中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生数学建模竞赛

学　　校	中国科学技术大学
参赛队号	20103580106
队员姓名	1.谢慧琴
	2.姜浩然
	3.王海

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生数学建模竞赛

题 目　基于卷积神经网络和坎尼算子的大雾演化分析

摘　　要:

在人工智能领域高速发展的大背景下，将计算机视觉应用到人们日常生产生活中显得越来越有必要。尤其是交通安全领域，本世纪初期，我国工业高速发展，交通事故频发，对我国经济、社会等多个方面造成了极大的消极影响。而近年来，随着人们交通安全意识的提升和科技的进步，我国交通的交通安全得到了进一步的保障。交通安全领域，能见度显得极为重要，目前我国主要采用激光能见度仪进行能见度检测，但其成本较高，并且不适用实际中多变的环境。所以，准确的运用深度学习算法对能见度进行精确检测显得越来越重要，本文主要旨在解决此问题。

在问题一中，本文充分对题目中已知数据，即机场 AMOS 观测数据，进行数据预处理，分析其随时间的变化趋势，并与能见度的变化趋势作比较，并计算出各气象因素变量间的相关系数，选取了气压、风速、平均温度、露点温度、湿度这五个气象学参数作为影响能见度的主要考虑参数。故建立具有 5 个解释变量的多元线性回归模型，定量所选气象观测指标和能见度数据之间的关系。本文将经过数据预处理之后的数据随机分为训练集和测试集，其中训练集占比 75%，测试集占 25%，利用测试集进行回归分析，最终推导出能见度与地面气象观测（气压、风速、温度、湿度）数据之间的关系公式。

在问题二中，本文建立了基于卷积神经网络的深度学习模型。首先，本文创建了能见度数据库，该数据库的来源主要是某机场视频数据（机场视频.zip）和能见度数据（机场 AMOS 观测.zip），利用机场视频每 25 帧提取一张图片，代表该时刻的能见度，将能见度真值作为图片标签建立对应关系。在将所有图片后按照每 100 米为一个分类区间长度进行分类，训练集与验证集的数量为接近 3:1；然后，将能见度训练集与测试集输入到 tensorflow 系统中进行能见度检测的训练，利用训练集与验证集的效果，通过不断调整训练的参数，反复迭代得到能见度检测效果更好的模型，利用网络选择最好的模型；最后，对测试集的图片进行能见度检测，验证训练好的模型的准确率及对该模型进行精度评估。

在问题三中，本文建立了基于坎尼算子和 bwarea 函数的能见度估算模型。首先，本文用高斯滤波器平滑题目中给定的 100 张高速公路截图；然后，用一阶偏导的有限差分来计算各图象梯度的幅值和方向，并对梯度幅值进行非极大值抑制；接下来，用双阈值算法检测和连接边缘，得出边缘图像；最后，本文利用 bwarea 函数计算二值边缘图像，即值为 1 的像素点组成的区域，求出其图像面积，利用自行推导的公式进一步计算出能见度数值，并将计算出来的能见度数值随时间的变化趋势绘图表现。

在问题四中，本文建立了基于时间序列的 ARIMA 预测模型。利用问题三模型计算出的能见度数值，进行平稳性检验和差分运算，然后进行模型拟合，并进行模型评估，最终成功地预测出了未来一段时间内能见度数值的变化趋势。

关键词：能见度、多元线性回归、卷积神经网络、坎尼算子、ARIMA

1 问题重述

在交通安全领域，能见度体现了十分重要的作用，尤其是近年来，环境恶化，一些地区出现了雾霾天，极大程度地影响了交通安全，所以说，发展一种高效、智能且准确的能见度识别方法尤为重要。

能见度，是指视力正常的人能将目标物从背景中识别出来的最大距离。是气象、公路行车、飞机飞行中常见指标，单位通常是米。影响能见度的因素主要是雾和霾。众所周知，能见度对高速公路行车安全非常重要，当能见度很低时，为了行车安全，高速公路管理者通常的做法是封路。而在航空领域，习惯用跑道能见度反映机场附近雾和霾的大小，其定义为在跑道的一端沿跑道方向能辨认出跑道或接近跑道的目标物（夜间为跑道边灯）的最大距离。一般情况下，当机场能见度只有 400 米左右时，会禁止航班起降。当机场能见度只有 600-800 米左右时航班虽然可以正常起降。但出于安全考虑，机场会采取临时控制航班流量的措施，拉大航班起飞间隔，容易造成航班延误。因此，能见度预测是高速公路管理部门和航空公司十分关注的问题。

激光能见度仪是常用的检测能见度的仪器。目前，我国高速路网已逐步形成，若大量使用激光能见度仪对全国高速路网进行全覆盖将耗资巨大，同时激光能见度仪还存在对团雾检测精度不高，探测的范围很小，维护成本高等不足。近年来，基于视频的路况（跑道）能见度检测方法受到人们的关注，它某种程度上克服了激光能见度仪的不足。视频能见度检测方法是将大气光学分析与图像处理及人工智能技术结合，通过对视频图像的分析处理，建立视频图像与真实场景之间的关系，再根据图像特征的变化，间接计算出能见度数值。但现有的基于视频图像的能见度检测方法，由于是间接计算，很难准确地估算能见度。特别地，这些方法中大多数只选取少量视频、截取图像中的某些固有特征[1,2]，基于 Koschmieder 定律[3,4]进行估计，并没有充分利用视频的连续信息，所以估计的精度不高，有较大的改进空间。

由于一般情况下，能见度究竟是 2000 米还是 3000 米对公路行车、飞机飞行几乎都没有影响，只是在恶劣天气，尤其是大雾情况下需要准确估计当前、特别是预测未来的能见度。所以本项目只关注大雾的演化规律。

事实上，大雾的形成和消散有其自身的规律，通常与近地层的气象因素有关。而视频资料包含了丰富的信息，特别是涵盖了大雾的变化过程信息。充分利用这些信息，不仅可以提高能见度估计精度，也可以对大雾的消散进行预测。

题目中出现的名词，对其进行解释：

1、 能见度，是指视力正常的人能将目标物从背景中识别出来的最大距离。所谓“能见”，在白天是指能看到和辨认出目标物的轮廓和形体，在夜间是指能清楚看到目标灯的发光点。

2、 雾：在水汽充足、微风及大气稳定的情况下，相对湿度达到 100%时，空气中的水汽便会凝结成细微的水滴悬浮于空中，使地面水平的能见度下降，这种天气现象称为雾。

3、 团雾：是受局部地区微气候环境的影响，在大雾中数十米到上百米的局部范围内，出现的更“浓”、能见度更低的雾。团雾外视线良好，团雾内一片朦胧。

4、 目标物和背景的亮度对比。在大气中目标物能见与否，取决于本身亮度，又与它同背景的亮度差异有关。比如，亮度暗的目标物在亮的背景衬托下，清晰可见。或者亮的目标物在暗的背景下，同样清晰可见。表示这种差异的指标是亮度的对比值 K。设 为目标物的固有亮度， 为背景的固有亮度，则亮度的对比值定义为：

$$K = \frac{|B_0' - B_0|}{B_0'}, \text{if } B_0' \geq B_0; \quad K = \frac{|B_0' - B_0|}{B_0}, \text{if } B_0' < B_0$$

5、 能见度测量基本方程：

$$F = F_0 e^{-\sigma z}$$

这里 F 和 F_0 分别表示观测和入射的光照强度，参数σ称为衰减系数，与雾的厚度有关：σ越大表明雾越浓。气象光学视程(MOR)

$$\mathrm{MOR} = \frac{\log(F/F_0)}{-\sigma} = \frac{\log(0.05)}{-\sigma}$$

6、 跑道视程(RVR)：是指在跑道中线上的航空器上的飞行员能看到跑道面上的标志或跑道边界灯或中线灯的距离。

为了估计不同大雾情况下对应的能见度以及预测大雾的消散，我们需解决如下问题：

1、 雾与近地面的气象因素有关。建立模型描述能见度与地面气象观测（温度、湿度和风速等）之间的关系，并针对题目所提供的数据（机场 AMOS 观测.zip）导出具体的关系式；

2、 根据题目提供的某机场视频数据（机场视频.zip）和能见度数据（机场 AMOS 观测.zip），建立基于视频数据的能见度估计深度学习模型，并对估计的能见度进行精度评估；

3、 高速公路某路段只有监控视频数据，建立不依赖能见度仪观测数据的能见度估计算法（提示：事实上，在有雾的情形可以估计视频中物体的景深[1]。反过来，理论上也可以利用视频中不同景深的物体，在不同能见度下的亮度差异估计能见度），讨论相关算法实现过程，并针对题目提供的一段视频（高速公路视频截图.zip）绘制该时间段这段高速公路能见度随时间变化曲线；

4、 利用问题三得到的能见度随时间变化规律，建立数学模型预测大雾变化趋势（加重或减弱）、何时散去（达到指定的能见度，比如 MOR=150m）？

2 问题假设

1、 假设高速公路监控摄像头符合国家标准：高度为 6.5m，且高速公路中间白线长 6 米，间隔 9 米，一个周期 15 米；

2、 假设在第四文中，大雾变化演变规律符合一般理想情况，不会出现极端天气明显影响大雾变化。

3 问题分析

2.1 问题一的分析

根据问题一的题目要求，我们需要根据已知数据，即题目附件中给的地面观测数据，包括：本站气压，飞机着陆地区最高点气压，修正海平面气压，温度，相对湿度，露点温度，灯光数据，2 分钟平均风速，2 分钟平均风向，2 分钟平均垂直风速。推导出能见度与地面观测数据之间的关系公式。根据查阅资料，我们发现以上数据并非全部都是影响能见度的重要参数，故需要对其进行合理化简化，考虑其中几个重要因素和能见度的具体关系。

2.2 问题二的分析

在此问题中，题目要求我们根据并基于机场视频数据和能见度数据，建立深度学习模型，对估计的能见度进行精度评估。本文考虑基于卷积神经网络模型（CNN），利用在多个卷积层和池化层之后的若干个全连接层，对结果进行分类识别，最后连接 Softmax 分类器，将网络识别结果输出，进行能见度检测。

2.3 问题三的分析

在此问题中，题目要求我们根据高速公路摄像头某时间段内拍摄下的 100 张视频截图，建立能见度估算模型，估算这 100 张视频截图的能见度数值。我们考虑使用基于深度学习

的图形处理方法，用坎尼算子对图像进行边缘化处理，计算处理后图像像素面积，进而表征能见度。

2.4 问题四的分析

在此问题中，题目要求我们根据问题三得出的高速公路能见度数据，对未来时间的能见度变化趋势进行预测，预测何时大雾能够散去。我们考虑建立差分自回归移动平均模型，充分利用现有数据和现有数据的变化规律，进行预测。

4 符号说明

符号	意义
Yi	能见度
f(x)	训练模型
G(x)	边缘函数
L	边缘定位精度
h(x)	滤波器的脉冲响应
W	滤波器的宽度
k	能见度与边缘轮廓线面积比例
k_i	自回归系数
λ_i	移动平均系数

