两组无线温湿度变送器，离网床高度 1m；生活区变送器间距为 7.2m，所有每隔一分钟记录一次数据。



环境监测系统



(2) 数据预处理

• 小时均值处理：

整理传感器每分钟采集到的数据，将 1 小时内测量到的 60 次数据加和平均处理，便于后续的数据处理和模型建立。

小时均值处理公式：

$$x_h = \frac{\sum_{i=0}^{60} x_i}{60}$$

式中： x_h 是小时均值处理后数据， x_i 是每分钟各采样点数据。

• 数据归一化：

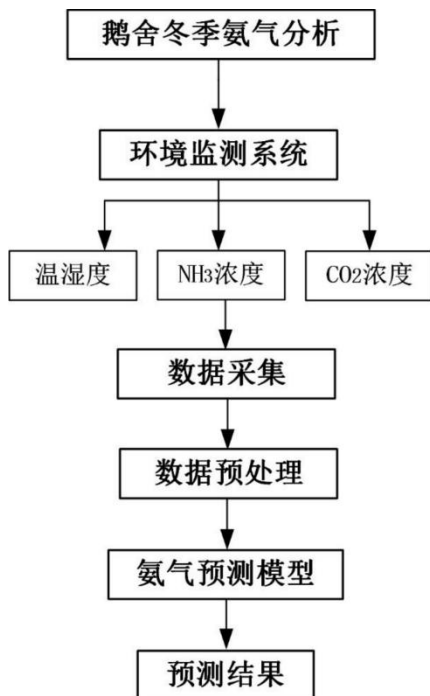
为提高算法收敛速度和精度，使模型建立、学习、训练和预测的效果更好，需要对数据进行标准化处理。

本实验采用数据归一化方法中的最大最小值归一化法，即线性函数归一化法。其原理是：通过使用数据集中数据的最大值和最小值进行标准化处理，使得处理后的数据集中在大于 0 小于 1 的区间范围内，具体公式为：

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}}$$

式中： X^* 为归一化处理数据， X 是采集的环境参数， X_{max} 、 X_{min} 是环境参数中最大值与最小值。

三、实验步骤



(1) 时间序列与监督学习

在可以使用机器学习之前，时间序列预测问题必须重新构建成监督学习问题，从一个单纯的序列变成一对序列输入和输出。



定义一个名为 `series_to_supervised()` 的新 Python 函数，它采用单变量或多变量时间序列，并将其作为监督学习数据集。

该函数有四个参数：

- `data`: 序列，列表或二维数组。
- `n_in`: 用于输入数据步数(x)。值可能介于 $[1, \text{len}(\text{data})]$ ，可选参数。
- `n_out`: 作为输出数据步数(y)。值可能介于 $[1, \text{len}(\text{data})]$ ，可选参数。
- `dropnan`: 用于滤除缺失数据。可选参数。默认为 `True`。

代码实现：

- 首先使用 `MinMaxScaler()` 函数对数据进行归一化处理。
- 然后通过 `series_to_supervised()` 函数将数据转换为有监督的数据。
- 最后利用 `drop()` 函数删除不预测的列。

其中删除列数为特征数(feature)-1，

起始列为 `n_in*feature+1`。

(2) 数据集划分

- 将数据集 70% 作为训练集，30% 作为测试集。
- 通过 `reshape()` 函数将训练集与测试集转化为 3 维，三个参数分别为：数据集行数(`shape[0]`)、输入序列步数(`n_in`)、特征数(feature)

(3) GRU 模型构建

GRU 模型代码如下图所示，

- 隐藏层数为 1，
- 神经元个数为 35，
- 输出层维度为 1，
- Epoch 为 1200，
- `Batc_size` 为 120，
- 损失函数为 `mae`，
- 优化器可选 `sgd` 与 `adam` 优化器。

(4) GRU 模型调参

根据需求对神经元个数及网络层数进行选择，如右图所示。

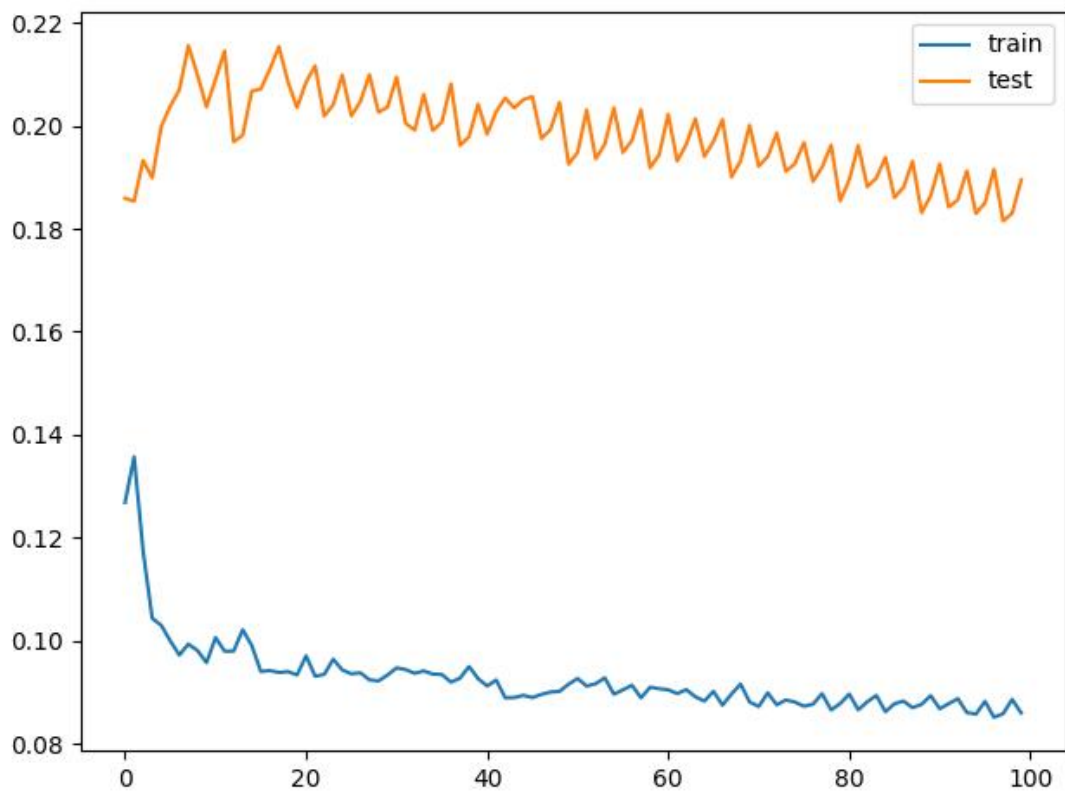
此外，可对 `Batc_size`、学习率等参数进行优化。

为防止过拟合，可采用 `Dropout` 方法，随机选择神经层中的一些单元并将其临时隐藏。

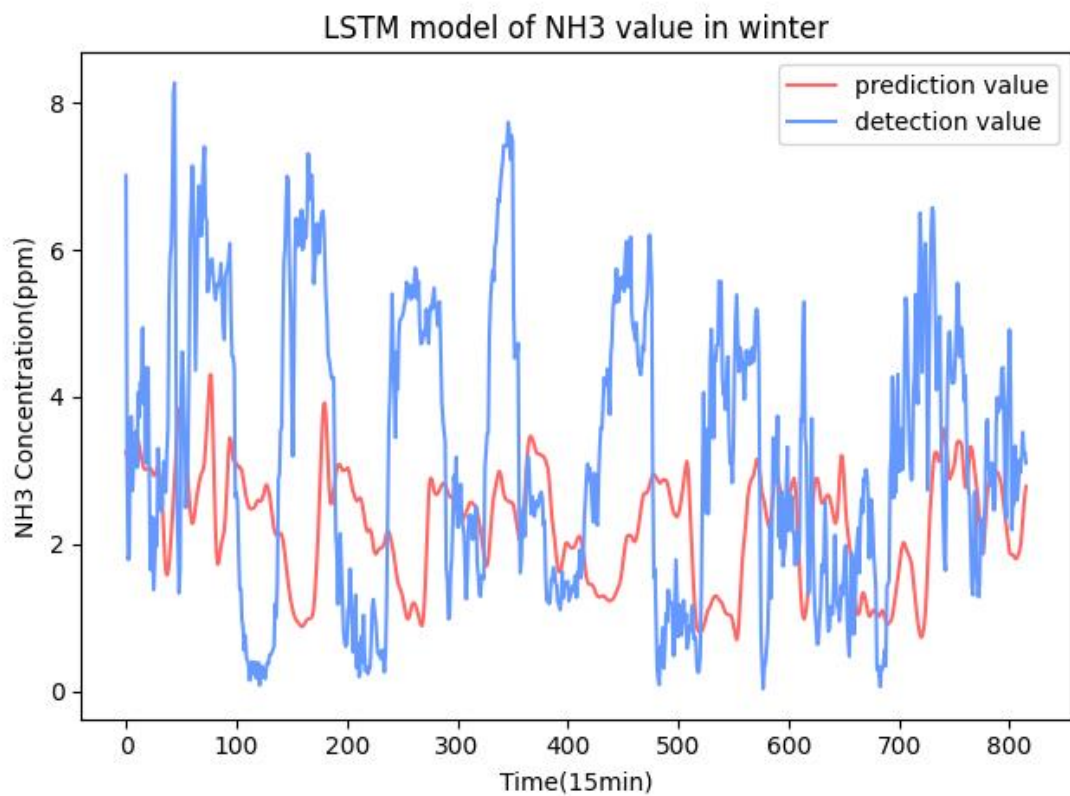
(5) 评价与制图

将 RMSE 与 MAE 作为评价指标，最后绘制预测值与真实值曲线图。

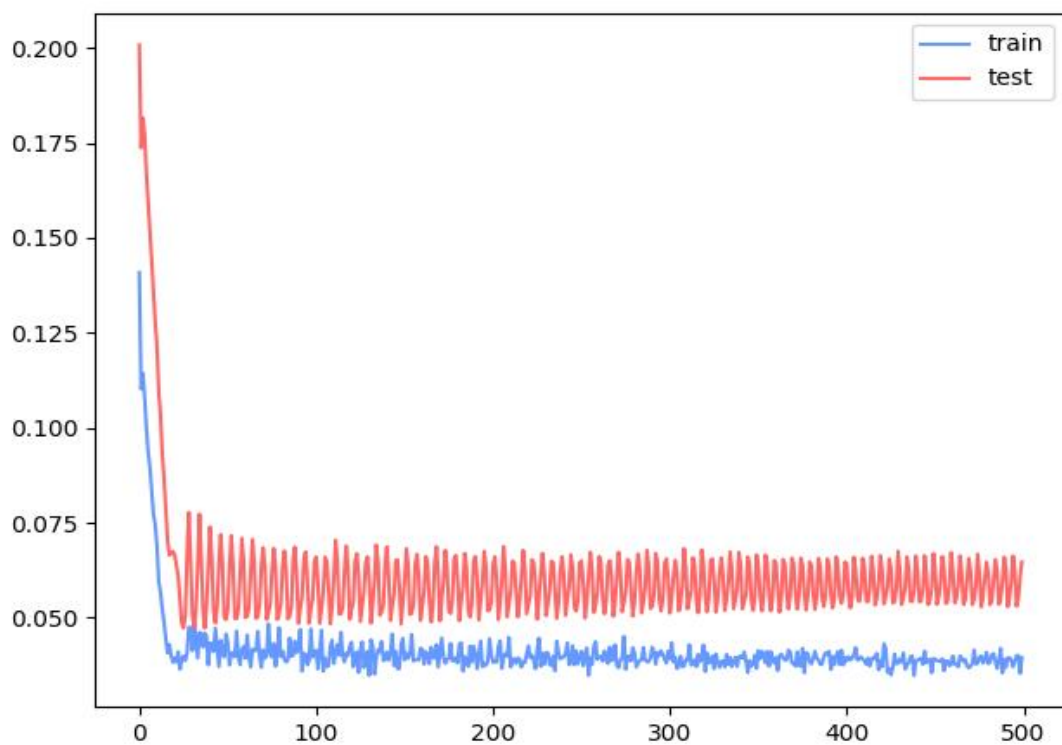
四、实验结果



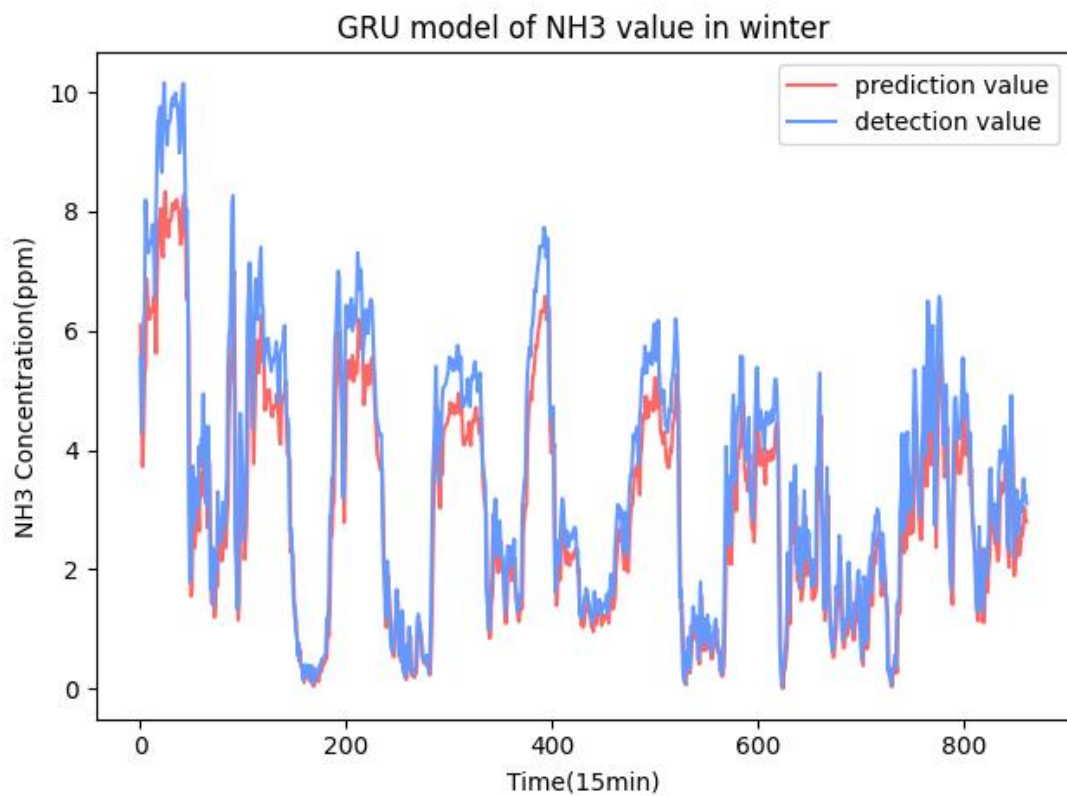
LSTM-M.py 训练集和测试集运行结果



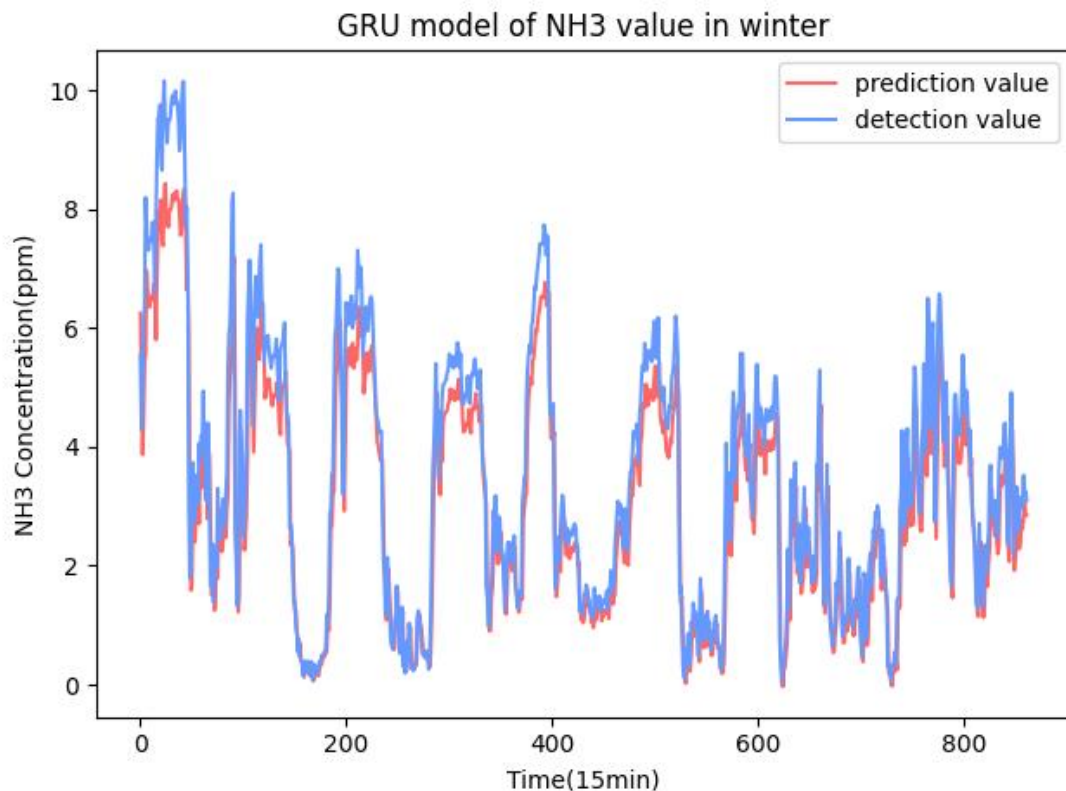
LSTM-M.py 运行后 NH3 预测值和测量值比较结果



GRU.py 训练集和测试集运行结果



在 epochs=500, batch_size=120 参数下 GRU 预测值和测量值对比图



在 epochs=250, batch_size=60 参数下 GRU 预测值和测量值对比图

五、实验总结

本实验通过系统的数据采集、预处理、模型搭建和训练，成功地建立了基于 GRU 的发酵床网养种鹅冬季氨气预测模型。绘制了预测值与真实值的曲线图，从图中可以看出，epochs=250, batch_size=60 参数下 GRU 预测值和真实值更加接近，而 epochs=500, batch_size=120 参数下 GRU 预测值和真实值相差更加大。通过训练 GRU 模型，能够准确地预测发酵床网养种鹅冬季的氨气浓度变化趋势。评价指标 RMSE（Root Mean Square Error）和 MAE（Mean Absolute Error）均较低，表明模型预测值与真实值之间的误差较小。

实验结果表明，GRU 模型在处理时序数据方面表现出色，能够有效地预测鹅舍内的氨气浓度。这不仅有助于改善种鹅的养殖环境，减少氨气对种鹅健康和生产性能的影响，还能为鹅业养殖提供科学依据和技术支持。

本实验还验证了数据预处理和模型调参在深度学习中的重要性。通过小时均值处理和数据归一化处理，提高了数据的质量和算法收敛速度；通过调整模型参数和采用 Dropout 方法，进一步优化了 GRU 模型的性能。



姓名：侯兆晗

学号：19122120

班级：信息 221

成绩：

实验二

一、实验目的

基于 EEMD-GRU 的猪舍二氧化碳浓度预测模型研究

本实验在农业农村部饲料工业中心动物试验基地妊娠猪舍进行了数据采集，主要采集猪舍内温度、湿度、风速和二氧化碳浓度数据。

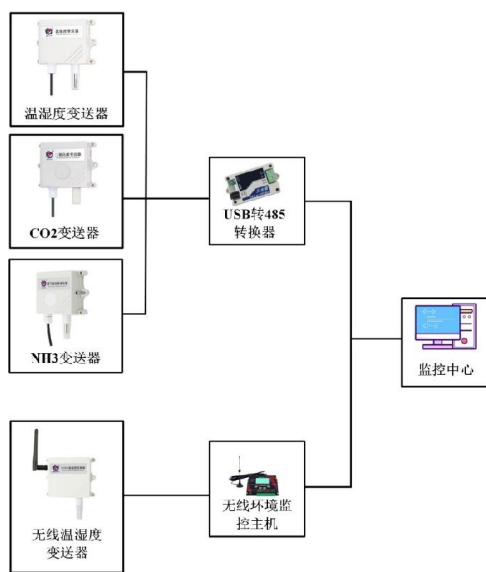
猪舍地点位于河北省承德市丰宁满族自治县，实验猪舍面积为 $12 \times 8 \text{m}^2$ ，采用双排限位栏和水泥地面，饲养 30 头妊娠前期的母猪，采用头对头饲养方式，中间是过道，后面是排水沟。