5 问题一模型的建立与求解

5.1 数据的预处理

对题目中已给的地面观测数据进行预处理，观察其在固定时间内（2019 年 12 月 15 日）的变化规律，并与能见度在相同时间内的变化规律进行比较。

5.1.1 本站气压

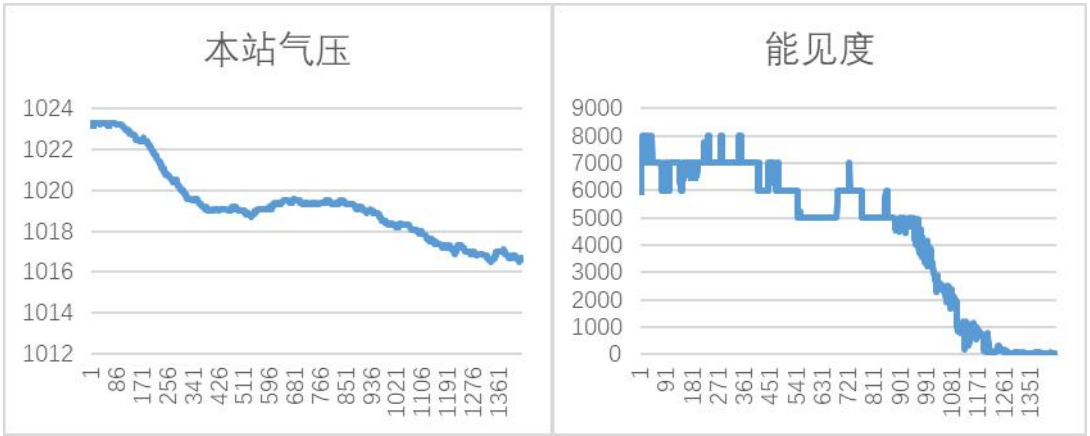

图 5-1 本站气压和能见度对比

5.1.2 飞机着陆地区最高点气压

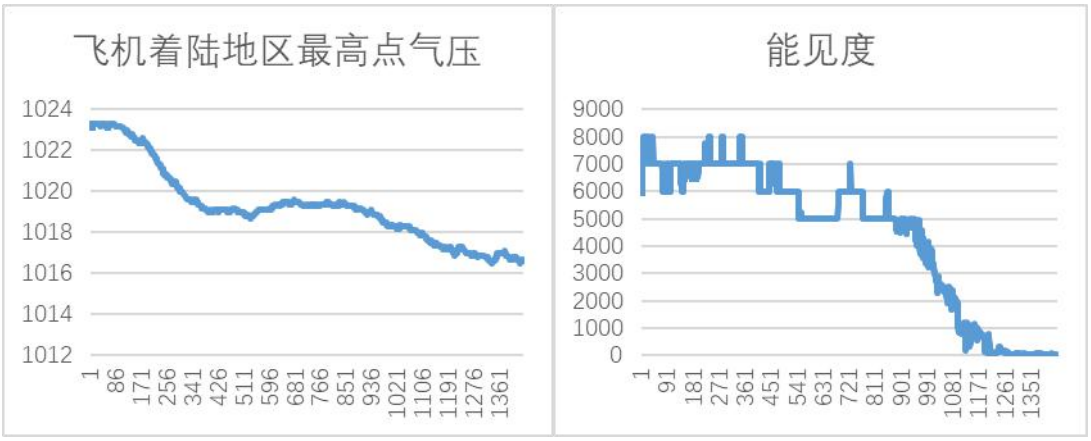

图 5-2 飞机着陆地区最高点气压和能见度对比

5.1.3 修正海平面气压

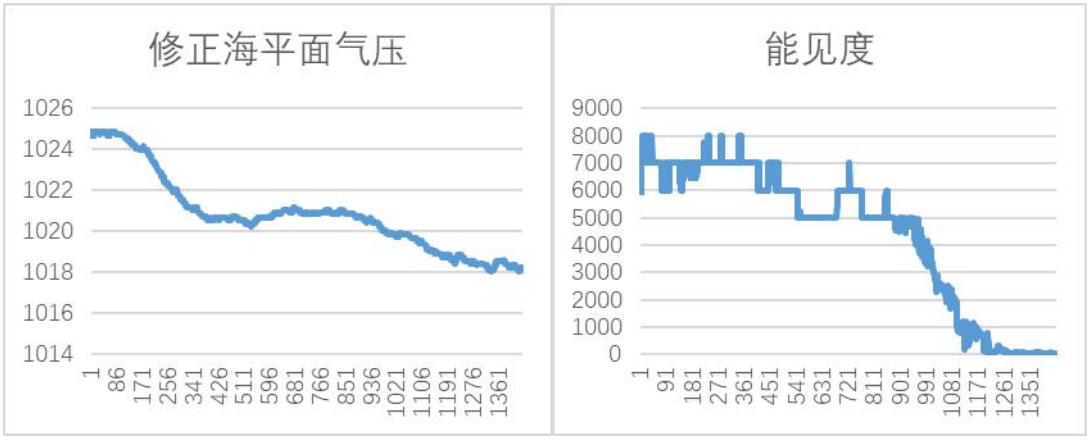

图 5-3 修正海平面气压和能见度对比

5.1.4 温度

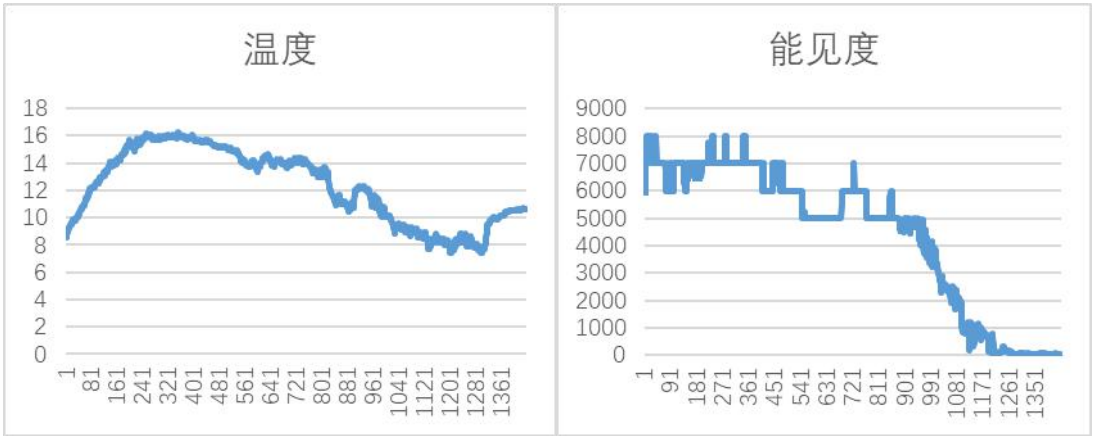

图 5-4 温度和能见度对比

5.1.5 相对湿度

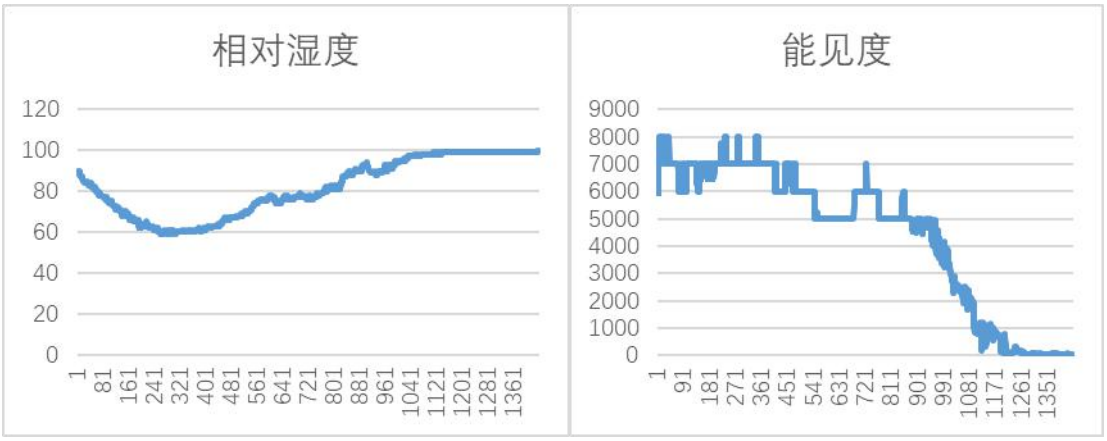

图 5-5 湿度和能见度对比

5.1.6 露点温度

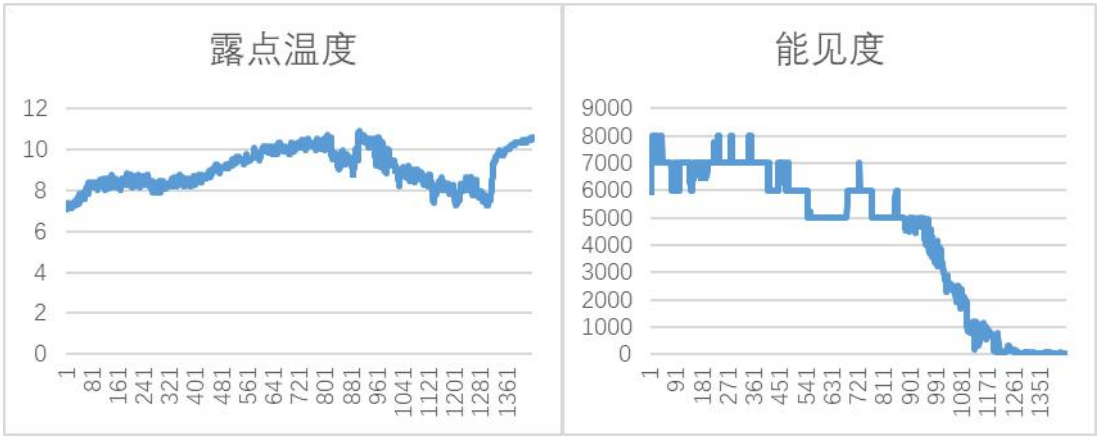

图 5-6 露点温度和能见度对比

5.1.7 灯光数据

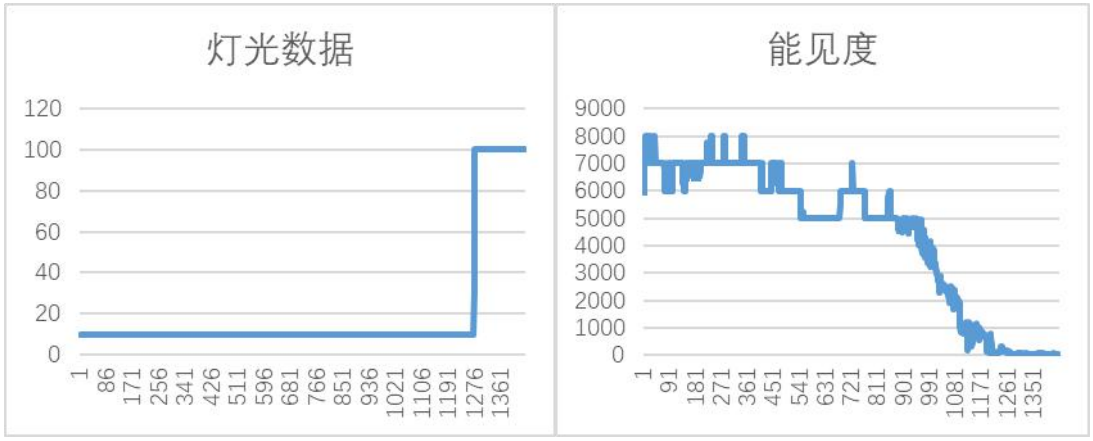

图 5-7 灯光数据和能见度对比

5.1.8 2 分钟平均风速

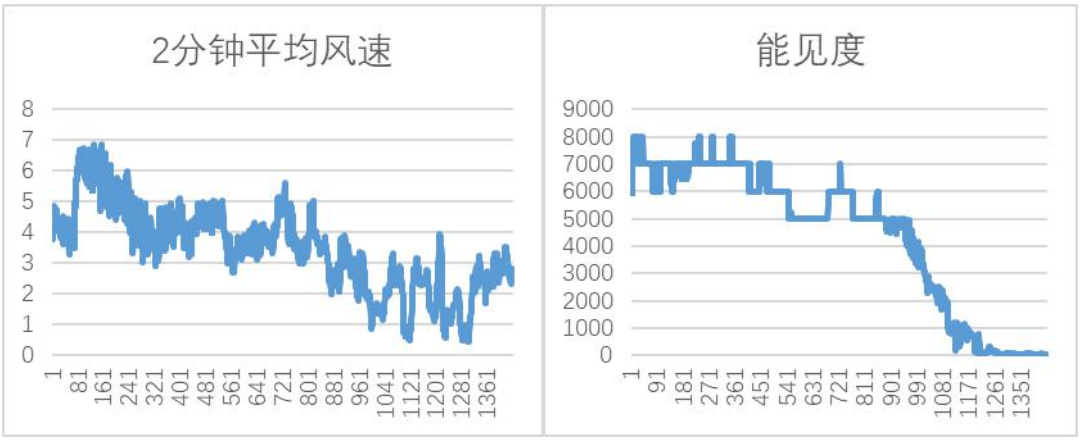

图 5-8 2 分钟平均风速和能见度对比

5.1.9 2 分钟平均风向

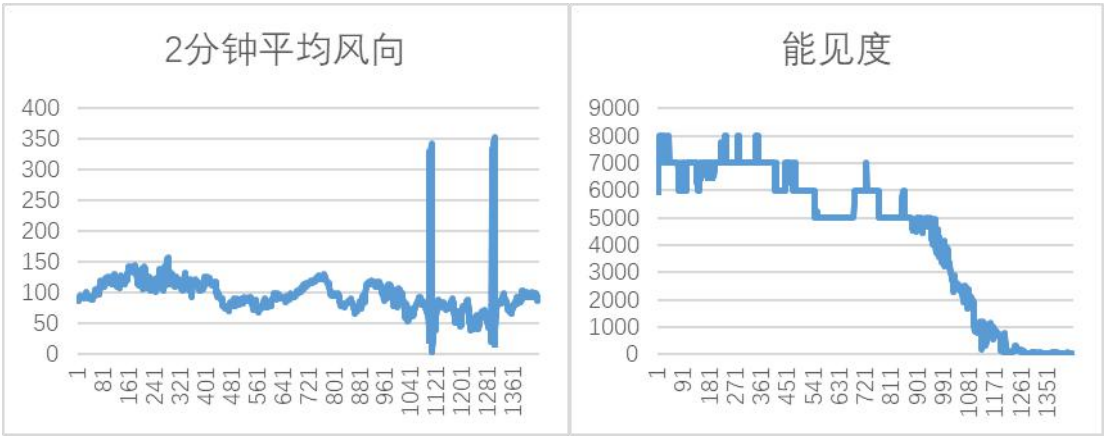

图 5-9 2 分钟平均风向和能见度对比

5.1.10 2 分钟平均垂直风速

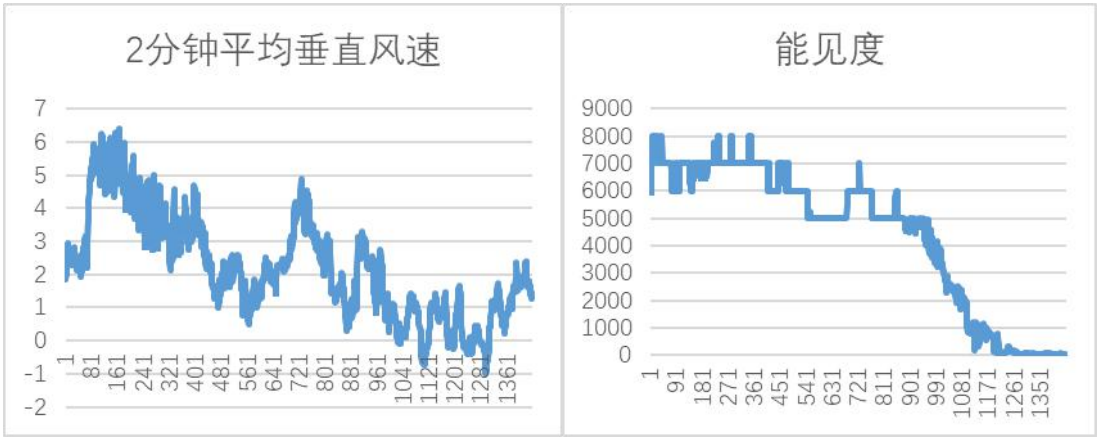

图 5-10 2 分钟平均垂直风速和能见度对比

根据观察以上与处理数据，我们可以直观地观察到，在选取数据的这一段时间段内，能见度的变化是逐渐降低。地面观测站观测的数据中，有本站气压，飞机着陆地区最高点气压，修正海平面气压，2 分钟平均风速和 2 分钟垂直风速在这一段选定时间内的变化规律是逐渐下降的，其他数据仍需要进一步处理。

5.2 模型建立

5.2.1 预处理

数据预处理之后，变量间相关系数如下热力图所示，各变量之间的分布情况如下图所示。我们将气候变量分为气压、风速、温度、湿度这四类，可以看出气压变量 PAINS、QFE、QNH 之间具有高度的相关性，结合三者的定义，选取 PAINS 作为气压的代表变量，风速相关变量 WS2A、CW2A 之间的相关系数高达 0.78，最终选取 WS2A 作为风速的代表变量。同时从表中可以发现，气压、风速、温度、湿度与响应变量能见度（VIS）之间具有较高的相关性，我们选取这四个数据作为模型的解释变量。