猪舍环境传感器购于山东省青岛大牧人机械股份有限公司，二氧化碳传感器测量范围为 0-5000ppm，湿度传感器采用 HTV597 型号，环境控制器采用 BH8118 型号，可监测显示并调控猪舍内温度、湿度和风速值等。

基于以上背景对猪舍二氧化碳浓度进行分析预测。

二、实验原理

(1) 数据收集



环境监测系统

(2) 数据预处理：

- 小时均值处理：

整理传感器每 10 分钟采集到的数据，将 1 小时内测量到的 6 次数据加和平均处理，便于后续的数据处理和模型建立。

小时均值处理公式：

$$x_h = \frac{\sum_{i=0}^6 x_i}{6}$$



式中： x_h 是小时均值处理后数据， x_i 是每 10 分钟各采样点数据。

- 数据归一化：

为提高算法收敛速度和精度，使模型建立、学习、训练和预测的效果更好，需要对数据进行标准化处理。

本实验采用数据归一化方法中的最大最小值归一化法，即线性函数归一化法。其原理是：通过使用数据集中数据的最大值和最小值进行标准化处理，使得处理后的数据集中在大于 0 小于 1 的区间范围内，具体公式为：

$$X^* = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

式中： X^* 为归一化处理后数据， X 是采集的环境参数， X_{\max} 、 X_{\min} 是环境参数中最大值与最小值。

三、实验步骤

(1) 时间序列与监督学习：

在可以使用机器学习之前，时间序列预测问题必须重新构建成监督学习问题，从一个单纯的序列变成一对序列输入和输出。

定义一个名为 `series_to_supervised()` 的新 Python 函数，它采用单变量或多变量时间序列，并将其作为监督学习数据集。

该函数有四个参数：

- `data`：序列，列表或二维数组。
- `n_in`：用于输入数据步数(x)。值可能介于 $[1, \text{len}(\text{data})]$ ，可选参数。
- `n_out`：作为输出数据步数(y)。值可能介于 $[1, \text{len}(\text{data})]$ ，可选参数。
- `dropnan`：用于滤除缺失数据。可选参数。默认为 `True`。

代码实现：

- 首先使用 `MinMaxScaler()` 函数对数据进行归一化处理。
- 然后通过 `series_to_supervised()` 函数将数据转换为有监督的数据。
- 最后利用 `drop()` 函数删除不预测的列。

(2) 数据集制作：

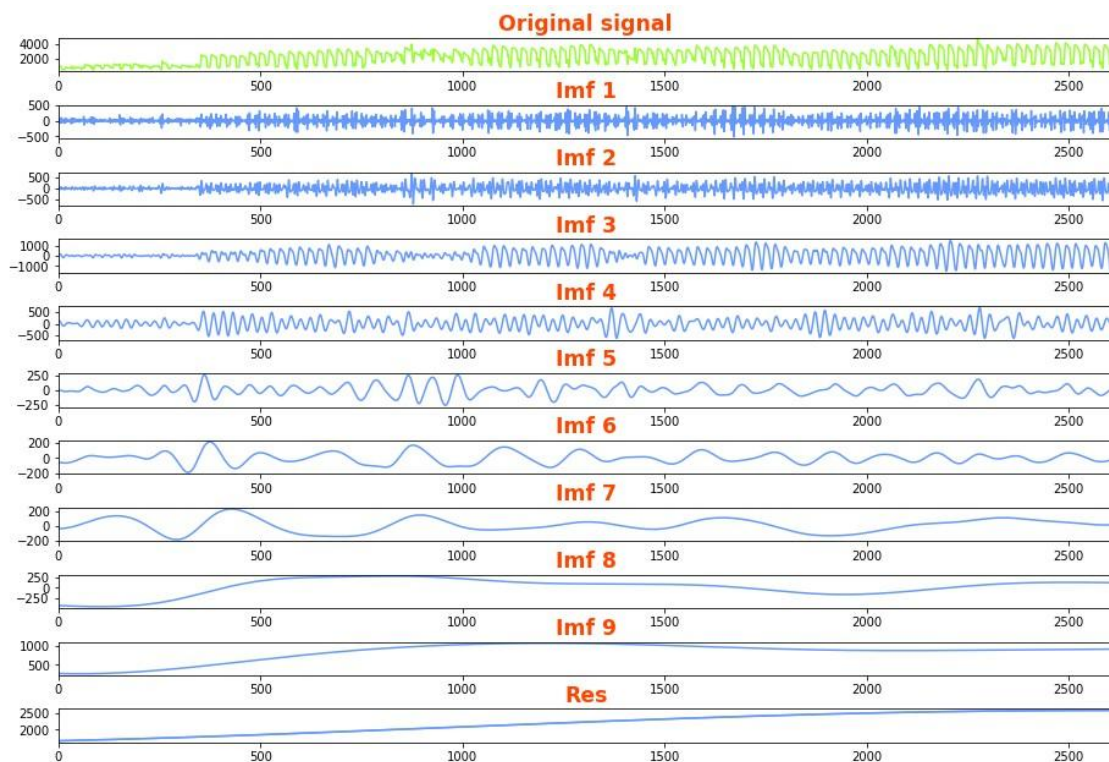
将前两小时的环境数据以及二氧化碳数据进行整合作为一个样本，所以模型的输入特征数据分别是（共计 8 维数据特征）：

- 两小时前的二氧化碳浓度
- 两小时前的温湿度
- 两小时前的风速
- 一小时前的二氧化碳浓度
- 一小时前的温湿度
- 一小时前的风速

(3) 数据集划分：

- 将数据集 70% 作为训练集，30% 作为测试集。
- 通过 `reshape()` 函数将训练集与测试集转化为 3 维，三个参数分别为：数据集行数(`shape[0]`)、输入序列步数(`n_in`)、特征数(`feature`)

(4) 对 EEMD 进行分解。



EEMD 分解结果

(5) GRU 模型构建: GRU 模型参数设置如下,

- 隐藏层数为 1,
- 神经元个数为 35,
- 输出层维度为 1,
- Epoch 为 200,
- Batch_size 为 128,
- 损失函数为 mae,
- 优化器可选 adam 优化器。