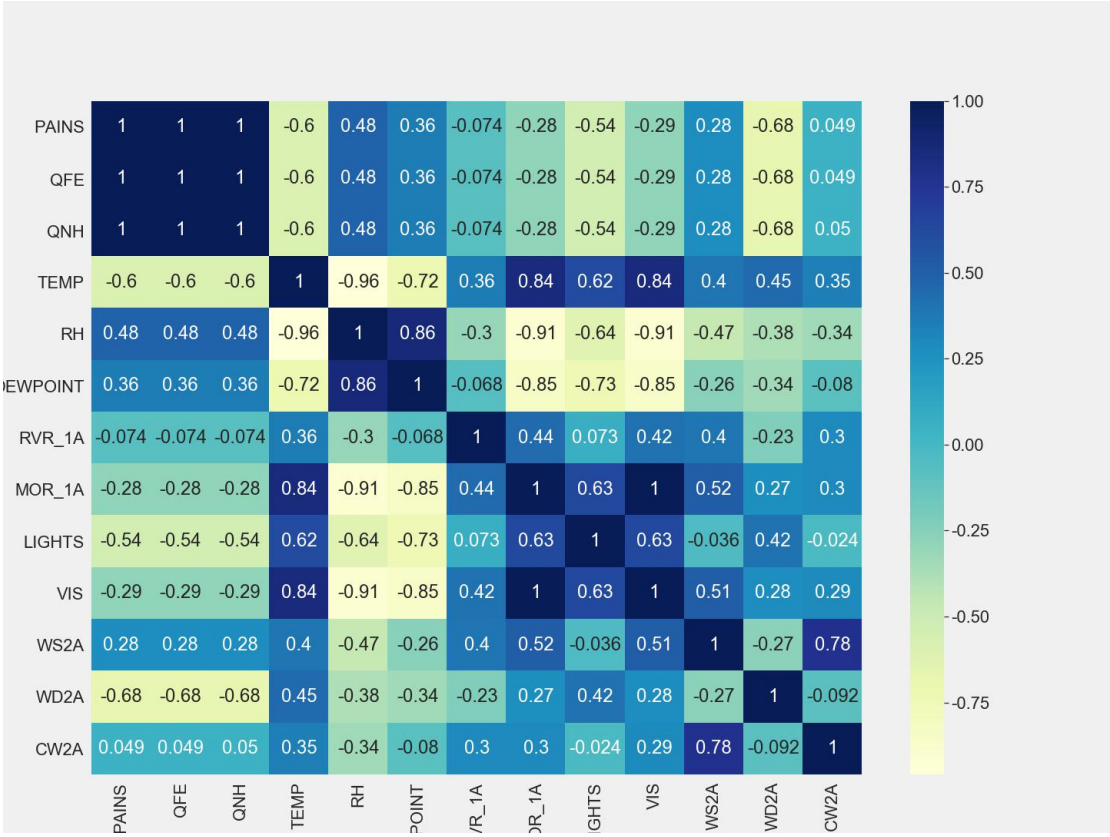

图 5-11 各变量关系热力图

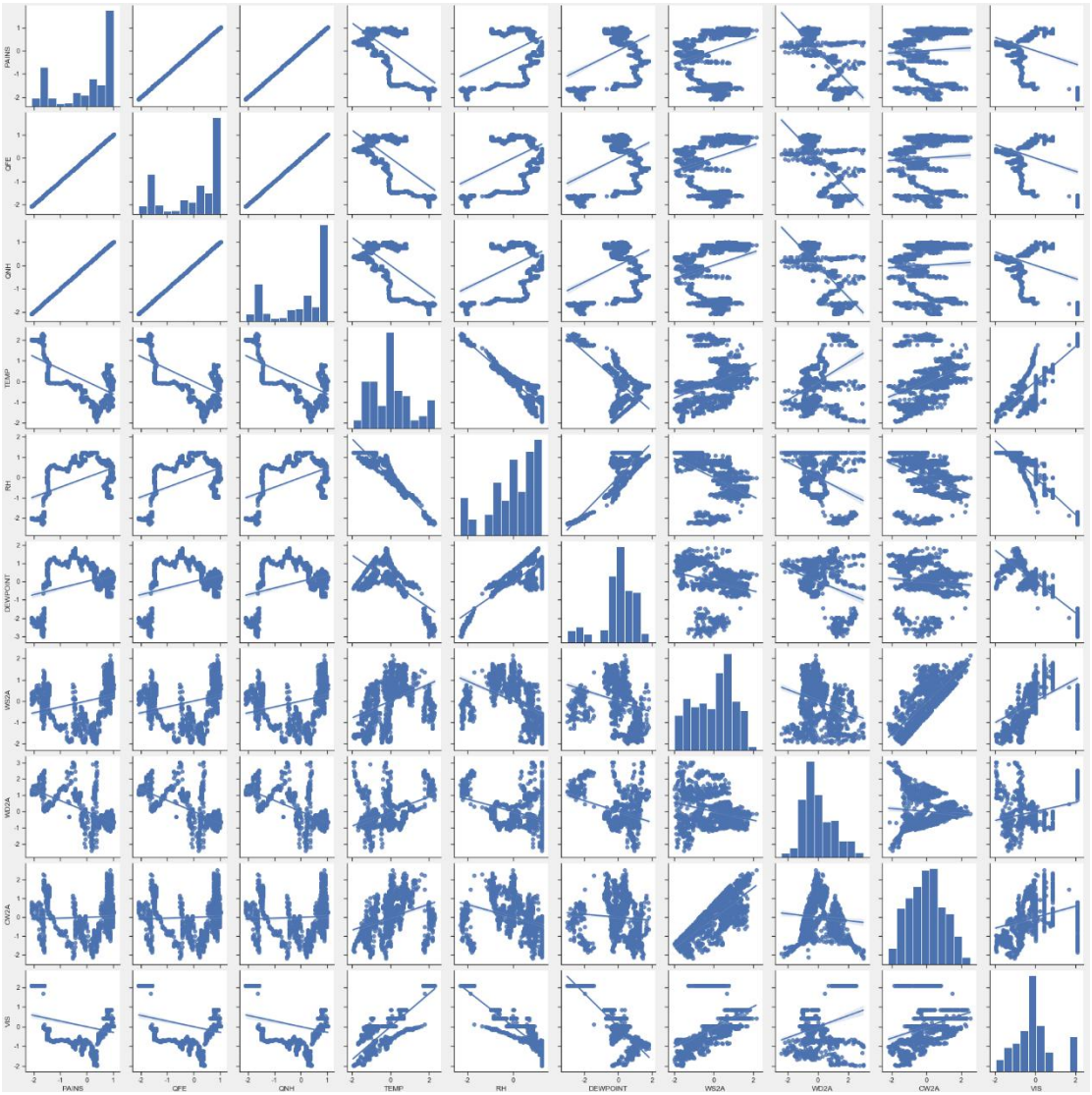

图 5-12 各变量关系 pairplot 图

5.2.2 多元线性回归模型

因为地面观测气象学数据的变化趋势与可见度变化趋势之间具有良好的线性相关性，故我们采用多元线性回归模型的思路[5]，定量分析气压、风速、温度、湿度这四个气象观测指标和能见度数据之间的关系。在此问题中，考虑到实际情况，机场能见度的影响因素极为繁多，为了尽可能实际的简化，我们采用了 4 个解释变量的多元回归模型：

$$Y_i=\beta_1 X_{1i}+\beta_2 X_{2i}+\beta_3 X_{3i}+\beta_4 X_{4i}+\beta+\mu_i$$

其中，Y_i 即能见度，单位为米；$\beta_i(i=1,2,3,4)$ 是偏回归系数；X_{1i} 表示本站气压；X_{2i} 表示温度；X_{3i} 表示相对湿度；X_{4i} 表示 2 分钟平均风速；β 是截距项。

表面上看，β 代表 X_{1i}、X_{2i}、X_{3i}、X_{4i} 均取 0 时的 Y 的均值，但这仅仅是一种机械的解释，实际上 β 是指所有未包含到模型中来的变量对 Y 的平均影响。μ_i 是随机干扰项。

本问题模型可以解释为：能见度主要受到本站气压、温度、相对湿度、2 分钟平均风速这 4 个变量的影响。

4 个解释变量的多元线性回归模型的 n 次观测数据，可以表示为：（我们中 n = 1322）

$$Y_1=\beta_0+\beta_1X_{11}+\beta_2X_{21}+\beta_3X_{31}+\beta_4X_{41}+\mu_1$$

$$Y_2=\beta_0+\beta_1X_{12}+\beta_2X_{22}+\beta_3X_{32}+\beta_4X_{42}+\mu_2$$

$$\cdots$$

$$Y_n=\beta_0+\beta_1X_{1n}+\beta_2X_{2n}+\beta_3X_{3n}+\beta_4X_{4n}+\mu_n$$

用矩阵可以表示为：

$$\begin{bmatrix}Y_1\\Y_2\\\vdots\\Y_n\end{bmatrix}=\begin{bmatrix}1&X_{11}&\cdots&X_{41}\\1&X_{12}&\cdots&X_{42}\\\vdots&\vdots&\ddots&\vdots\\1&X_{1n}&\cdots&X_{4n}\end{bmatrix}\begin{bmatrix}\beta_0\\\beta_1\\\vdots\\\beta_4\end{bmatrix}+\begin{bmatrix}\mu_1\\\mu_2\\\vdots\\\mu_n\end{bmatrix}$$

可以列出回归函数

$$Y=X\beta+\mu$$

我们将经过数据预处理之后的数据随机分为训练集和测试集，其中训练集占比 75%，测试集占比 25%，利用测试集进行回归分析[6]，获得一个包含解释变量的多元线性表达式，如下所示。分别对训练集和测试集进行检验，检验指标选取 R 平方（R^2）、均方误差(Mean Squared Error, MSE)、均方根误差(Root Mean Squared Error, RMSE)。

$$VIS=\beta_0+\beta_1\bullet PAINS+\beta_2\bullet TEMP+\beta_3\bullet RH+\beta_4\bullet WS2A$$

MSE（均方误差）函数一般用来检验模型的预测值和真实值之间的偏差[7-9]。其中，训练集为：

$$Train=\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n),...,(x_N,y_N)\}$$

式中，N 为训练样本总数，n=1,2,…,N。

测试集为：

$$Text=\{(x_1,y_1),(x_2,y_2),...,(x_m,y_m),...,(x_M,y_M)\}$$

式中，M 为训练样本总数，m=1,2,…,M。

训练模型：f(x)

预测值（估计值）：

$$\hat{y}=\{\hat{y}_1,\hat{y}_2,...,\hat{y}_m,...,\hat{y}_M\}$$

$$MSE:Mean\quad Squared\quad Error\begin{cases}\text{真实值与预测值（估计值）差平方的期望}\\\text{公式：}MSE=\dfrac{1}{M}\sum_{m=1}^{M}(y_m-\hat{y}_m)^2\\\text{分析：值越大，表明预测结果越差}\end{cases}$$

RMSE：Root Mean Squared Error（均方根误差），公式：

$$RMSE = \sqrt{MSE}$$

5.3 模型求解

5.3.1 计算数据选定

参照上文相关性分析，去除对最终能见度影响较小的气象学因素，我们选取了 PAINS（本站气压）、TEMP（温度）、RH（相对湿度）、WS2A（2 分钟平均风速）作为解释变量，解释变量与响应变量（即能见度）的分布情况如下图所示。