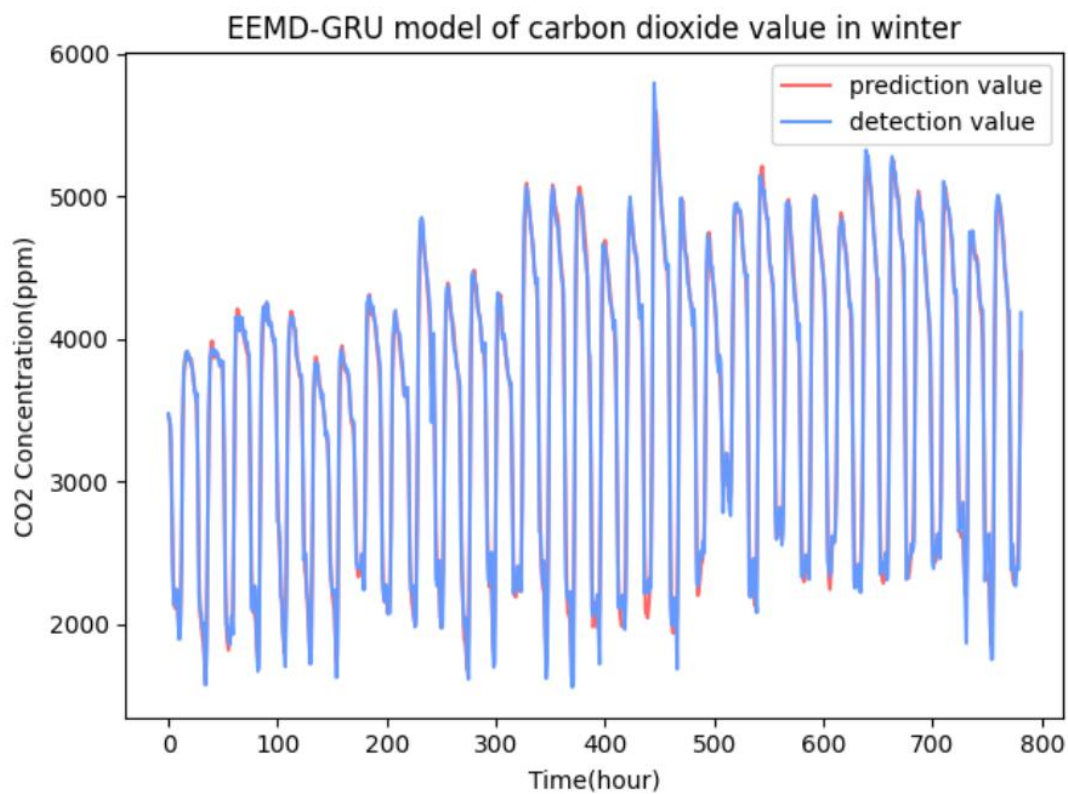
(6) GRU 模型调参: 根据需求对神经元个数及网络层数进行选择。

此外, 可对 Batch_size、学习率等参数进行优化。

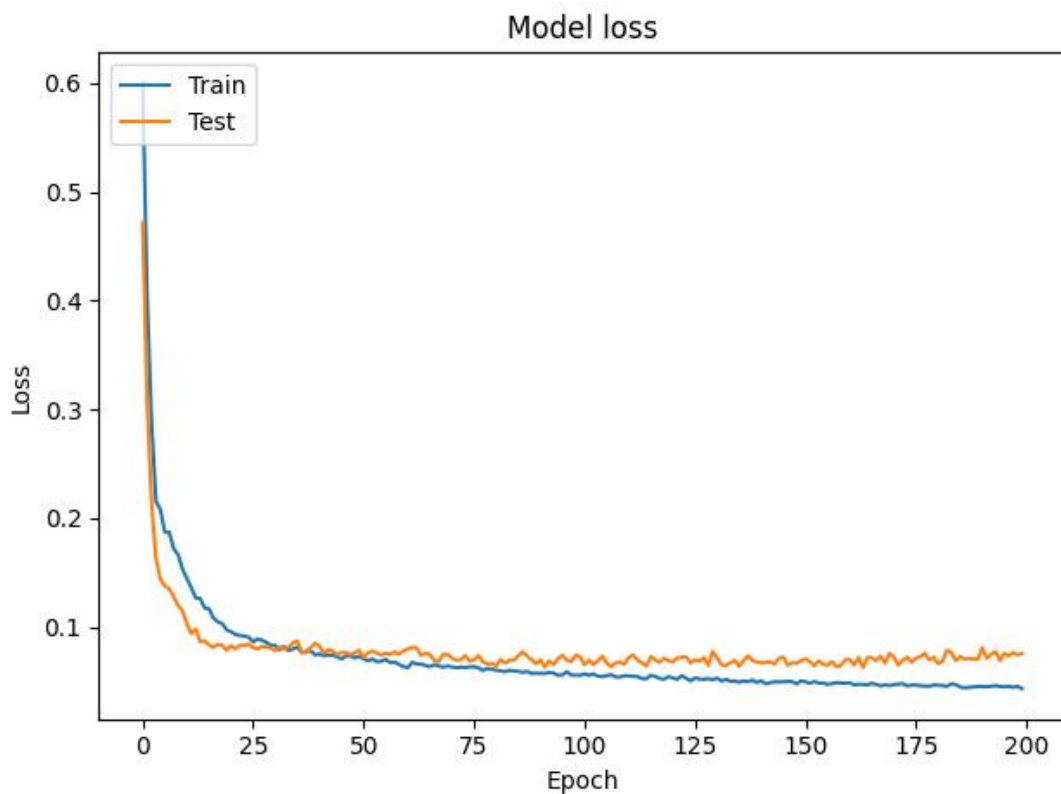
为防止过拟合, 可采用 Dropout 方法, 随机选择神经层中的一些单元并将其临时隐藏。

(7)评价与绘图: 将 RMSE(root mean squared error)与 MAE(mean absolute error)作为评价指标, 最后绘制预测值与真实值曲线图。

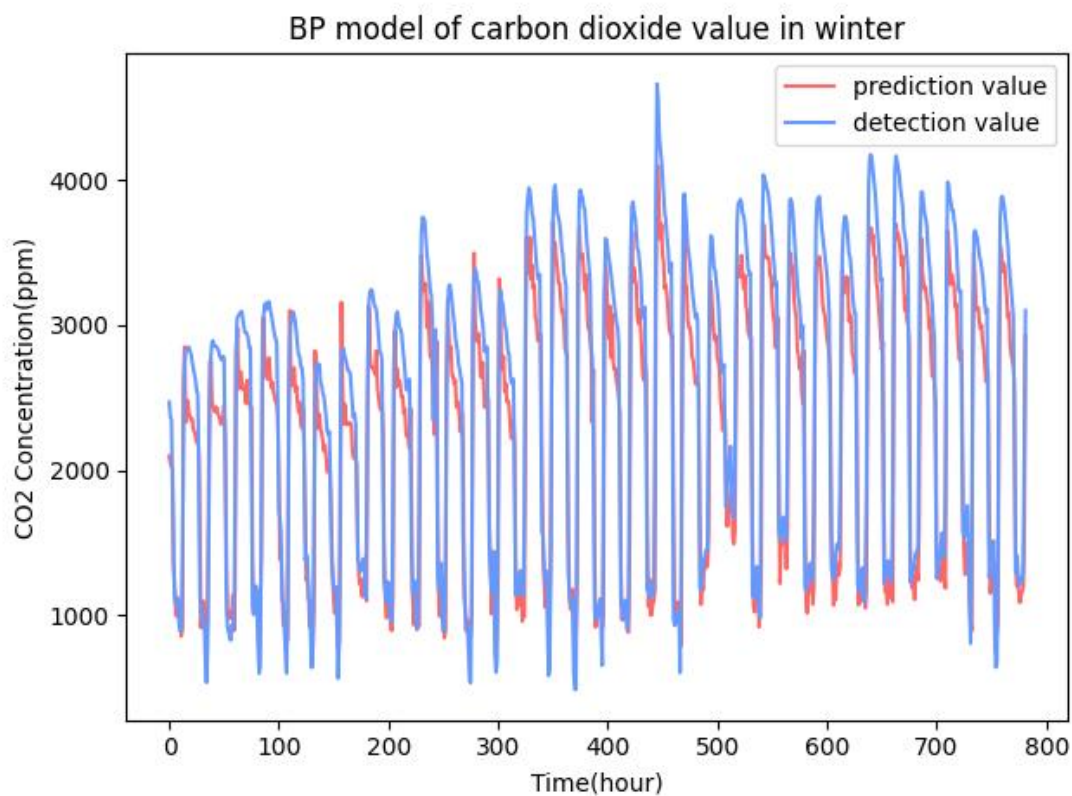
四、实验结果



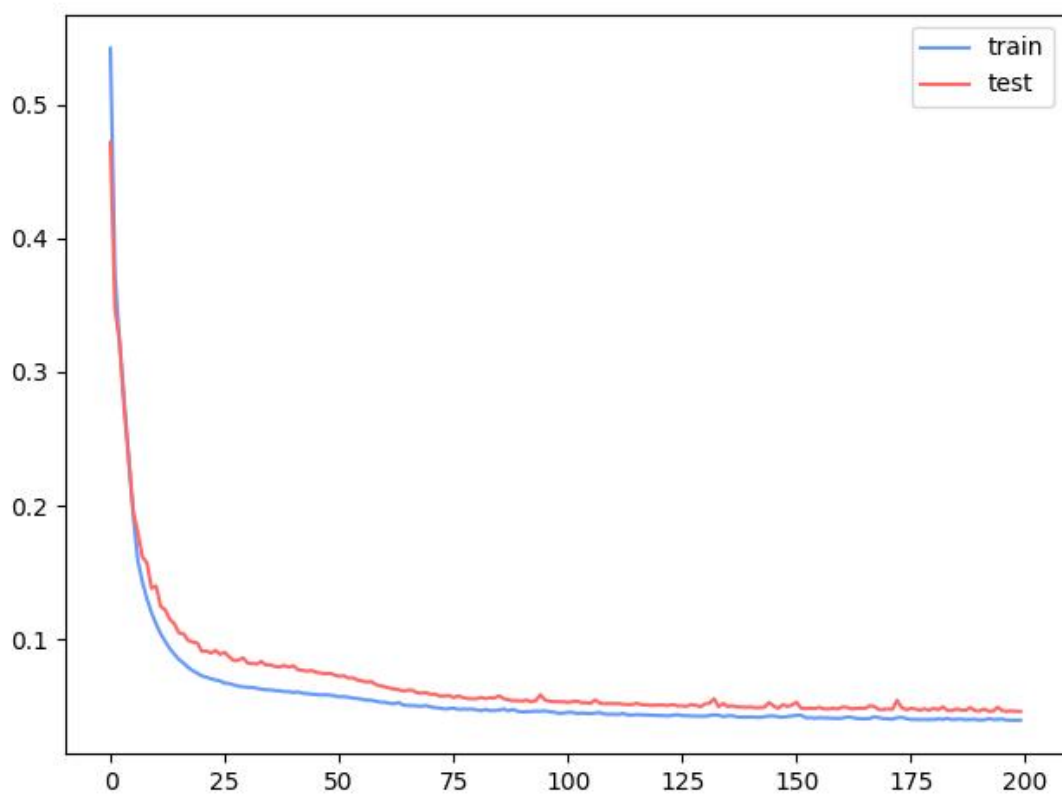
EEMD-GRU.py 运行出的 CO2 预测值和测量值对比图



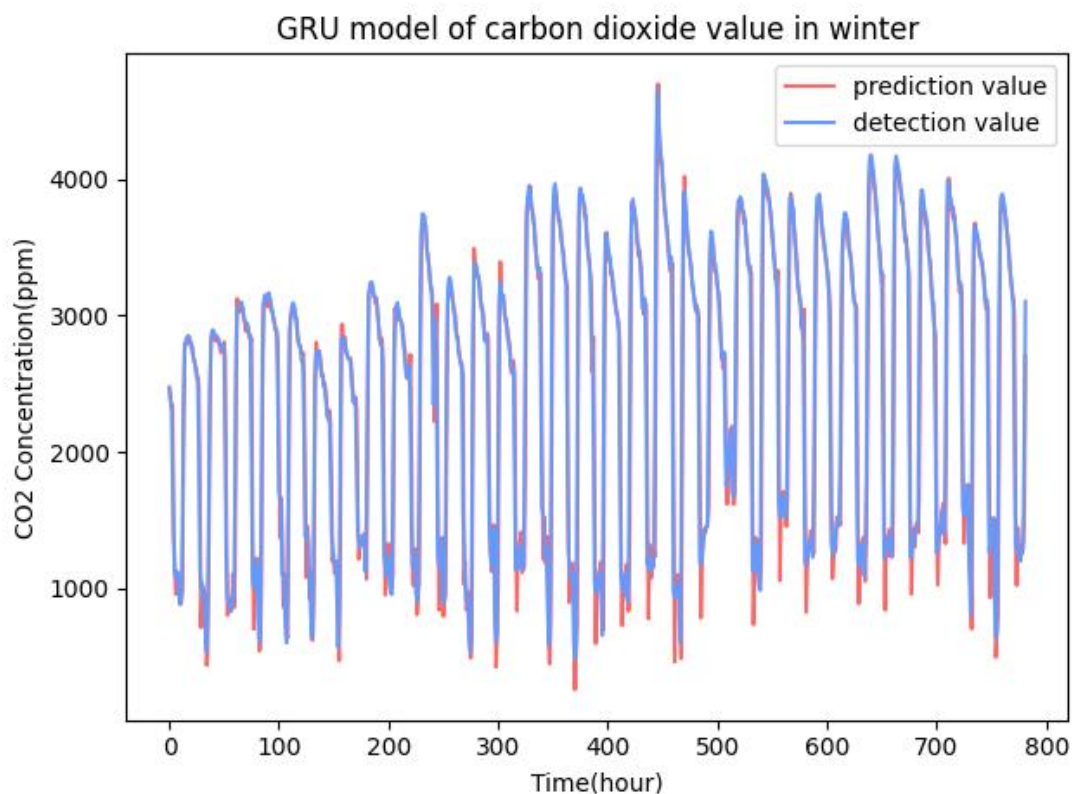
WinBP.py 运行出的 Model loss 训练集和测试集结果



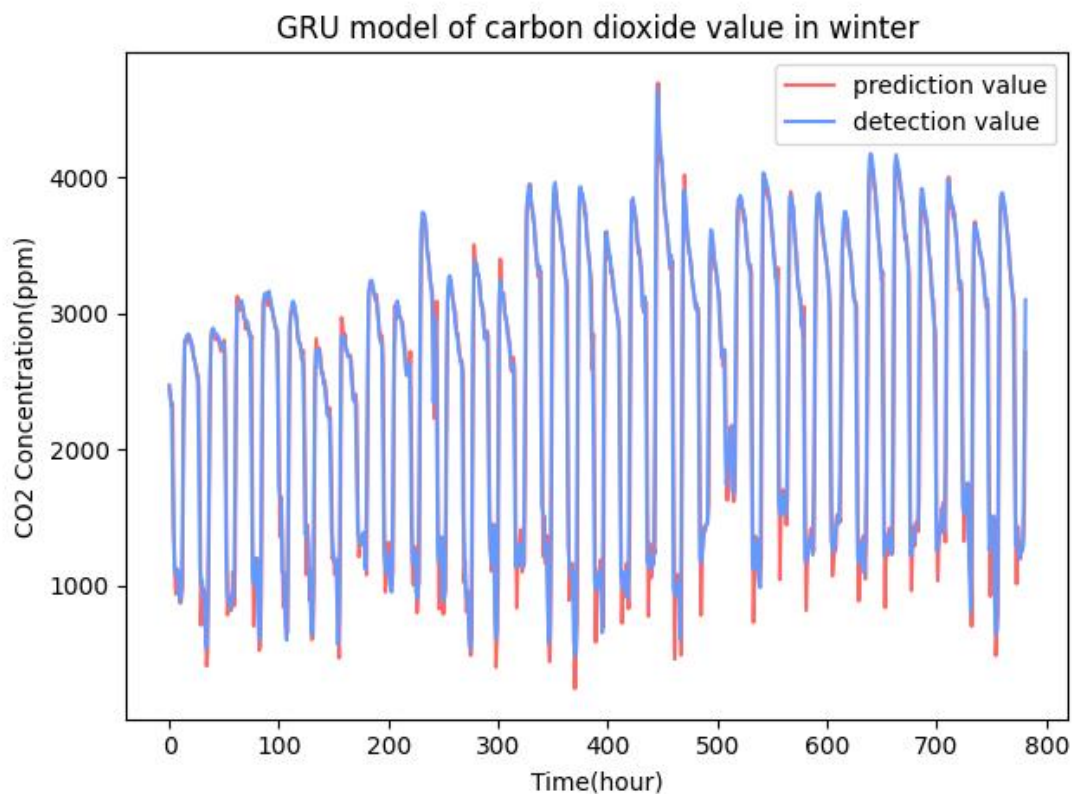
WinBP.py 运行出的 BP 模型的 CO₂ 预测值和测量值对比图



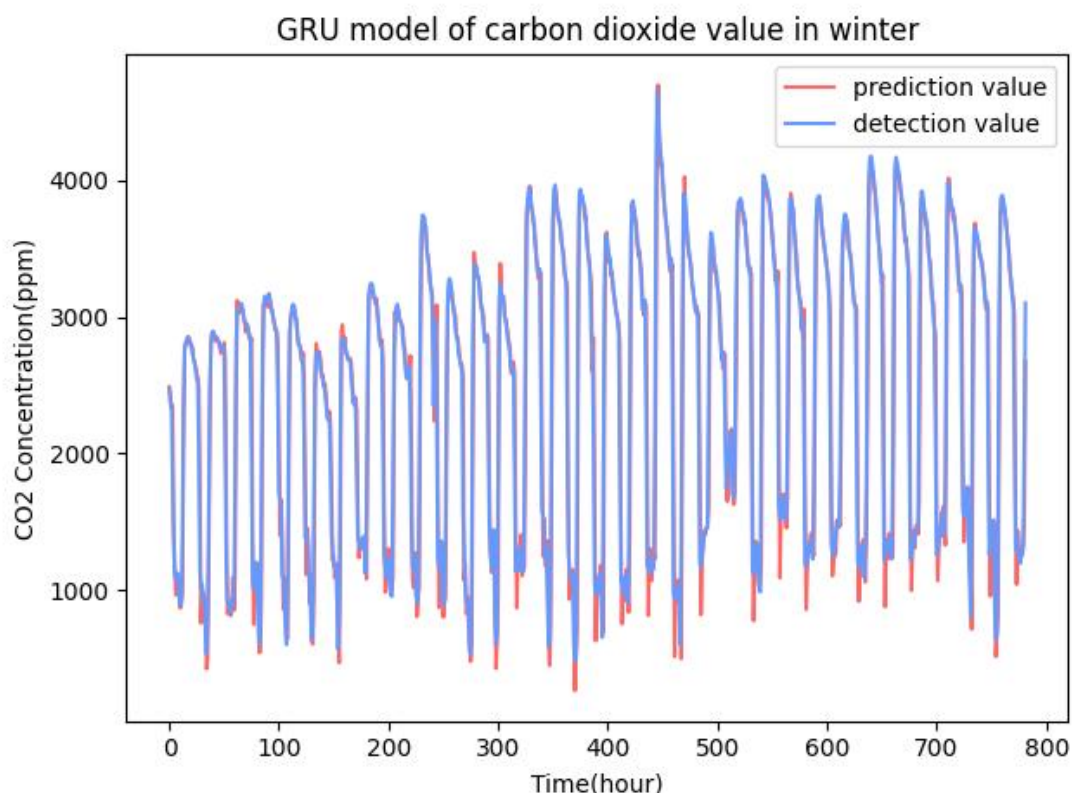
GRU.py 运行后的训练集和测试集结果



在 epochs=200, batch_size=128 参数下 GRU 对 CO₂ 预测值和真实值对比图



在 epochs=400, batch_size=256 参数下 GRU 对 CO₂ 预测值和真实值对比图



在 epochs=100, batch_size=64 参数下 GRU 对 CO₂ 预测值和真实值对比图

五、实验总结

本研究通过构建基于 EEMD-GRU 的猪舍二氧化碳浓度预测模型，实现了对猪舍内二氧化碳浓度的准确预测。实验结果表明，该模型具有较高的预测精度和稳定性，能够为猪舍环境管理提供科学依据。

通过 EEMD 分解，实验成功提取了原始数据中的潜在特征，为后续的 GRU 模型构建提供了有力的支持。同时，GRU 模型凭借其强大的时间序列处理能力，有效地捕捉了猪舍内二氧化碳浓度的变化趋势。在 epochs=200, batch_size=128 参数下 GRU 预测值和真实值最为接近，在 epochs=400, batch_size=256 参数下和在 epochs=100, batch_size=64 参数下 GRU 预测值和真实值相对较差。说明在 epochs=200, batch_size=128 参数下 GRU 更能够准确预测出猪舍内二氧化碳浓度。

本研究通过模型调参等一系列措施，进一步提高了模型的预测性能。



姓名：侯兆晗

学号：19122120

班级：信息 221

成绩：

实验三

一、实验目的

基于 VGG 的 17 类花卉分类

花卉分类作为植物学领域中最常见的基础工作，是对花卉种类习性等深入研究的必要条件。目前已知有超过 250000 种开花植物，大约可分为 350 个科。花卉种类繁多，且这其中很多花卉具有相同特征，不同类型的花卉可能具有相似的颜色、形状和外观等特征。对于没有深入了解植物花卉的普通人来说，脱离植物学专业人士引导后能够独立识别常见的花卉仍是一个具有较大难度的工作。

基于以上背景利用卷积神经网络帮助进行花卉分类。

二、实验原理

(1) 图像分类方法

传统机器学习算法：

- 问题迁移，即将多标签分类问题转化为单标签分类问题，如将标签转化为向量、训练多个分类器等；
- 根据多标签特点，提出新的适应性算法，包括 KNN、SVM、Decision Tree 等。

深度学习算法：CNN(1980s)，AlexNet(2012)，VGG(2014)等多种图像分类算法。

CNN-Convolutional Neural Network

VGGNet 是牛津大学计算机视觉组（Visual Geometry Group）和谷歌 DeepMind 一起研究出来的深度卷积神经网络，因而冠名为 VGG。

(2) 数据集介绍

数据集为 17 Category Flower Dataset，是牛津大学 Visual Geometry Group 选取的在英国比较常见的 17 种花；其中每种花有 80 张图片，整个数据集有 1360 张图片。

(3) 算法介绍

本次所用的 vgg16 模型是卷积神经网络的一种，包括 13 个卷积层和 3 个全连接层。卷积过程是使用 1 个卷积核，在每层像素矩阵上不断按步长扫描下去，每次扫描到的数值会和卷积核中对应位置的数进行相乘，然后相加求和，得到的值将会生成一个新的矩阵。

卷积核相当于卷积操作中的一个过滤器，用于提取我们图像的特征，卷积核的大小一般选择 3x3 和 5x5，比较常用的是 3x3，训练效果会更好。

(4) 激活函数

ReLU 函数：

- 解决了梯度消失的问题（在正区间）
- 计算速度非常快，只需要判断输入是否大于 0
- 收敛速度远快于 sigmoid 和 tanh



- 缓解了过拟合问题的发生

Softmax 函数:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{c=1}^C e^{z_c}}$$