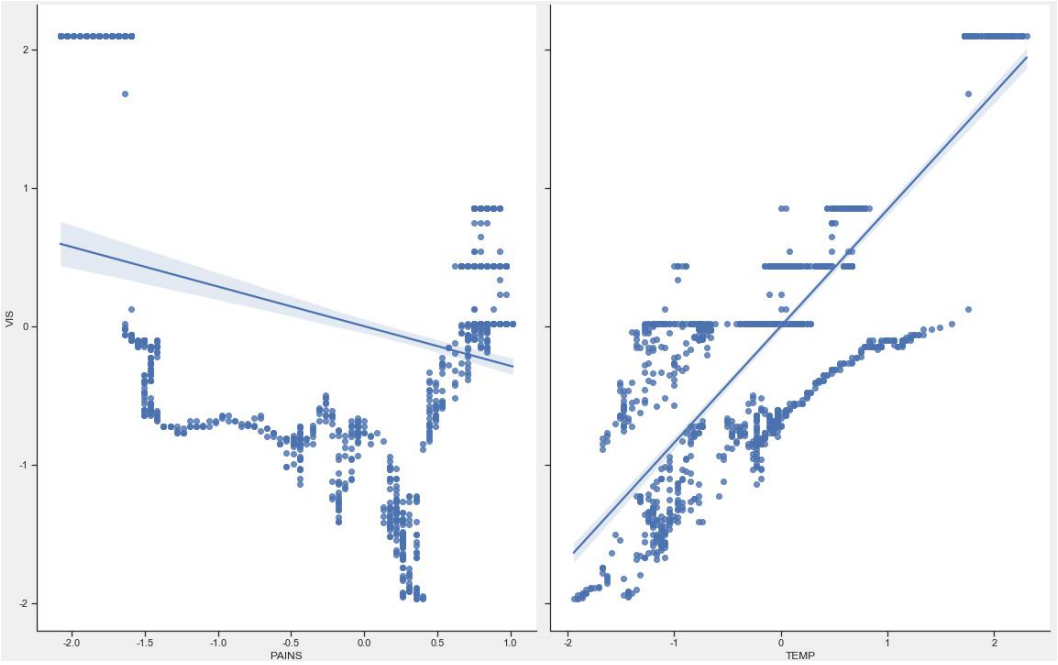

图 5-13 本站气压和温度与响应变量的对比

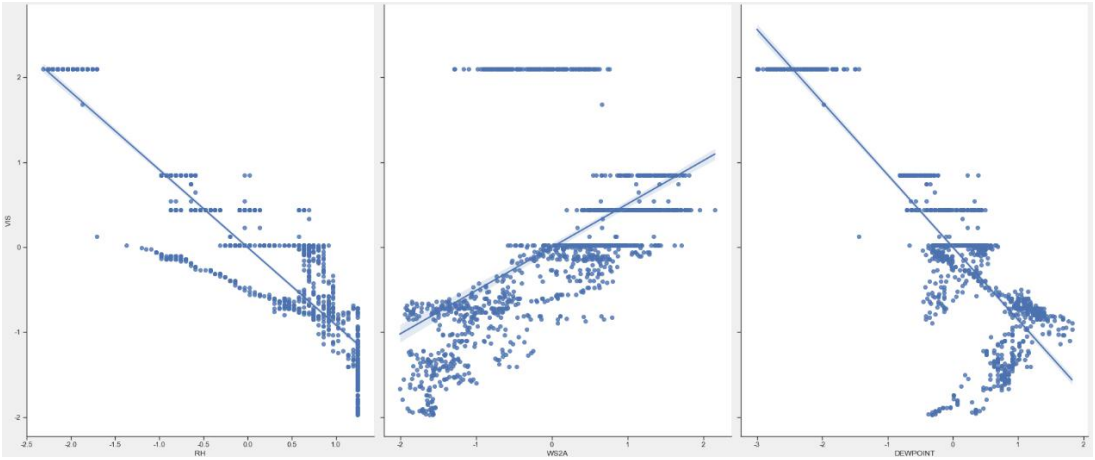

图 5-14 相对湿度和 2 分钟平均风速与响应变量的对比

然后，将训练接传入线性模型训练，获得回归模型，经过检验，计算得：

	R^2	MSE	RMSE
训练集	0.9031	0.0969	0.3114
测试集	0.8833	0.1160	0.3406

其中，总数据量：1322；训练集：991 （75%）；测试集：331 （25%）。为了更加直观观察，测试集真实值与预测值对比如下图：

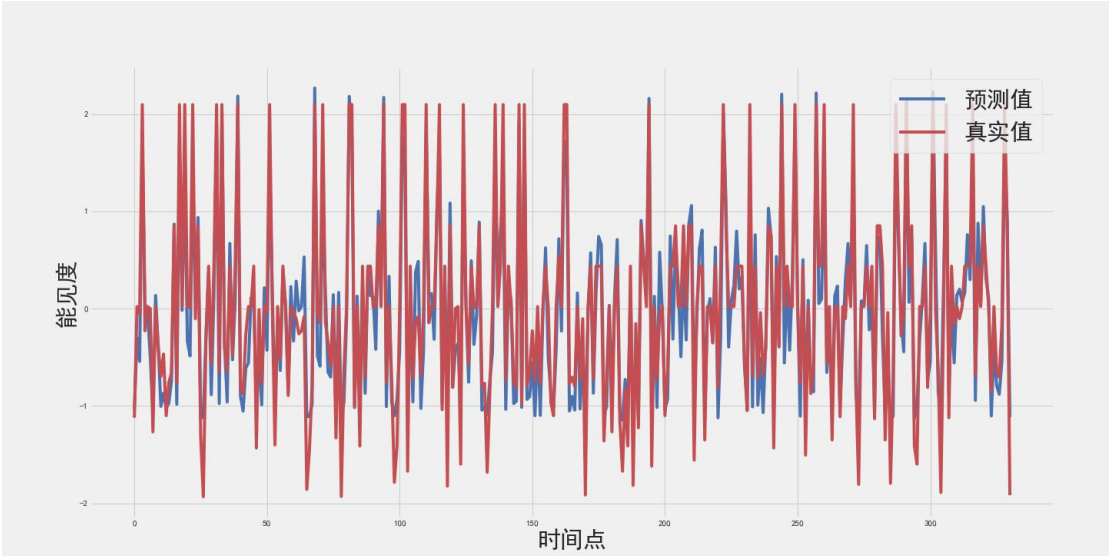

图 5-15 测试集真实值与预测值对比图

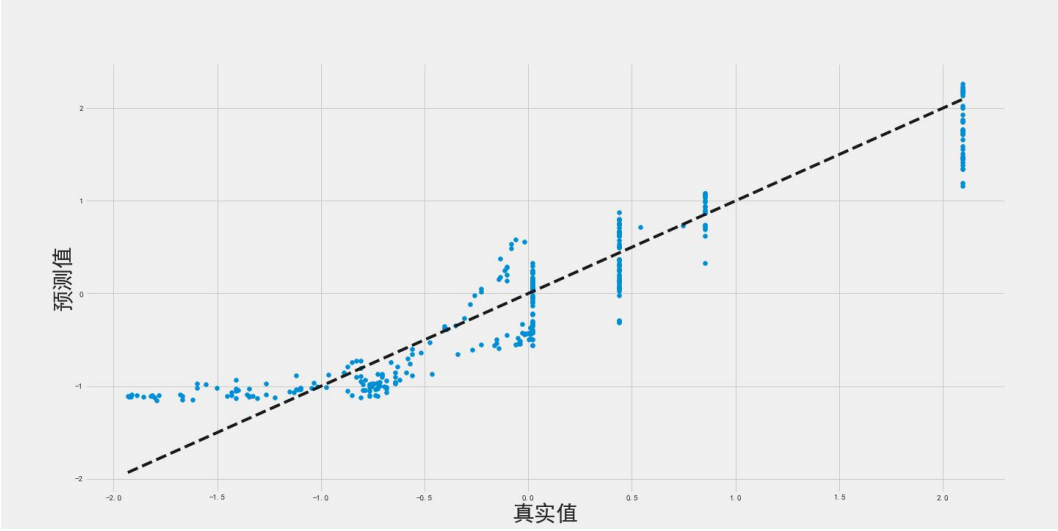

图 5-16 测试集真实值与预测值对比图

根据观察结果数据，R^2 接近 0.9，并且 MSE 和 RMSE 均在一个较小的范围内，因此拟合效果通过测试。最终 2019 年 12 月 15 日的观测数据与预测数据如下图所示：

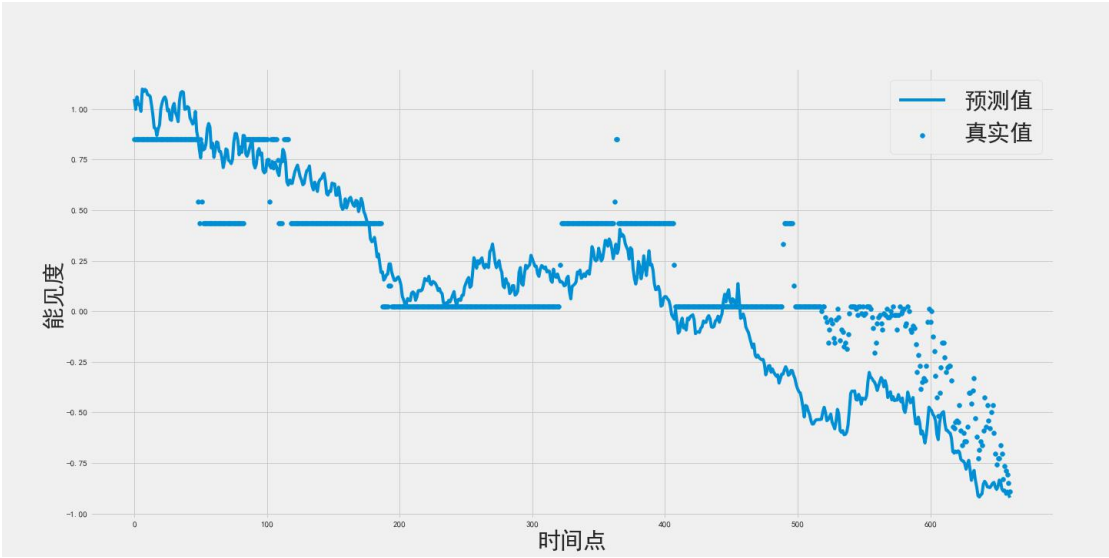

图 5-17 2019 年 12 月 15 日训练预测成果

2020 年 3 月 12 日的观测数据与预测数据如下图所示：

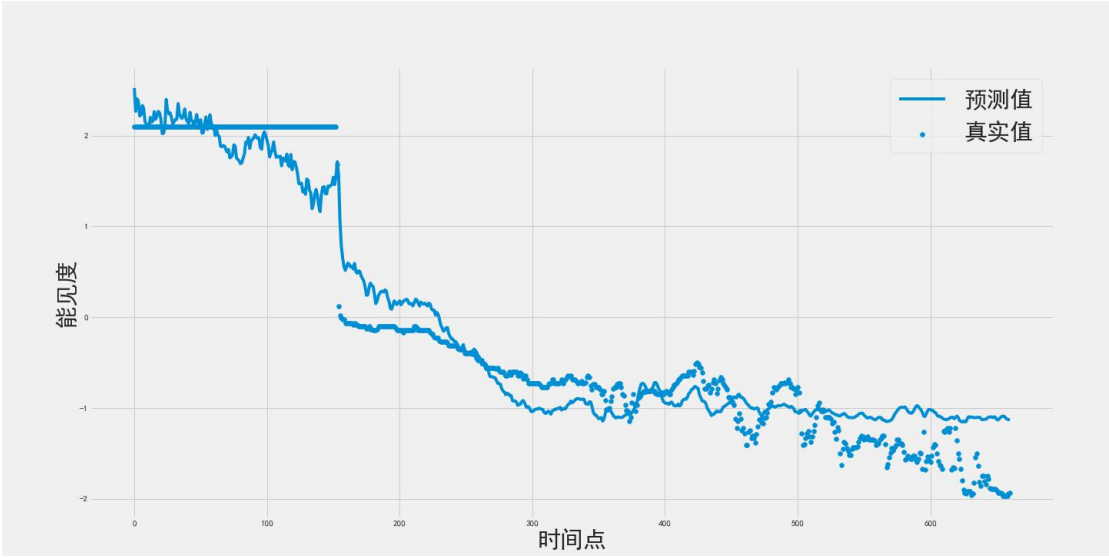

图 5-18 2020 年 3 月 12 日训练预测成果

经过训练上文建立的线性回归模型，得到如下的表达式，表示能见度与地面气象观测数据（本站气压、温度、相对湿度和 2 分钟平均风速）之间的关系：

$$VIS = (0.2506) * PAINS + (1.251) * TEMP + (0.9776) * RH + (0.1818) * WS2A + (-0.8302) * DEWPOINT + (-0.012869743000469608)$$

6 问题二模型的建立与求解

在问题二中，题目要求使用机场视频数据和能见度数据，建立基于视频数据的能见度估计深度学习模型，并对估计的能见度进行精度评估。我们考虑将深度学习的方法运用到能见度检测中。

6.1 深度学习模型的建立

卷积神经网络（Convolutional Neural Network，CNN）是一种由信号处理发展而来数字信号处理的方法，发展到图像信号处理上演变成一种专门用来处理具有矩阵特征的网络结构处理方式[10]。卷积神经网络最初设计是用来解决诸如图像识别之类的问题，但它目前的应用不仅仅只限于图像和视频，同时也用于时间序列信号，例如音频信号、文本数据等，卷积神经网络在很多应用上都有独特的优势[11]。

卷积神经网络是各种深度神经网络中最重要也是应用最广泛的一种，在许多计算机视觉问题上均表现出色，取得最好的效果。Le Cun 在 1989 年提出第一个真正意义上的卷积神经网络，用于对手写数字的识别，之后在它的基础进行改进，发展出各种深度卷积神经网络[12]。卷积神经网络主要由两部分组成，一部分是特征提取，另一部分是分类识别。在卷积神经网络中，可以将原始图像直接输入网络，不仅可以减少传统算法中图像特征提取和分类过程中数据重建的过程，还减少了计算参数降低了出错的概率，实现端到端的训练。

Alex Net 是第一个被广泛关注和使用的卷积神经网络模型，其在海量图像数据集上的优异表现揭示了卷积神经网络强大的学习能力和表示能力[13]。Alex Net 共包含 5 层卷积层、2 层全连接层和 1 层分类层，其模型结构如下图所示。

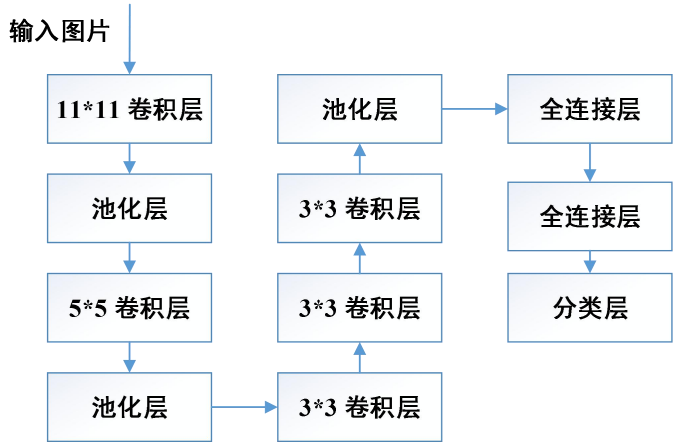

图 6-1 Alex Net 网络结构流程图

后续的卷积神经网络模型都是在 Alex Net 模型的基础之上加以改进的。目前常用的卷积神经网络分类模型有 VGGNet、Goog Le Net、Res Net、C3D、Squeeze Net 等[14]。另外，在卷进神经网络分类模型的基础上，通过利用神经网络提取图像中的通用特征，然后在根据所提取特征进行类别的分类与位置的回归，实现利用卷积神经网络进行检测，目前常用的检测模型有 RCNN、Fast RCNN、SPPNet 等。近年来，卷积神经网络模型在不断地向更宽更深的方向和不断调整网络结构方向发展。不管是分类网络模型还是检测网络模型，它们都是一种层次化的模型结构[15]，由不同种类的操作层如卷积层、池化层等堆叠而成，经过不同操作层的作用后原始的数据空间会映射为特征空间，从而完成分类或回归任务。

6.1.1 卷积神经网络

卷积神经网络是一种包含卷积计算且具有深度结构的前馈人工神经网络，一般包含输入层、卷积层、池化层、全连接层和输出层等。一般的卷积神经网络前几层由卷积层和池化层交替，逐层提取特征，靠近输出层为若干个全连接层，一个典型的卷积神经网络结构如下图所示。

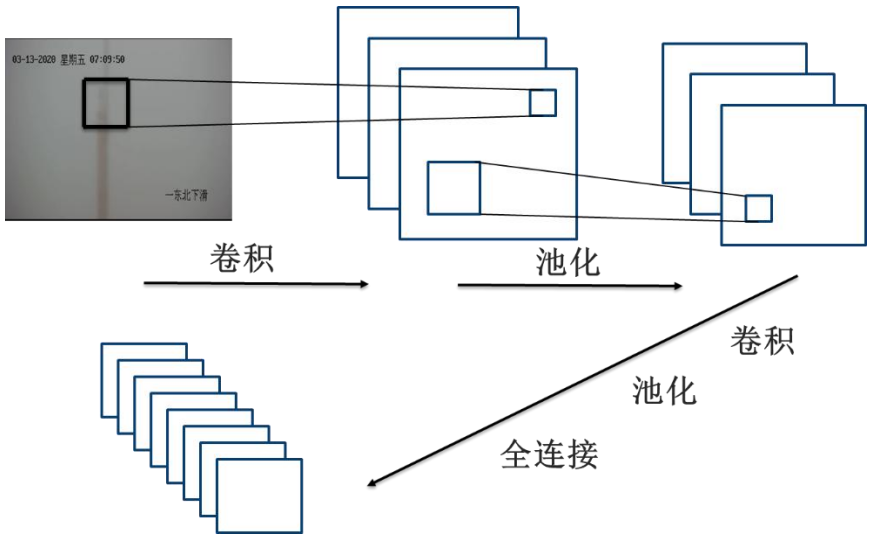

图 6-2 卷积神经网络结构流程图

上图是一个用于图像分类的 CNN 模型。我们将其改进以用于能见度精确评估。可以看到最左边的能见度图像为输入层，计算机将其理解为若干个矩阵，接着是卷积层和池化层的组合，根据模型的需要，这个组合在隐藏层可多次出现，上图出现了两次。当然在构建模型时，我们可根据实际情况的需要灵活使用卷积层加卷积层，或者是卷积层加卷积层加池化层的组合，以适用不同类型的数据库，但是最常见的卷积神经网络都是多个卷积层和

池化层的堆叠。在多个卷积层和池化层之后是若干个全连接层，当对结果进行分类识别，最后连接 Softmax 分类器，将网络识别结果输出[16]，我们是需要得到能见度值而不是进行能见度检测，所以本文在输入时将能见度值作为图片标签即特征输入进行训练。

1、数据输入层

数据在输入卷积神经网络之前往往要进行一系列的预处理操作，这些操作过程在输入层实现。数据预处理在构建网络模型时时十分重要，在很大程度上能够影响训练结果。对于不同的数据集，预处理方法存在一定的差异，常用的数据与处理方法有去均值和归一化。去均值：先计算出训练集图像像素均值，将训练集、验证集和测试集的每一张图片的每个像素值分别减去该均值，使得输入图片各个维度都中心化为 0。卷积神经网络提取图像信息不是提取绝对的像素值，而是来自像素之间的相对值。去均值操作仅仅是过滤了直流信息，并没有消除像素之间的相对差异。归一化：将所有图像像素值变换到同一范围内，具体方法为图像的每一个像素值除以像素值最大值，使所有像素值都在 0 到 1 之间。采用数据归一化后方便找到最优解，提高收敛效率。