引入指数形式的优点,使用指数形式的 Softmax 函数能够将差距大的数值拉的更大,适用于多分类问题。

三、实验步骤

(1) 图片读取

- 建立空的列表 X,Y
- 遍历文件夹,用 `resize()`函数将图片大小缩放到同一尺寸
- 用 `append()` 函数将图像数据添加到列表 X 中。
- 用 `append()` 函数将对应的文件名称添加到列表 Y 中。

(2) 数据处理

数据处理代码如下图所示,

- 用 `np.array()`函数将图片信息转换为矩阵形式
- 用 `np.save()`函数将数据存储为 `numpy` 文件
- 用 `np.load()`函数再次读取文件
- 建立字典,将花卉文件名进行类别分类

(3) 数据集划分

根据需求对数据集进行划分,这里划定训练集占 80%,测试集占 20%,同时希望每次划分的数据集结果都是一样的,将 `random_state` 参数设置为 1。

(rs 保持初始值和最后一个值,并随机化其余值)

(rs = 0 只是对所有数据执行正常随机)

(4) 模型建立

- 将层的列表传递给 `Sequential` 的构造函数,来创建一个 `Sequential` 模型
- 利用 `model.add()`函数添加层。
- 利用 `Model.compile()`函数完成模型训练的 BP 模式设置
- 注意:最后一层根据自己需要分类选择 `dense` 的节点数量,这里设置为 17

(5) 模型训练

为了防止过拟合加入 `EarlyStopping()`函数,在训练过程中,评价指标不在上升,将提前结束训练。

- 利用 `ModelCheckpoint()`实现断点续训功能。
- 利用 `model.fit()`函数进行模型训练。
- 将最终的模型保存为“`model1.h5`”,并用于之后的分类。

(6) 模型预测

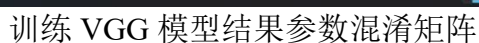
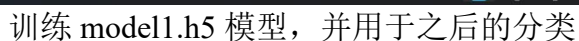
- 利用 `predict()`函数进行模型预测
- 调用 `confusion_matrix()`函数调用混淆矩阵,作为模型的评价指标

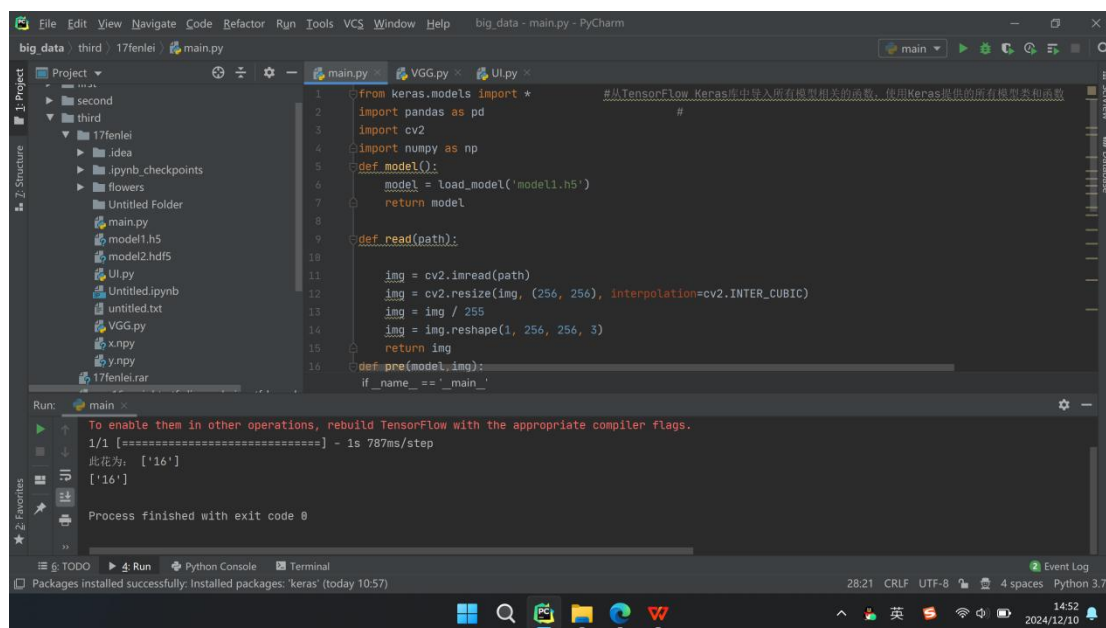
(7) 预测界面

- 运行上述所示模型 VGG,完成图像的训练。
- 建立新的 `main` 文件,编写预测代码。
- 建立新的 UI 文件进行预测界面设置,采用 `PyQt5` 设计,完成图片加载和

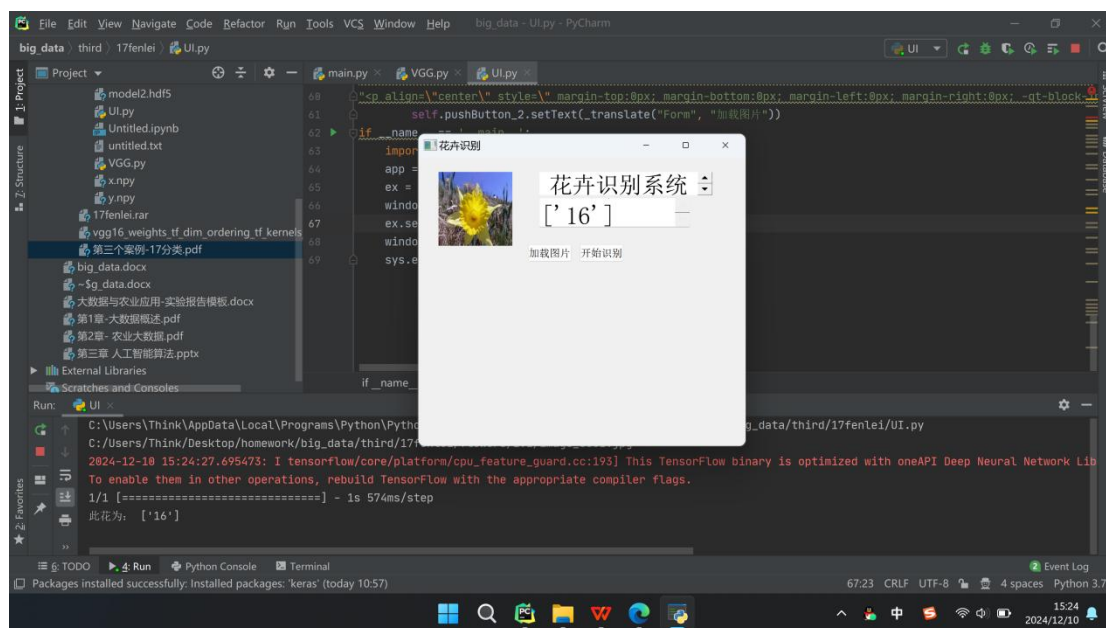


四、实验结果





直接调用主函数识别 flowers/image_0073.jpg 图片为第 16 类



使用人机交互程序 UI.py 进行预测界面设置（采用 PyQt5 设计）加载图片开始识别，得到 flowers/image_0073.jpg 图片为第 16 类。

五、实验总结

实验结果显示，VGG16 模型在 17 类花卉分类任务中表现出色，达到了较高的分类准确性。这是基于卷积神经网络强大的特征提取能力和 VGG16 模型较深的网络结构。

通过将数据集划分为训练集和测试集，实验验证了模型的泛化能力。结果显



示，模型在测试集上的表现与训练集相近，说明模型具有较好的泛化能力。

实验结果表明，卷积神经网络在花卉分类任务中具有强大的特征提取能力和分类准确性。同时，通过不断优化模型训练过程，可以进一步提升模型的性能。



姓名：侯兆晗

学号：19122120

班级：信息 221

成绩：

实验四

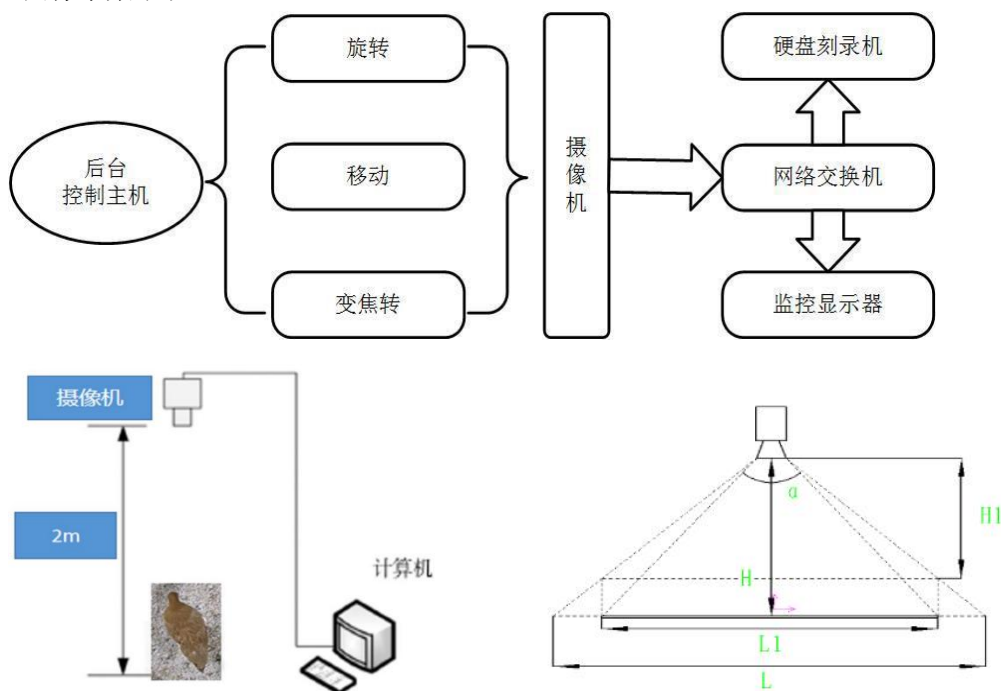
一、实验目的

基于 YOLOv5 的鸡热应激行为识别

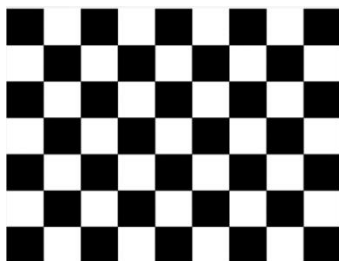
热应激是指在较高温度环境下肉鸡机体对热环境所做的非特异性反应的总和，包括生理生化反应、行为反应等。目前，环境参数法和行为分析法可用于快速判断肉鸡热应激，其中环境参数中常用温湿度指数（Temperature and Humidity Index, THI）进行判断，但此为间接指标，无法准确评估热应激。行为分析法则通过行为动作判断鸡只状态。随着视觉技术与人工智能算法的发展，利用图像分析行为方法更多地应用于家禽行为的监测，这已成为一个重要研究领域。

二、实验原理

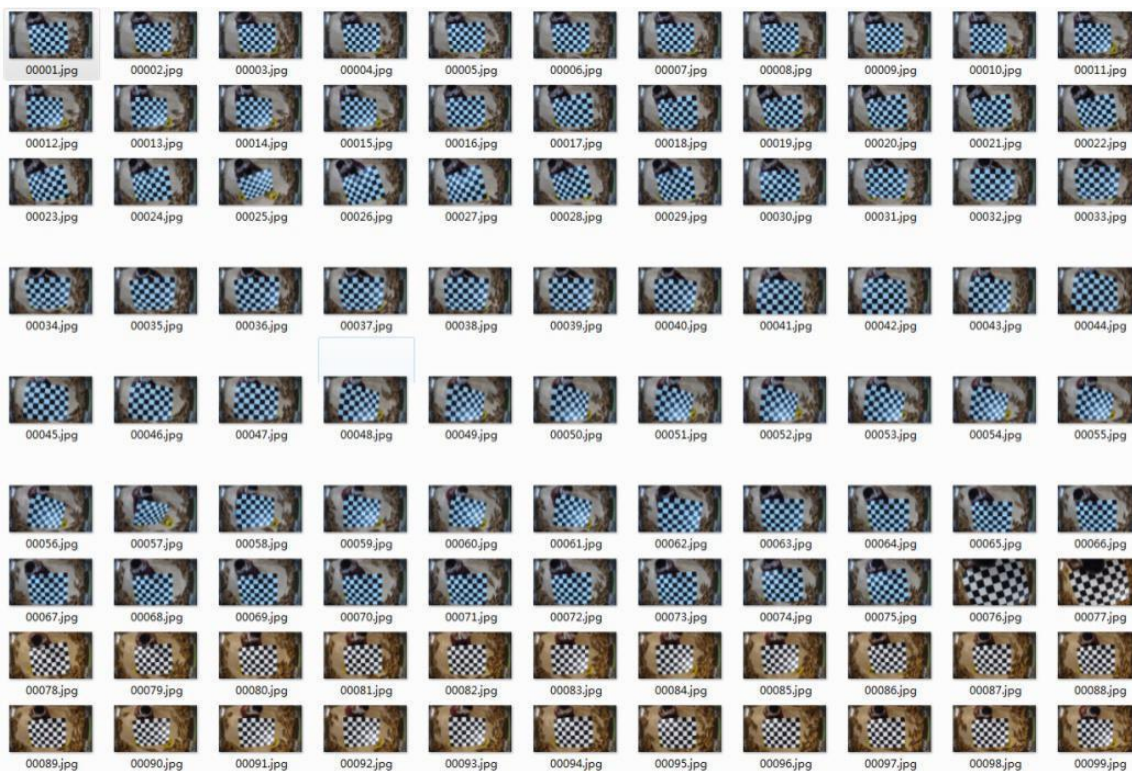
(1) 图像采集系统



(2) 张正友标定法



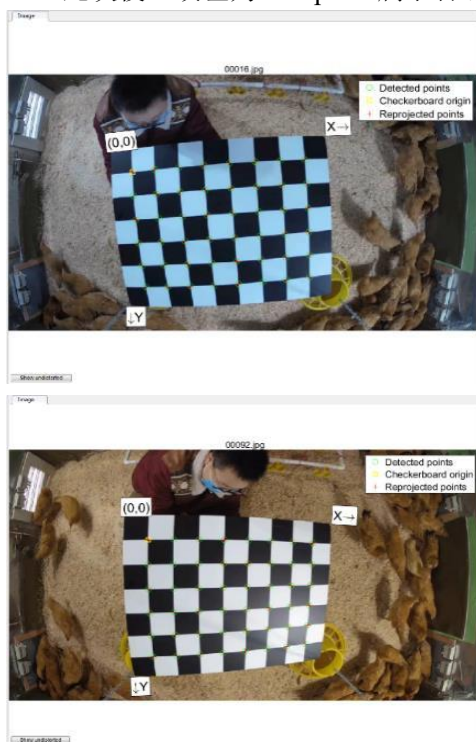
标定模版图



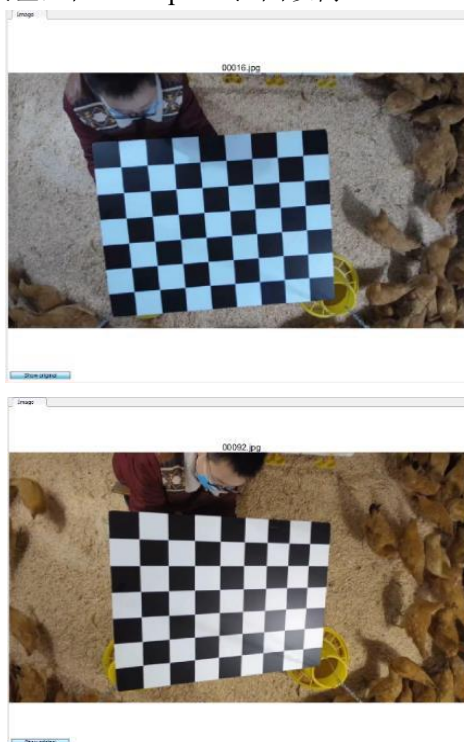
摄像机拍摄的不同角度的棋盘格图片

$$K = \begin{bmatrix} 505.4552 & 0 & 0 \\ 0.1921 & 507.391 & 0 \\ 429.491 & 252.5465 & 1 \end{bmatrix}$$

此次校正误差为 0.18pixel 属于合理误差，在 0-0.5 pixel 范围以内。



(a) 校正前图



(b) 校正后图



(3) 行为分类

分类名称	分类定义
啄食 (Eating)	鸡只头部伸入食槽, 与食槽有接触
饮水 (Drinking)	鸡只头部伸向喂水器及水杯, 与喂水器和水杯有接触
热应激行为 (Heat Stress)	鸡只双翅有明显抬高动作, 或者鸡只 俯卧在地
其它行为 (Others)	包括站立、行走、整理羽毛等

(4) PyTorch

PyTorch 是 torch 的 python 版本, 是由 Facebook 开源的神经网络框架。

Torch 是一个经典的对多维矩阵数据进行操作的张量 (tensor) 库, 在机器学习和其他数学密集型应用有广泛应用。与 Tensorflow 的静态计算图不同, PyTorch 的计算图是动态的, 可以根据计算需要实时改变计算图。

作为经典机器学习库 Torch 的端口, PyTorch 为 Python 语言使用者提供了舒适的写代码选择。

(5) YOLO 模型

“You Only Look Once” 或 “YOLO” 是一个对象检测算法的名字, 这是 Redmon 等人在 2016 年的一篇研究论文中命名的。YOLO 实现了自动驾驶汽车等前沿技术中使用的实时对象检测。

YOLO 将对象检测重新定义为一个回归问题。它将单个卷积神经网络(CNN)应用于整个图像, 将图像分成网格, 并预测每个网格的类概率和边界框。例如, 以一个 100x100 的图像为例。我们把它分成网格, 比如 7x7。

然后, 对于每个网格, 网络都会预测一个边界框和与每个类别 (汽车, 行人, 交通信号灯等) 相对应的概率。

YOLO 非常快。由于检测问题是一个回归问题, 所以不需要复杂的管道。它比 “R-CNN” 快 1000 倍, 比 “Fast R-CNN” 快 100 倍。它能够处理实时视频流, 延迟小于 25 毫秒。它的精度 是以前实时系统的两倍多。同样重要的是, YOLO 遵循的是 “端到端深度学习” 的实践。

三、实验步骤

(1) 鸡图像数据集的制作

- 图像重新命名
- 图像数据集增广 (旋转、平移等)
- Labelme 软件的制作
- Labelme 标准格式转化

(2) 模型训练

- 用 train.py 文件训练模型
- 训练好的模型会得到一个权重文件后缀名为.pth, 存放位置如下。
- 根据对模型精度的要求, 我们可以重复训练自己的模型直到有一个最优的