2、卷积层

卷积层是进行特征提取的一层，利用卷积核对输入的图像进行卷积操作提取图像特征，并保留像素间的空间关系。对于同一个输入图像，不同的卷积核会生成不同的的特征图，所以在卷积层一般采用多个大小相同、权值不同的卷积核来提取多种图像特征。卷积操作其实就是通过滑动卷积核窗口遍历整张图像，将图像不同局部的像素矩阵和卷积核矩阵对应位置的元素相乘后相加的过程。卷积操作过程如下图所示。

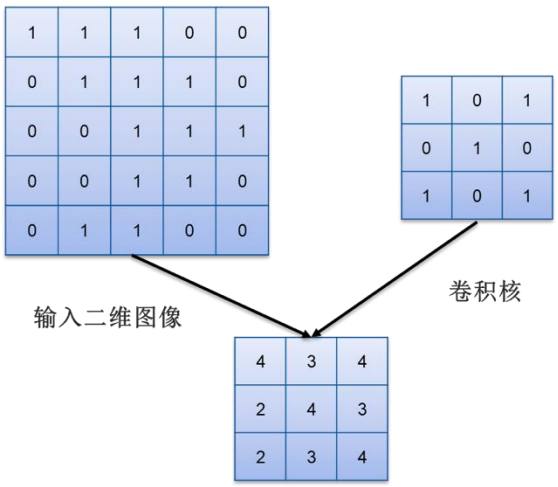

图 6-3 卷积神经网络操作流程图

卷积过后输出特征图的大小由卷积核大小、卷积步长和边缘填充大小三个参数控制。卷积步长表示完成一次卷积后，卷积核在输入图像两个方向上移动的像素个数。卷积步长越大，得到的特征图就越小，特征表示越稀疏。边缘填充通过在输入图像矩阵的边界进行零填充，来控制输出特征图的大小。假设输入图像尺寸为 m×n，卷积核大小为$k1\times k1$，两个方向的卷积步长分别为为$s1$ 和$s2$，图像边缘填充，那么经过卷积操作后输出特征图尺寸$m1\times n1$ 的计算公式如下式，特征图的个数等于卷积核的个数。

$$m_1 = \left\{\left[\frac{m-k_1}{s_1}\right]+1\right\}$$

$$n_1 = \left\{\left[\frac{m-k_1}{s_2}\right]+1\right\}$$

与传统神经网络不同，卷积神经网络通过局部感知和权值共享两种方法大大减少了模型参数个数，缩短了训练时间。在全连接网络中，隐藏层每个神经元要和图像的每一个像

素点相连，产生了大量参数，导致网络模型复杂，同时全连接网络忽视了输入的拓扑结构，无论输入是什么顺序，都不会影响训练结果，而图像具有很强的二维局部结构，其空间附近的像素高度相关。局部感知正是考虑到这一点，每个神经元只与图像的部分区域的像素点相连，通过在高层综合不同局部区域的神经元得到全局信息。为了进一步减少参数，卷积神经网络采用权值共享策略，对于图像的不同区域，共享连接参数，即卷积核参数，从而降低了模型复杂度。

3、池化层

在卷积神经网络中，卷积层之后通常会加一个池化层，用来降低特征图维数。在图像识别领域，输入图像尺寸通常比较大，导致卷积后特征图维数过高，模型参数量剧增。池化层通过对卷积层生成的特征图进行下采样操作，在保留显著特征的基础上对特征图进行压缩，降低特征向量的维数，加快计算速度。池化层只改变特征图的大小，不会改变特征图的个数。使用池化层不仅可以防止网络过拟合，还对图像轻微平移和旋转具有一定的不变性。常用的池化方法有最大值池化（Max Pooling）、平均值池化（Mean Pooling）、随机池化（Stochastic Pooling）。最大值池化指选取采样窗口内最大像素值作为输出值；平均值池化指计算采样窗口内所有像素点平均值作为输出值；随机池化指随机选择采样窗口内一个像素值作为输出值，通常像素值越大被选中概率也大。最大池化过程如下图所示，池化窗口为 2*2，窗口滑动步长为 2，不进行特征填充。

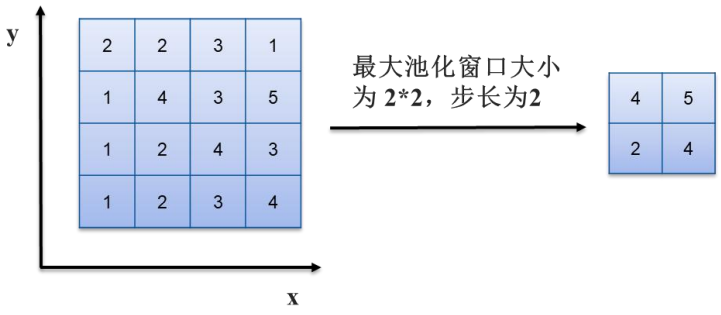

图 6-4 最大池化流程图

池化过后输出特征图尺寸的计算方法与卷积层类似，假设输入特征图尺寸为$m1\times n1$，采样窗口大小为$p1\times p1$，两个方向的滑动步长分别为为$s3$ 和$s4$，特征图边缘填充，那么经过池化操作后输出特征图$m2\times n2$ 的计算公式为：

$$m_2 = \left\{\left[\frac{m_1 - p_1}{s_3}\right] + 1\right\}$$

$$n_2 = \left\{\left[\frac{n_1 - p_1}{s_4}\right] + 1\right\}$$

4、全连接层

全连接层将经过多个卷积层和池化层的图像特征图进行整合，获取图像特征的高层含义，之后用于图像分类。全连接层的每神经元与其前一层的所有神经元连接，把卷积层或池化层输出的二维特征图转映射成一个固定长度的一维特征向量，这个特征向量包含了输入图像所有特征的组合信息，相当于在分类层之前进行了特征降维，减少了分类层的计算量。由于其全连接特性，全连接层参数是最多的，整个卷积神经网络大部分参数都来自于全连接层，因此实验中要合理控制其神经元数目，减少网络参数。

全连接层可以通过卷积层来实现，其本质原理是一样的，通过对输入进行加权求和得到输出，只是卷积层和输入之间采用局部连接，全连接层与输入之间是全连接，因此可以通过全局卷积操作实现全连接功能。若全连接层的上一层为卷积层，且该卷积层输出特征图大小为 h*w，则全连接层可以用卷积核大小为 h*w 的卷积层来代替，卷积核的个数等于

全连接层的神经元个数；若全连接层的上一层仍为与其结构相同的全连接层，则可以用卷积核大小为 1*1 的卷积层来代替，卷积核的个数等于全连接层的神经元个数。在早期的卷积神经网络中，大多都存在全连接层，对于全连接层参数过多产生的过拟合问题，一般采用 Dropout 技术来解决。最近一些网络模型采用全局均值池化来代替全连接层，取得了不错的效果。

5、分类层

分类层完成对输入图像的分类识别，多分类任务中一般采用 Softmax 回归分类器，此时该层也被成为 Softmax 层。Softmax 属于多类分类器，输入为样本特征，输出为样本属于各个类别的概率。输出类别有 n 类时，Softmax 会输出一个 n 维列向量，每一维列向量取值介于 0 到 1 之间，代表图像属于该类别的概率，最大概率值对应的类别即为输入图像所属类别。卷积神经网络进行训练时，网络模型的参数初始化至关重要，会直接影响训练结果。良好的初始化参数能够加速网络的收敛速度，更容易找到最优解；而较差的初始化参数可能导致极端的梯度问题，不仅降低网络训练速度，还会影响模型的识别效果。如果权重被初始化为较小的值，那么到达最后的信号值也比较小；相反，权重被初始化为较大的值，到达最后的信号值也会较大。不合适的参数初始化会导致网络无法进行参数更新，例如卷积神经网络中经常采用 Sigmoid 函数作为激活函数，通过观察 Sigmoid 函数曲线发现当输入为很大或者很小时，神经元的梯度接近 0，激活值则接近 0 者 1，神经元在此时达到饱和状态。这会导致反向传播过程中梯度迅速衰减，在深度神经网络中，传递到前面层的梯度极小甚至消失。此时的梯度信息对模型优化算法没有提供任何有效信息，导致前面网络层的参数无法进行更新，从而无法进行深层网络的训练。所以在进行参数初始化时需满足两个必要条件：保证各层激活值即不能为 0 也不能达到饱和状态。

最简单的参数初始化方法是将网络参数全部初始化为 0，参数变化完全依靠反向传播算法进行更新。但是这种初始化方法存在一个问题，在正向传播过程中，同一层神经元进行相同的运算得到一样的值，经过反向传播计算得到的梯度值也相同，最终同一层的神经元得到相同参数。在同一层设置不同神经元的目的希望学习到数据不同的特征，区分特征之间的重要程度。如果神经元参数相同，所有特征的权重都是一样的，网络对特征之间没有区分度，从而失去特征学习的意义。

为了避免上述情况，可以为不同特征配与不同的权重，从而使得网络对特征具有区分度。

常用的初始化方法有高斯分布初始化（Gaussian）、Xavier 初始化和 MSRA 初始化，将参数初始化为很接近 0 的值[18]。高斯分布初始化将权重随机初始化为均值为 0，方差为固定值的高斯分布，如均值为 0，方差为 0.01 的高斯分布。这种初始化方法只适用于网络层数较少的神经网络，对于深度神经网络，后面层的激活函数输出值趋向于 0，导致反向传播过程中的梯度值小，容易发生梯度弥散现象。为了避免这种现象，Glorot 等人提出了 Xavier 初始化，其基本思想是保持每层输入与输出具有相同的方差，信号在多层神经网络中传递后依然保持在合理

的范围内，从而使得信号可以在深度神经网络中传递，有两种常见的初始化方法：Xavier 均匀分布初始化和 Xavier 正态分布初始化。Xavier 初始化的前提条件是激活函数是线性的或者近似线性，而卷积神经网络常用的 Relu 激活函数显然不满足这一点。为此，何恺明提出了针对 Relu 激活函数的 MSRA 初始化，将权重初始化为均值为 0，方差为 n/2 的高斯分布，其中 n 为输入神经元个数，考虑了 Relu 激活函数对输出数据分布的影响，使输入和输出方差保持一致。

另外，预训练法也是一种简单有效的参数初始化方法。预训练法将预训练的模型参数作为新任务的初始化参数，在此基础上进行参数微调，以适应特定新任务，也就是迁移学习。要训练一个深度卷积神经网络，需要有足够多的带标注数据，同时训练时间特别长，

一般需要几天甚至几周的时间。对于小样本集来说，数据量远远不够，导致参数较难学习。但是通过迁移学习，无论网络结构多复杂，都可以应用于小样本集的任务。迁移学习时，通常有一个在大型数据集上训练好的网络模型文件，里面保存所有的网络预训练参数，迁移学习将网络模型对应层的参数加载进来后，用新的数据库训练网络进行参数微调。网络偏置项通常初始化为 0。对于采用 Relu 激活函数的神经网络，偏置项也可被始化方法为一个较小的常数，但并没有结果显示该方法否能促进网络收敛，所以在实际网络中，一般将偏置项设置为 0。

6.1.2 基于 CNN 的能见度分类模型

本文使用 CNN 进行模型训练，下面详细讲解利 CNN 进行能见度检测的过程，如图 所示为基于 CNN 的雾霾能见度检测流程。

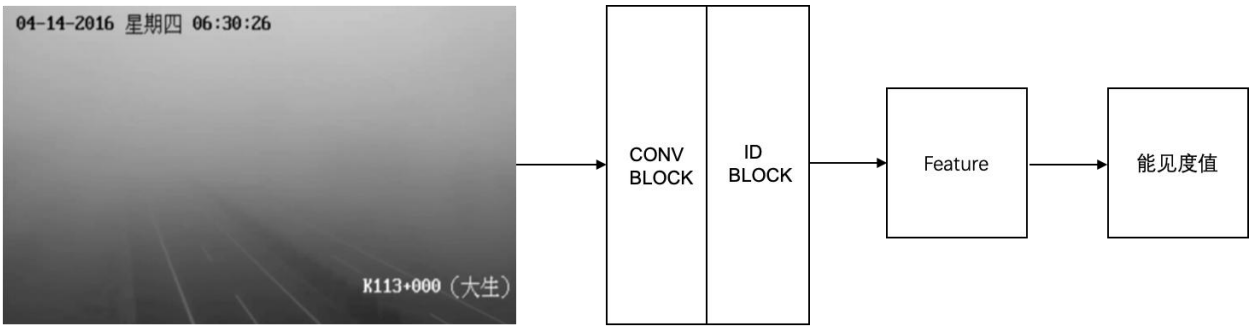

图 6-5 基于 CNN 的能见度检测流程图

1、模型网络结构

该网络结构共包含两个卷积层（Conv2D）、两个池化层（MaxPooling2D）、之后吧第二个池化层的输出扁平化为 1 维度（Flatten），之后就是隔着一个 Dropout 的两个全连接层（Dense）。模型具体参数见附录代码。

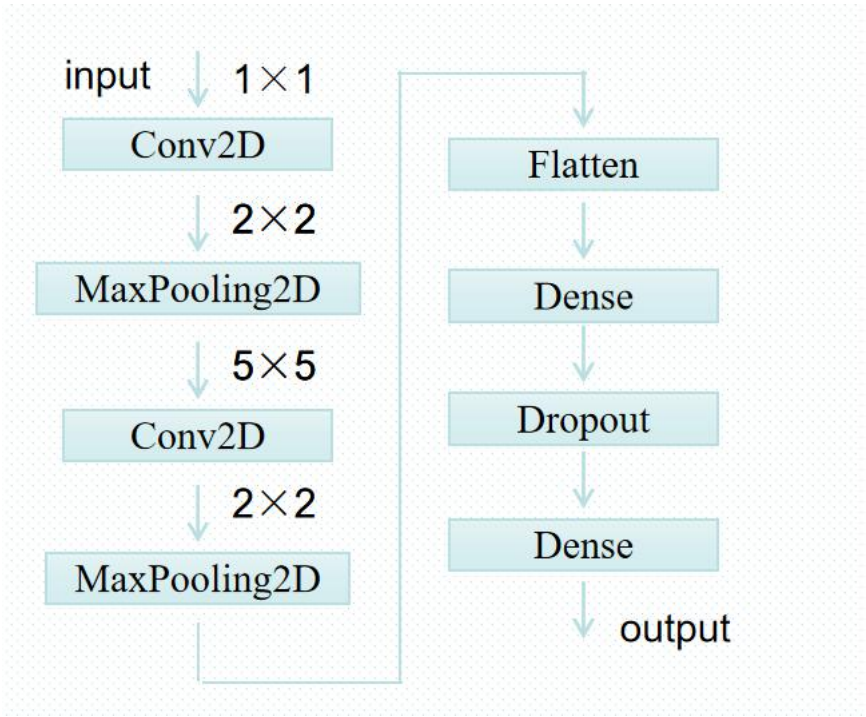

图 6-6 CNN 网络模型结构

2、建模步骤

本文选择使用的是 tensorflow 深度学习平台，基于深度残差网络的能见度检测方法包括

以下三个步骤：

步骤 1、创建能见度数据库，该数据库的来源主要是某机场视频数据（机场视频.zip）和能见度数据（机场 AMOS 观测.zip），利用机场视频每 25 帧提取一张图片，代表该秒的能见度，将能见度真值作为图片标签建立对应关系。在将所有图片后按照每 50 米为一个分类区间长度进行分类，训练集与验证集的数量为接近 3:1。