epoch



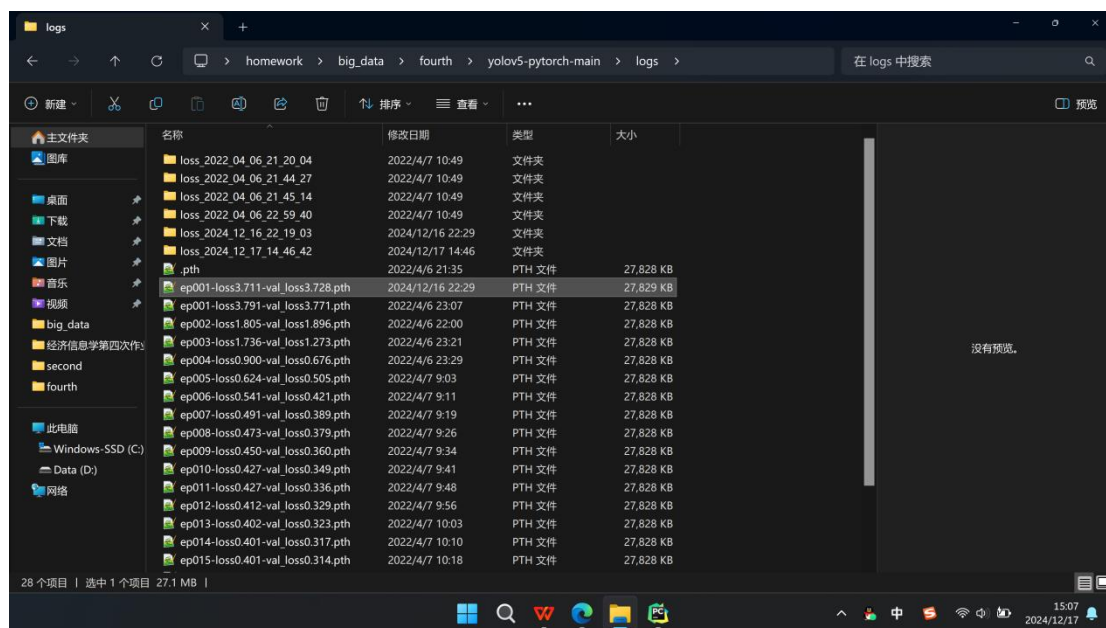
四、实验结果

```
151 #-----#
152 # Init_lr      模型的最大学习率
153 # Min_lr      模型的最小学习率，默认为最大学习率的0.01
154 #-----#
155 Init_lr      = 1e-2
156 Min_lr      = Init_lr * 0.01
157 #-----#
158 # optimizer_type 使用到的优化器种类，可选的有adam,sgd
159 #               当使用Adam优化器时建议设置 Init_lr=1e-3
160 #               当使用SGD优化器时建议设置 Init_lr=1e-2
161 # momentum     优化器内部使用到的momentum参数
162 # weight_decay  权重衰减，可防止过拟合
163 #               adam会导致weight_decay错误，使用adam时建议设置为0。
164 #-----#
165 optimizer_type = "sgd"
166 momentum      = 0.937
```

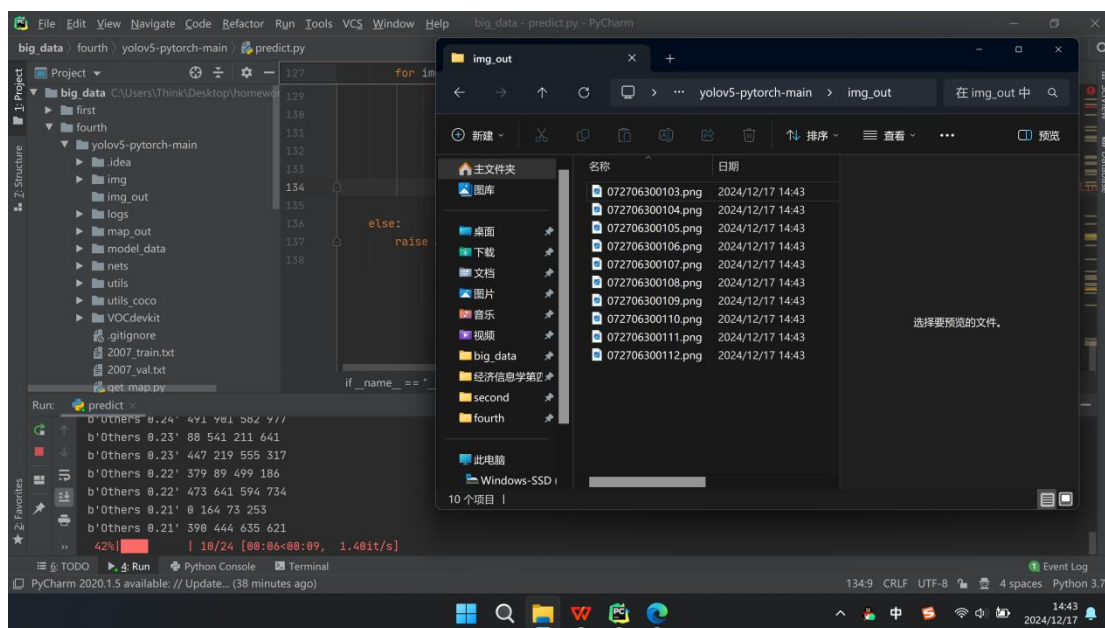
Run: train

```
Epoch 1/50: 0% | 0/6 [00:00<?, rlt/s<class 'dict'>]rtnisn train
Start Validation
Epoch 1/50: 100% | 6/6 [00:38<00:00, 6.38s/it, val_loss=3.73]
Finish Validation
Epoch:1/50
Total Loss: 3.711 || Val Loss: 3.728
Epoch 2/50: 0% | 0/6 [00:00<?, ?it/s<class 'dict'>]Start Train
Epoch 2/50: 15% | 9/6 [01:46<00:40, 10.21s/it, loss=3.42, lr=0.00025]
```

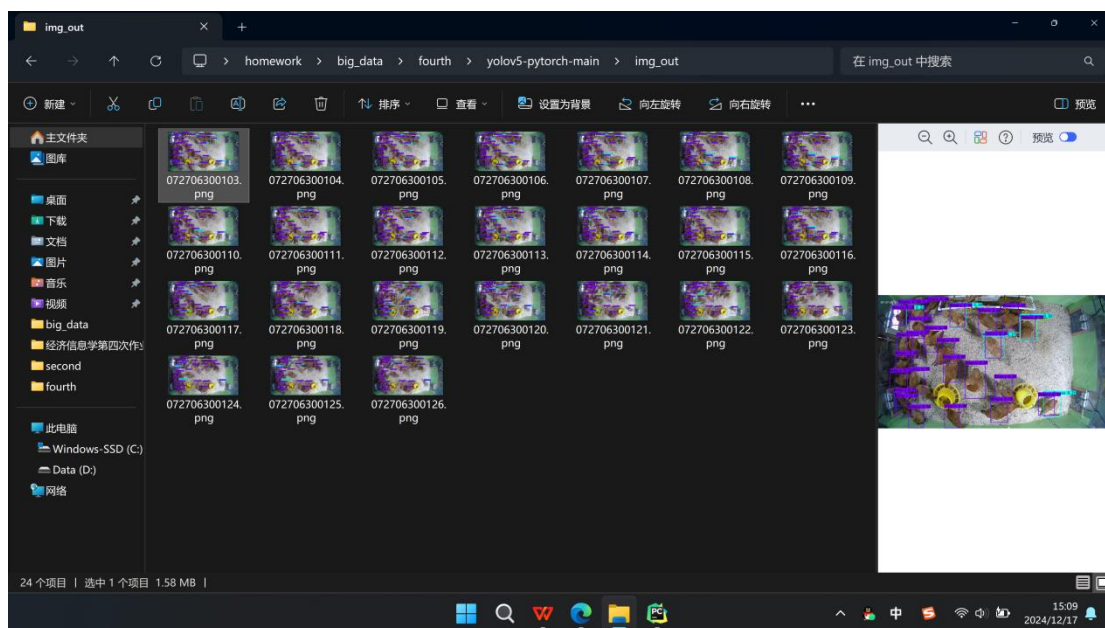
正在用 train.py 训练模型



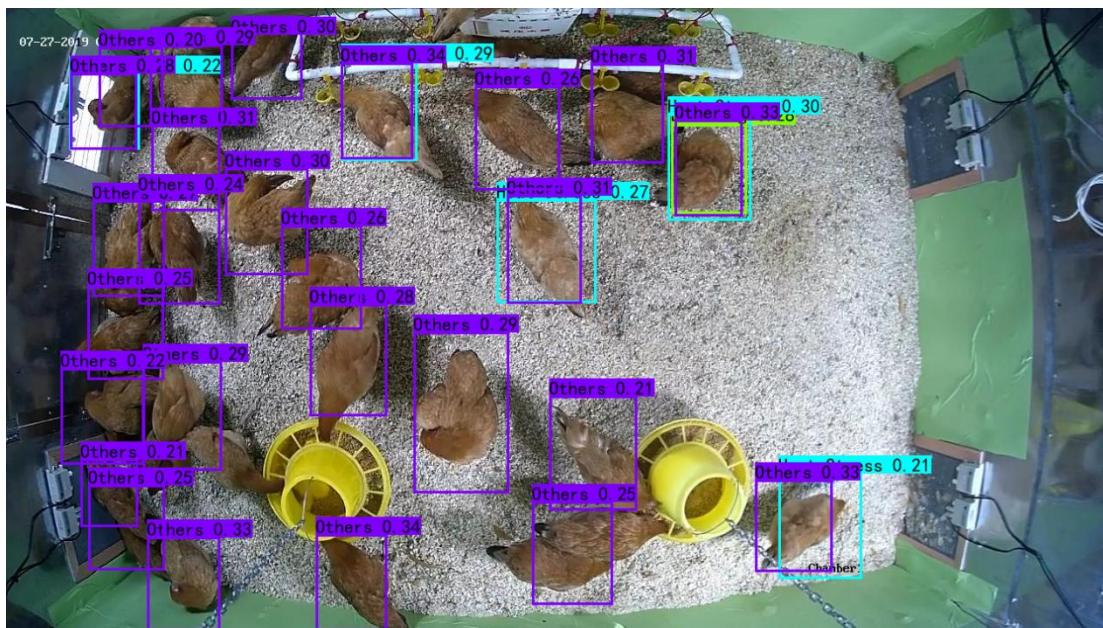
ep001-loss3.711-val_loss3.728.pth 为新训练的模型



正在运行 predict.py 生成标注的图片，右侧文件夹是正在生成的图片



predict.py 最终生成结果



以运行 072706300103.png 结果为例

五、实验总结

在本次基于 YOLOv5 的鸡热应激行为识别实验中，成功构建了图像采集系统，并利用该系统采集了肉鸡在不同状态下的图像数据。通过对图像数据的预处理和标注，可以得到一个包含啄食、饮水、热应激和其他行为类别的图像数据集。

在模型搭建阶段，选择了 PyTorch 作为深度学习框架，并采用了 YOLOv5 模型进行对象检测。通过对标注好的 json 格式数据集进行训练，可以得到一个能够识别鸡只不同行为类别的模型。训练过程中，根据模型精度的要求，不断调整训练参数，并重复训练模型，直到获得了一个最优的 epoch。

在结果输出阶段，运行 predict.py 文件，对测试集中的图像进行了行为识别，并得到了相应的识别结果。