步骤 2、将步骤 1 中做好的两个能见度的训练集与测试集输入到 tensorflow 系统中进行能见度检测的训练，利用训练集与验证集的效果，通过不断调整训练的参数，反复迭代得到能见度检测效果更好的模型，利用网络选择最好的模型。

步骤 3、利用步骤 2 中得到的能见度检测模型，对测试集的图片进行能见度检测，验证训练好的模型的准确率及对该模型进行精度评估。

6.2 模型求解

本文基于 Res Net-50 深度学习模型，利用实际采集的机场视频数据和能见度数据，训练出一个能见度分类预测模型。

6.2.1 视频预处理

将机场视频每 25 帧取一张截图，即每一秒取一张机场视频中的截图，将所有截图批量转成灰度图片，并且去掉每张图片中的文字部分，以帧数序号命名每张图片，如第一张图片为 25.jpg，第二张图片为 50.jpg，以此类推。

图 6-7 图像处理流程

6.2.2 处理数据集

根据视频帧数与时间的对应关系，建立公式，处理视频中缺帧现象，最后将图片名称与世界时间严格对应，取 VIS1A，即 1 分钟的平均能见度作为响应变量。

表 6-1 图像处理数据

LOCALDATE	VIS1A	jpg_name
2020-03-13 00:01:00	2625.00	2800.jpg
2020-03-13 00:02:00	3000.00	3175.jpg
2020-03-13 00:03:00	3000.00	3275.jpg
…	…	
2020-03-13 07:58:00	150.0	695575.jpg
2020-03-13 07:59:00	150.0	697075.jpg

能见度每相差 50 米（m）分为一类，共分为 65 类，将同一类别的图片放在同一文件

夹下,并将文件夹以类别名命名。将所有数据随机分为训练集（75%）和测试集（25%），其中训练数据共 349 张，测试数据共 116 张，共 465 张图片。

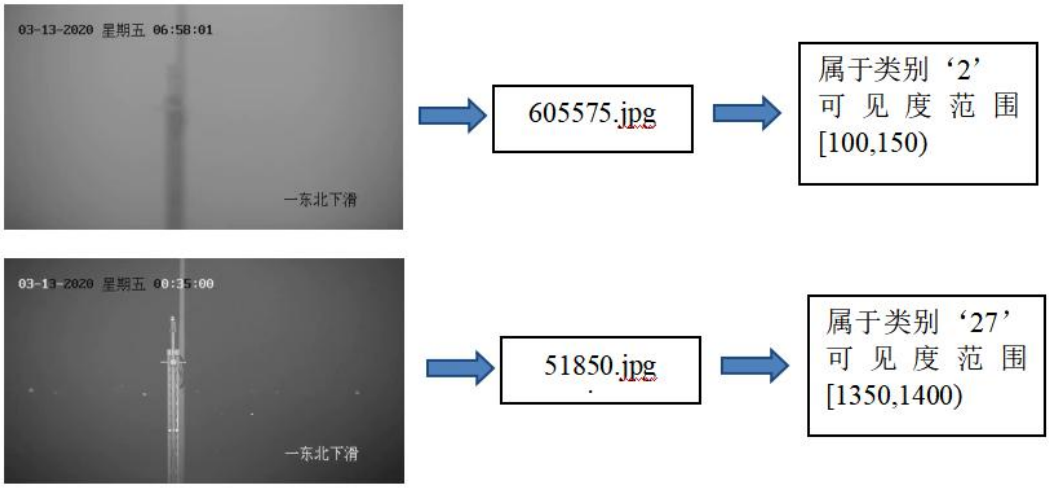

图 6-8 图像处理步骤

6.2.3 精度评估

1、数据集 Accuracy 和 Loss

表 6-2 精度处理结果

	Accuracy	Loss
训练集	0.8175	0.6995
测试集	0.8000	0.7061

$accuracy$：分类模型所有判断正确的结果占总观测值的比重

$$accuracy = \frac{TP + TN}{TP + TF + NP + NF}$$

2、混淆矩阵

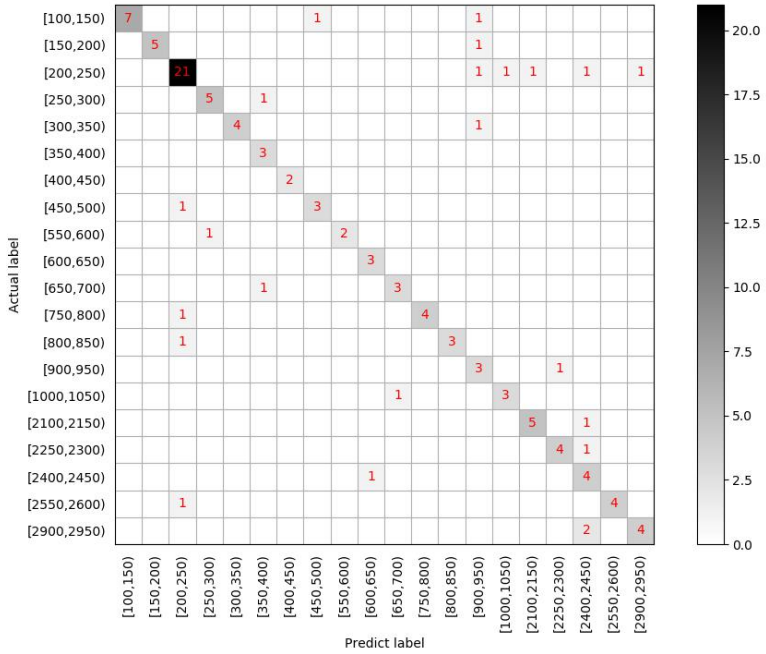

图 6-9 混淆矩阵

3、评估指标

为了评估训练模型的性能，本文选用如下评估指标：

FP：负样本被分类为正样本的数量

FN：正样本被分类为负样本的数量

TP：正样本分类正确的样本数

TN：负样本分类正确的样本数

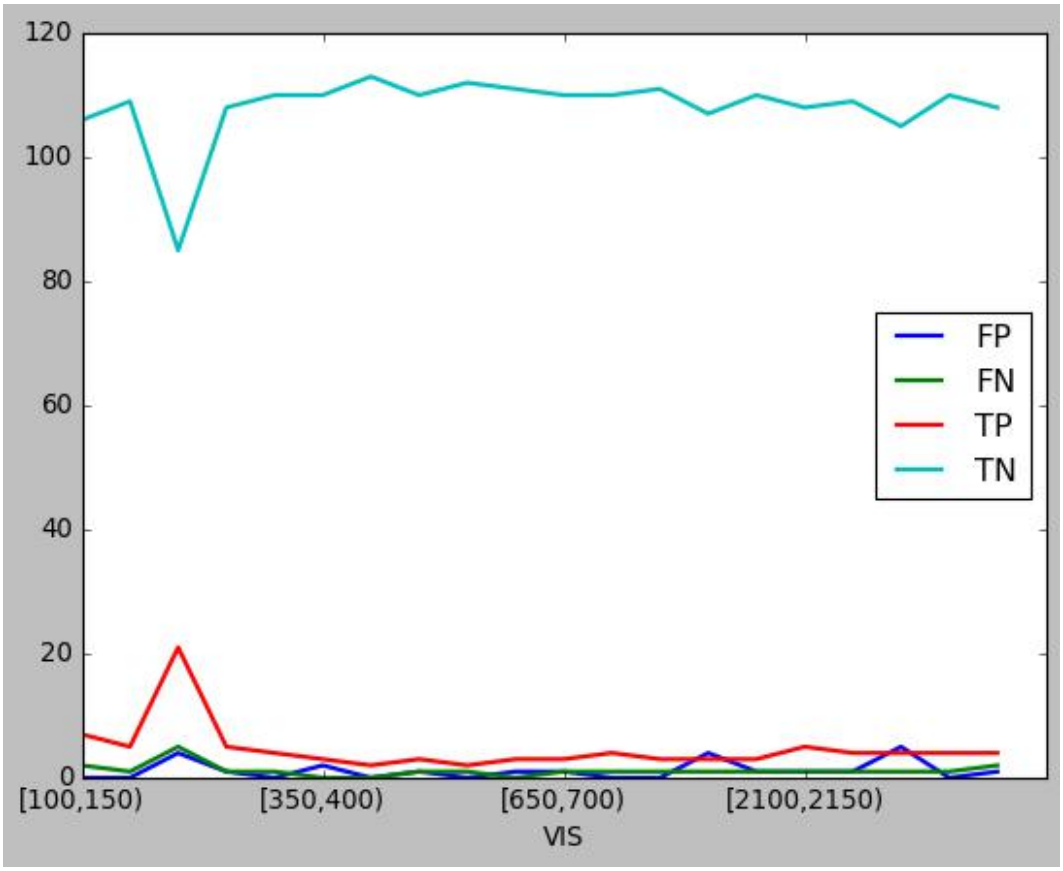

图 6-10 FP、FN、TP、TN 数量曲线

TPR（True Positive Rate）可以理解为所有正类中，有多少被预测成正类（正类预测正确），即召回率，给出定义如下：

$$TPR = \frac{TP}{TP + FN}$$

TNR（True Negative Rate）可以理解为所有反类中，有多少被预测成反类（反类预测正确），给出定义如下：

$$TNR = \frac{TN}{TN + FP}$$

阳性预测值 (PPV),又称 precision:

$$PPV = \frac{TP}{TP + FP}$$

阴性预测值 (NPV):

$$NPV = \frac{TN}{TN + FN}$$

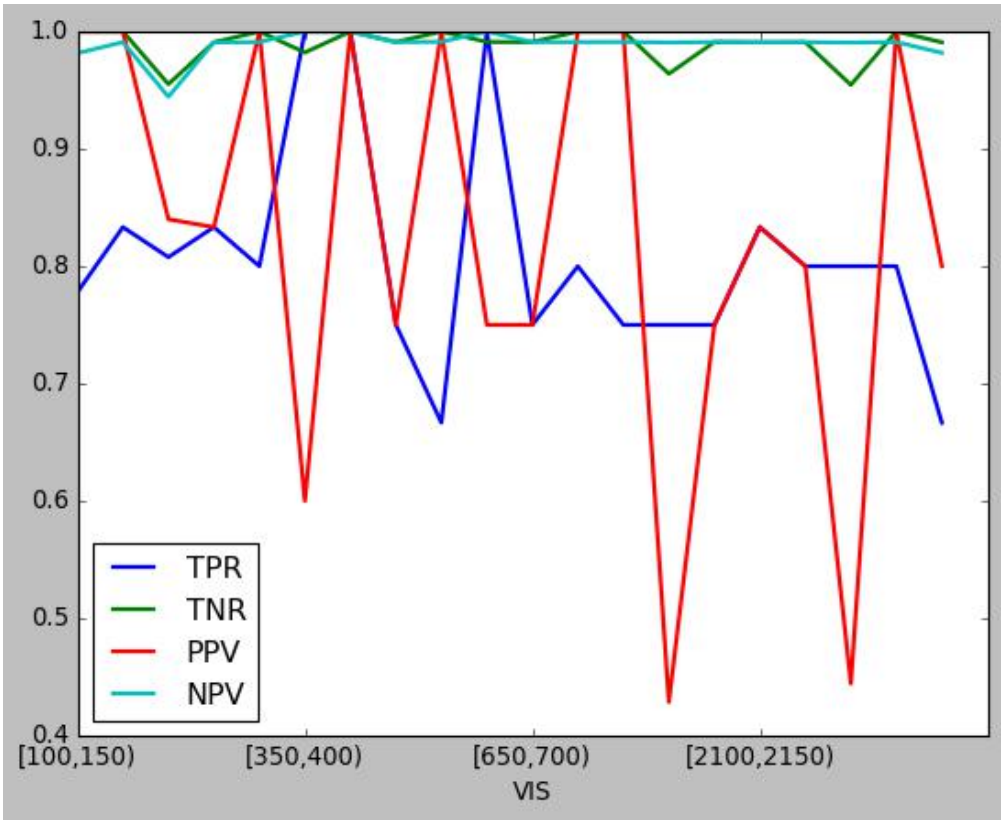

图 6-11 FPR、FNR、PPV、NPV 数量曲线

FPR（False Positive Rate）可以理解为所有反类中，有多少被预测成正类（正类预测错误），给出定义如下：

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$$

FNR（False Negative Rate）可以理解为所有正类中，有多少被预测成反类（反类预测错误），给出定义如下：

$$FNR = \frac{FN}{TP + FN}$$

假发现率 (FDR):

$$FDR = \frac{FP}{FP + TP}$$

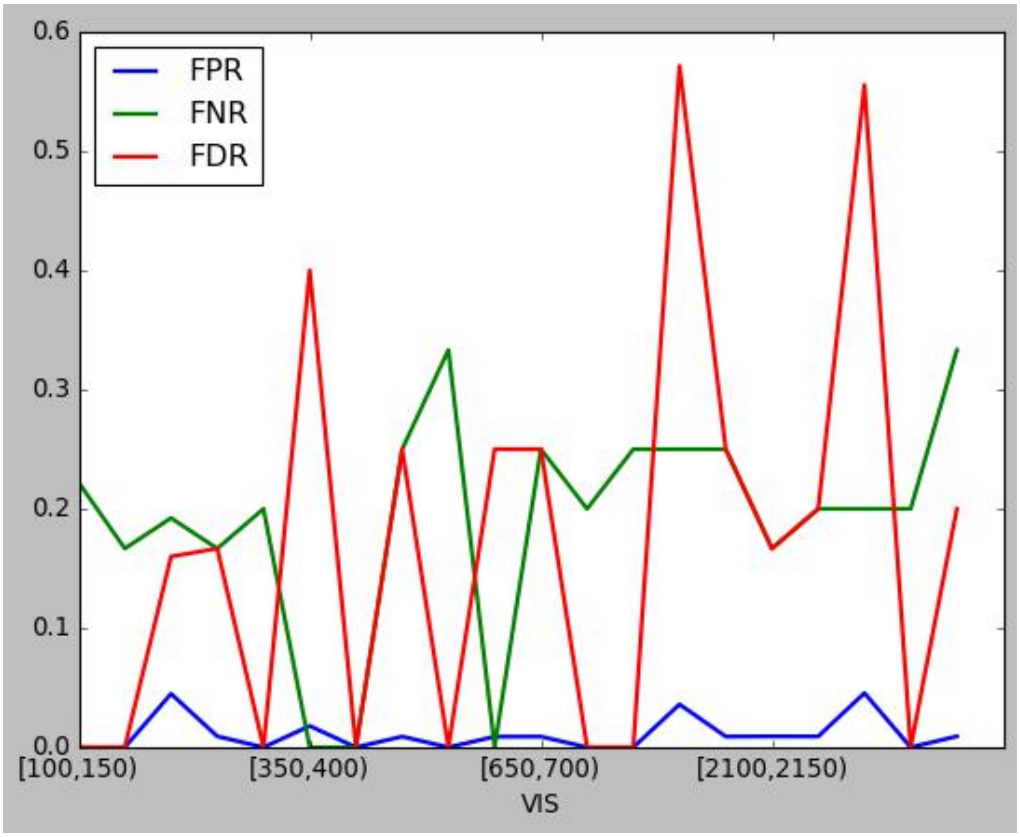

图 6-12 FPR、FNR、FDP 数量曲线

$recall = \frac{TP}{TP+FP}$，指正样本被正确分类数量与总正样本的比率

$precision = \frac{TP}{TP+FP}$，在模型观测是 positive 的所有结果中，模型预测正确的比重。

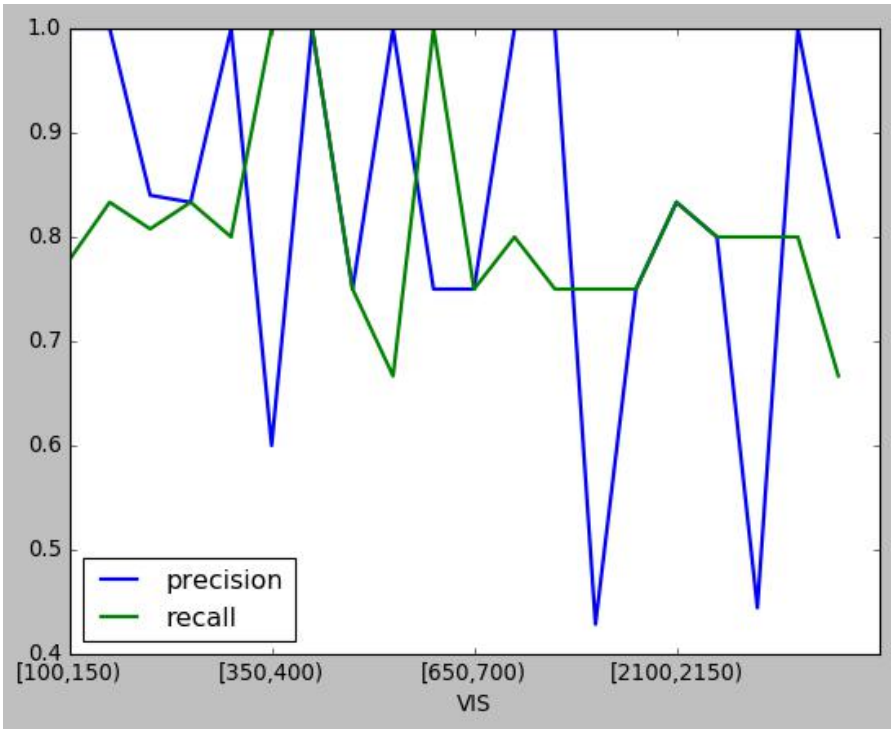

图 6-11 recall、precision 比率

对于多分类问题，通常有“宏”和“微”之分，下表是宏指标。宏查全率（Macro_R）：

$$\text{Macro-P}=\frac{1}{\text{n}}\sum_{i=1}^{n}P_i$$

宏查准率（Macro-P）：

$$\text{Macro-R}=\frac{1}{\text{n}}\sum_{i=1}^{n}R_i$$

宏 F1（Macro_F1）：

$$Macro\_F1=\frac{2\times macro-P\times macro-R}{macro-P+macro-R}$$

表 6-2 宏指标表

	准确率	宏召回率（Macro_R）	宏精准率（Macro_P）	宏 F1（Macro_F1）
性能	0.8	0.829	0.8084	0.8186

4、预测值和真实值的对比曲线

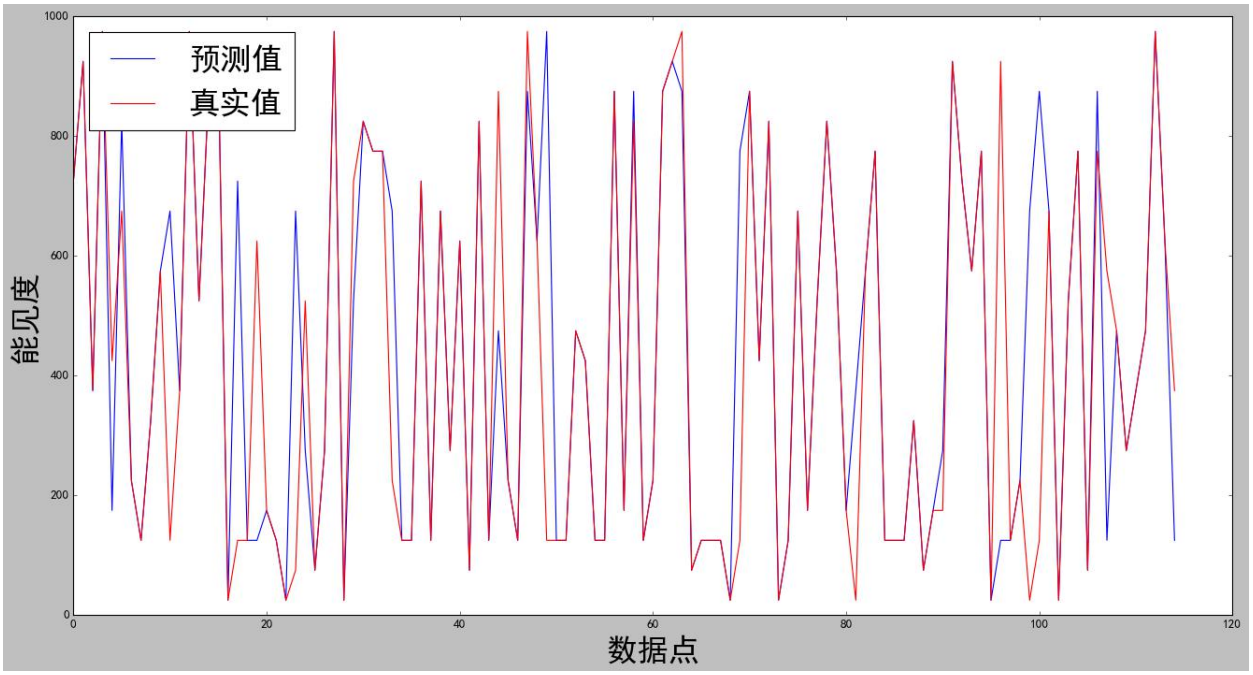

图 6-12 训练集预测值与真实值的对比曲线

由图可以看出预测值与真实值基本可以对应上，模型预测效果较好。

7 问题三模型的建立与求解

问题三要求我们根据高速公路某路段的监控视频数据，建立不依赖能见度仪观测数据的能见度估计算法，并绘制针对题目提供的一组高速公路视频截图，绘制该时间段这段高速公路能见度随时间变化曲线。

7.1 基于坎尼算子的能见度估计模型

检测阶跃边缘的基本思想是在图像中找出具有局部最大梯度幅值的像素点[19]。检测阶跃边缘的大部分工作集中在寻找能够用于实际图像的梯度数字逼近。

由于实际的图像经过了摄像机光学系统和电路系统（带宽限制），固有的低通滤波器的平滑，因此，图像中的阶跃边缘不是十分陡立。

图像也受到摄像机噪声和场景中不希望的细节的干扰。图像梯度逼近必须满足以下两个要求[20]：

1、 逼近必须能够抑制噪声效应；

2、必须尽量精确地确定边缘的位置。

抑制噪声和边缘精确定位是无法同时得到满足的，也就是说，边缘检测算法通过图像平滑算子去除了噪声，但却增加了边缘定位的不确定性；反过来，若提高边缘检测算子对边缘的敏感性，同时也提高了对噪声的敏感性。

有一种先行算子可以在抗噪声干扰和精确定位之间选择一个最佳折中方案，它就是高斯函数的一阶导数，对应于图像的高斯函数平滑和梯度计算[21]。

在高斯噪声中，一个典型的边缘代表一个阶跃的强度变化。根据这个模型，好的边缘检测算子应该有 3 个指标：

1、低失误概率，即真正的边缘点尽可能少的丢失又要尽可能避免将非边缘点检测为边缘；

2、高位置精度，检测的边缘应尽可能接近真实的边缘；

3、对每一个边缘点有惟一的响应，得到单像素宽度的边缘。

基于坎尼算子的能见度估计模型需要遵循以下 3 个边缘算子的准则[22,23]。

7.1.1 信噪比准则

信噪比越大，提取的边缘质量越高。信噪比 SNR 定义为：

$$\mathrm{SNR}=\frac{\left|\int_{-w}^{+w}G(-x)h(x)dx\right|}{\sigma\sqrt{\int_{-w}^{+w}h^{2}(x)dx}}$$

其中，$G(x)$代表边缘函数，$\mathrm{h}(x)$代表宽度为 W 的滤波器的脉冲响应。

7.1.2 定位精确度准则

边缘定位精度 L 如下定义：

$$\mathrm{L}=\frac{\left|\int_{-w}^{+w}G'(-x)h'(x)dx\right|}{\sigma\sqrt{\int_{-w}^{+w}h'^{2}(x)dx}}$$

其中，$G^{'}(-x)$和$h^{'2}(x)$分别是$G(x)$和$\mathrm{h}(x)$的导数。L 越大表示定位精度越高。

7.1.3 单边缘响应准则

为保证单边缘只有一个响应，检测算子的脉冲响应导数和零交叉点平均距离$D(f')$应满足：

$$D(f')=\pi\left\{\frac{\int_{-\infty}^{+\infty}h'^{2}(x)dx}{\int_{-\infty}^{+\infty}h''(x)dx}\right\}^{\frac{1}{2}}$$

其中，$h^{''}(x)$是$h(x)$的二阶导数。

以上述指标和准则为基础，利用泛函数求导的方法可导出坎尼边缘检测器是信噪比与定位之乘积的最优逼近算子，表达式近似于高斯函数的一阶导数。将坎尼 3 个准则相结合可以获得最优的检测算子。

基于坎尼边缘检测的能见度估计算法步骤如下：

（1）用高斯滤波器平滑图像；

（2）用一阶偏导的有限差分来计算梯度的幅值和方向；

（3）对梯度幅值进行非极大值抑制；

（4）用双阈值算法检测和连接边缘，得出边缘图像；

（5）利用 bwarea 函数计算二值边缘图像，即值为 1 的像素点组成的区域，求出其图像面积，利用自行推导的公式进一步计算出能见度数值。

7.2 能见度估计模型的求解

对题目中已知的 100 张高速公路截图进行处理，论文中以‘riginal_frame1’和‘original_frame27’两张图片为例表述能见度估计算法计算流程：

7.2.1 读取原图像

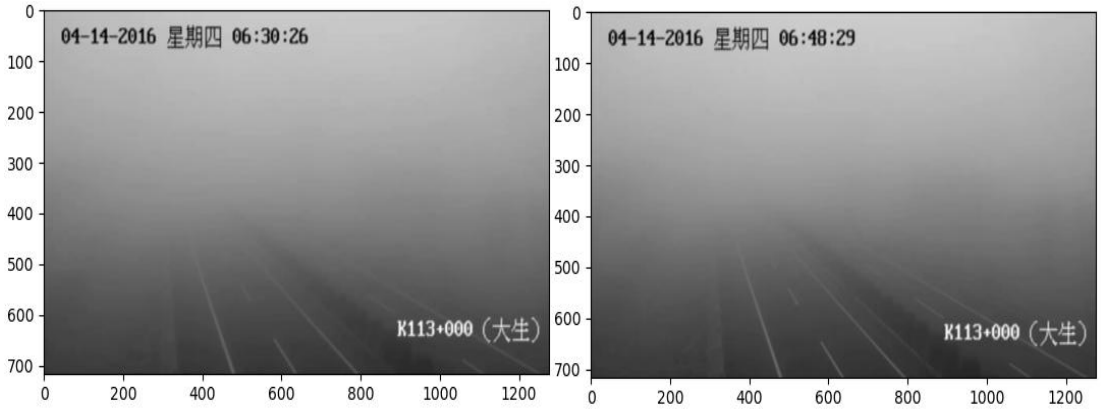

图 7-1 高速公路截图（左：第 1 张；右：第 27 张）

7.2.2 高斯滤波平滑图像

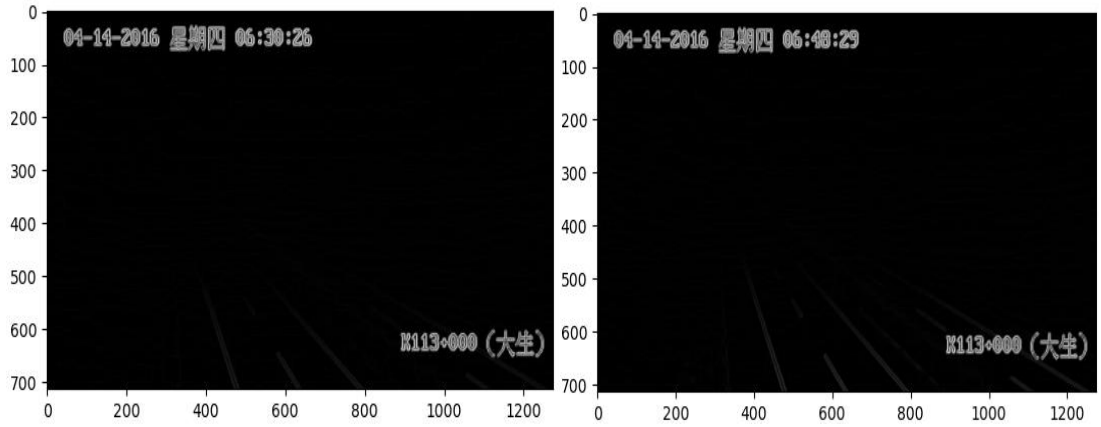

图 7-2 高斯滤波平滑图像（左：第 1 张；右：第 27 张）

7.2.3 用一阶偏导计算梯度和幅值方向

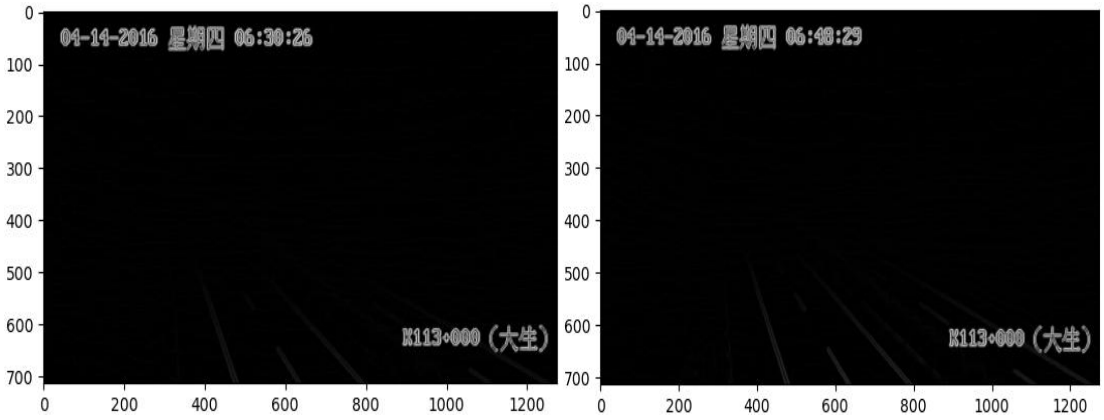

图 7-3 梯度和幅值方向图像（左：第 1 张；右：第 27 张）

7.2.3 对梯度幅值进行非极大值抑制

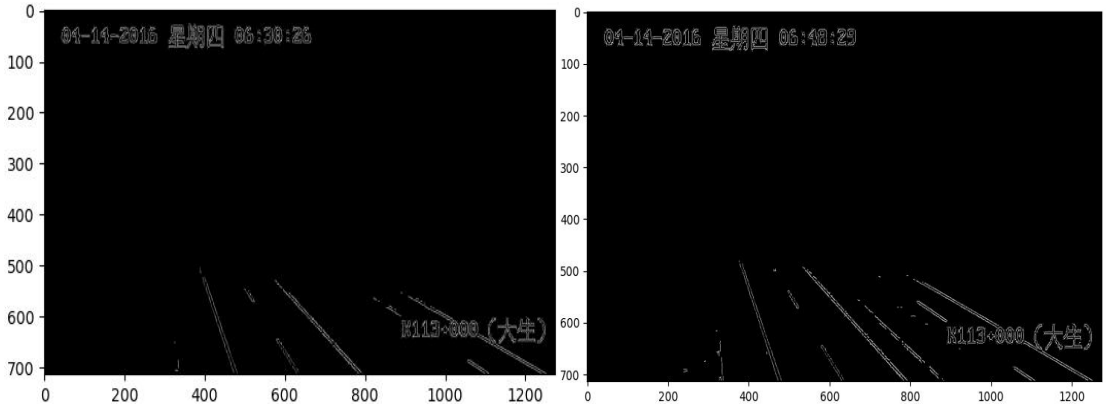

图 7-4 非极大值抑制后的图像（左：第 1 张；右：第 27 张）

7.2.4 用双阀值算法检测和连接边缘

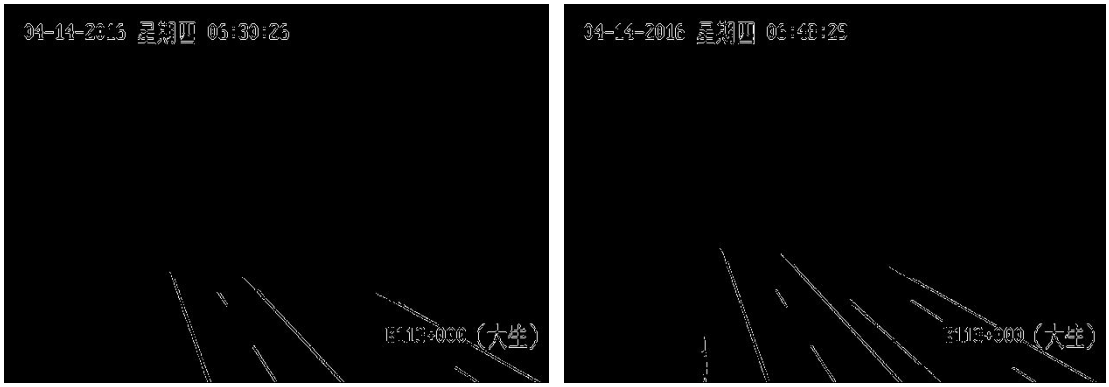

图 7-5 最终边缘检测图像（左：第 1 张；右：第 27 张）

7.2.5 计算图像面积进而估计能见度

根据坎尼算子得出 100 张照片的边缘检测图，根据 Matlab 中 bwarea 函数计算二值图像前景（值为 1 的像素点组成的区域）的面积，即用图片中白色轮廓的面积（如下图所示）大小来表征能见度大小。如“riginal_frame1”和“original_frame27”两张图片为例，其计算二维像素面积如下表所示：

表 7-1 图像边缘面积值

riginal_frame1	riginal_frame27
8,773.625	10,003.375

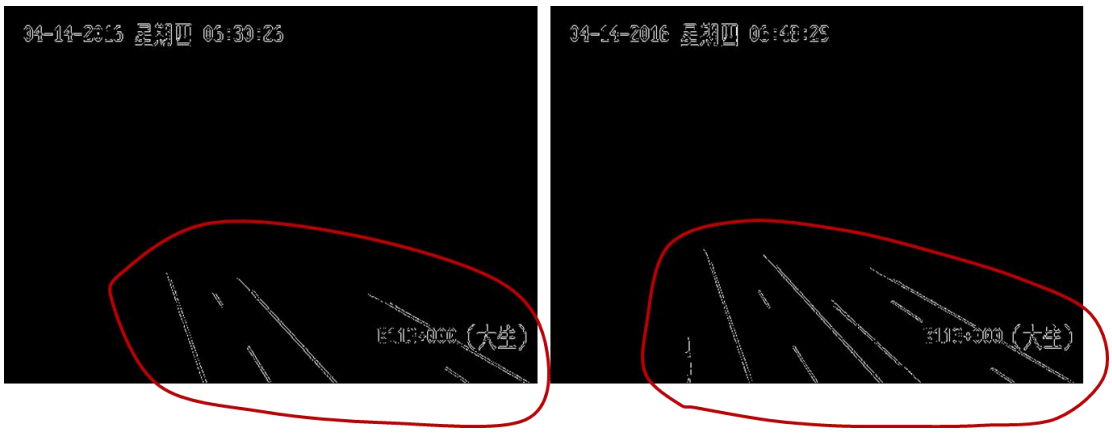

图 7-6 图像边缘面积差异（左：第 1 张；右：第 27 张）

7.2.6 将边缘面积转换为能见度

能见度的大小与边缘轮廓线的面积呈固定比例关系，即

$$VIS = k \times S$$

式中，VIS表示能见度，S表示轮廓线面积。

根据实际情况，并查阅权威资料，高速公里上摄像头高度为 6.5m，高速公路分道线为 69 线，白线 6 米，间隔 9 米，一个周期 15 米。根据以上数据，选取三组图片，根据已知两根中白线间隔的距离为 9m，通过比较比例关系即可得到中线左旁边缘轮廓的实际距离，利用简单三角函数计算即可得出能见度实际数值，如下图所示。利用如上方法，计算 3 组图片，算术平均得出最终 k=0.114。

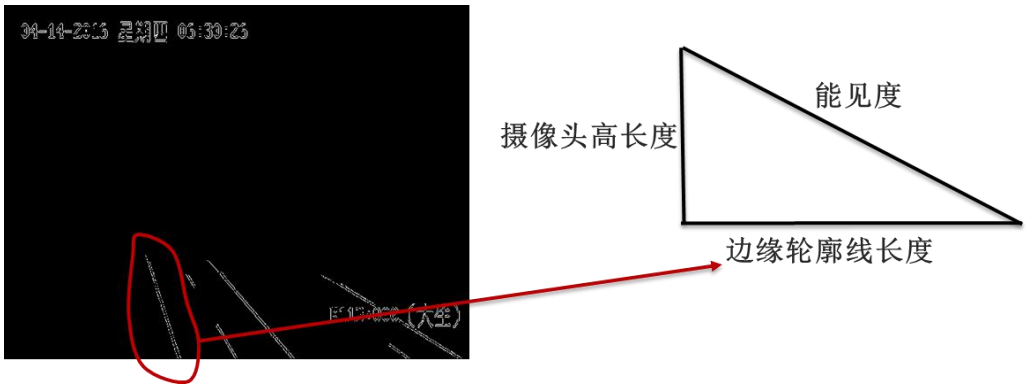

图 7-7 能见度转化示意图

通过以上方法，即可求出高速公路截图中每张照片的能见度具体数值，计算出的能见度随时间变化趋势如下图所示：

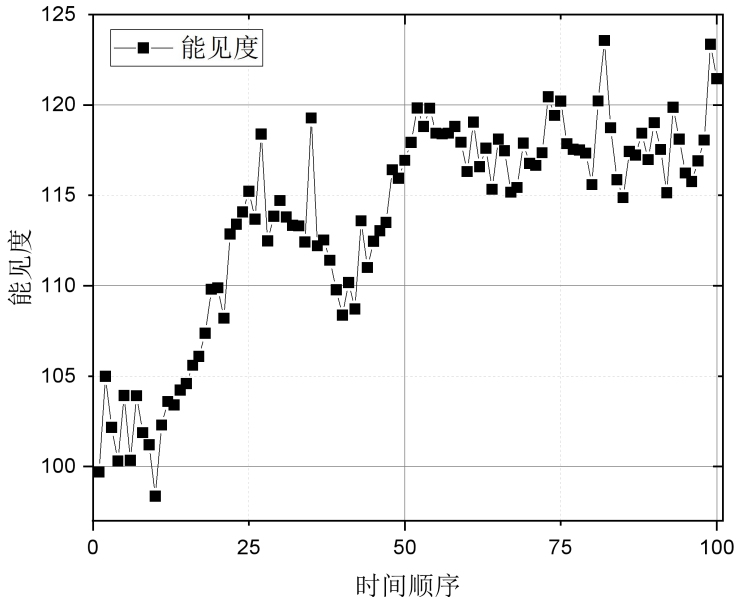

图 7-8 高速公路能见度随时间变化示意图

8 问题四模型的建立与求解

要求利用问题三得到的能见度随时间变化规律，建立数学模型预测大雾变化趋势（加重或减弱）、何时散去（达到指定的能见度，比如 MOR=150m）？

8.1 基于时间序列的 ARIMA 模型

ARIMA（p，d，q）模型全名为差分自回归移动平均模型（Autoregressive Integrated Moving

Average Model，ARIMA）。其中，MA 是移动回归模型，q 为移动平均项数；AR 是自回归模型，p 为自回归项数，d 为时间序列变为平稳时所做的差分阶数[24]。

ARIMA 模型根据原时间序列平稳性及回归过程的不同，包括自回归过程（AR）、自回归移动平均过程（ARMA）以及 ARIMA 过程。

对于非平稳的时间序列，需要对数据进行差分处理[25]，使其变为平稳过程，而对于平稳的时间序列过程，通常在任意时刻 t，都有

$$E(X_t)=\mu,\ Var(X_t)=\sigma^2$$

时间间隔为 k 的两个随机变量x_t与x_{t-k}的协方差即滞后 k 期的自协方差，定义γ_k为自协方差序列，有

$$\gamma_k=Cov(x_t,x_{t-k})=E[(x_t-\mu)(x_{t-k}-\mu)]$$

式中，$k=0,1,2,\cdots,\gamma_k$为随机过程$\{x_t\}$的自协方差函数。当$k=0$时，得

$$\gamma_0=Var(X_t)=\sigma^2{}_x$$

则自相关系数为

$$\rho_k=\frac{Cov(x_t,x_{t-k})}{\sqrt{Var(X_t)}\sqrt{Var(X_{t-k})}}$$

若时间序列是平稳过程，有

$$Var(X_t)=Var(X_{t-k})=\sigma^2{}_x$$

$$\rho_k=\frac{Cov(x_t,x_{t-k})}{\sigma^2{}_x}=\frac{\gamma_k}{\sigma^2{}_x}=\frac{\gamma_k}{\gamma_0}$$

式中，ρ_k为自相关函数，是自相关系数列且变量为滞后期 k。当$k=0$时，有$\rho_0=1$，自相关函数是零对称的，即$\rho_k=\rho_{-k}$。ARMA（p，q）为

$$X_t=\theta_0+\varepsilon_t+k_1X_{t-1}+k_2X_{t-2}+\cdots+k_pX_{t-p}-\lambda_1\varepsilon_{t-1}-\lambda_2\varepsilon_{t-2}-\cdots-\lambda_q\varepsilon_{t-q}$$

式中，$X_t,X_{t-1},\cdots X_{t-p}$为在$t,t-1,\cdots,t-p$时刻得到的观测值；$k_i$为自回归系数；$\theta_0$为常数项；$\varepsilon_t$为误差项；$\lambda_i$为移动平均系数。

对于平稳时间序列 AR（p）模型，有

$$E(x_t)=\mu=\frac{\theta_0}{1-k_1-k_2-\cdots-k_p}$$

$$Var(X_t)=\gamma_0=\frac{\sigma^2{}_a}{1-k_1\rho_1-k_2\rho_2-\cdots-k_p\rho_p}$$

式中，$\sigma^2{}_a$为误差项方差

对 k 阶自回归模型 AR（k），有

$X_t=\theta_0+\varepsilon_t+k_1X_{t-1}+k_2X_{t-2}+\cdots+k_kX_{t-k}$

这里系数k_k恰好表示X_t与X_{t-k}在排除了其中变量。$X_{t-1},X_{t-2},X_{t-k+1}$影响之后的相关系数，即偏自相关系数$\varphi_{kk}$。

当$k=1$，AR（1）模型为

$$\begin{cases} X_t = \theta_0 + \varepsilon_t + k_1 X_{t-1} \\ \varphi_{11} = k_1 = \rho_1 \\ \varphi_{kk} = k_k = 0, k > 1 \end{cases}$$

对平稳时间序列 MA（q）模型

$$X_t = \theta_0 + \varepsilon_t - \lambda_1 \varepsilon_{t-1} - \lambda_2 \varepsilon_{t-2} - \cdots - \lambda_q \varepsilon_{t-q}$$

$$E(x_t) = \mu = \theta_0$$

$$Var(X_t) = \gamma_0 = {\sigma^2}_a \cdot (1 + {\lambda_1}^2 + {\lambda_2}^2 + \cdots + {\lambda_q}^2)$$

$$\rho_k = \begin{cases} \left(-\lambda_k + \lambda_1\lambda_{k+1} + \lambda_2\lambda_{k+2} + \cdots + \lambda_{q-k}\lambda_k\right) \cdot \left(\dfrac{{\sigma^2}_a}{\gamma_0}\right) & k = 1,2,\cdots,q \\ 0 & \text{其他} \end{cases}$$

$$\emptyset_{kk} = -{\lambda_1}^k \left(\frac{1 - {\lambda_1}^2}{1 - {\lambda_1}^{2k+2}}\right)$$

数据预测模型如下图所示，主要有四步。

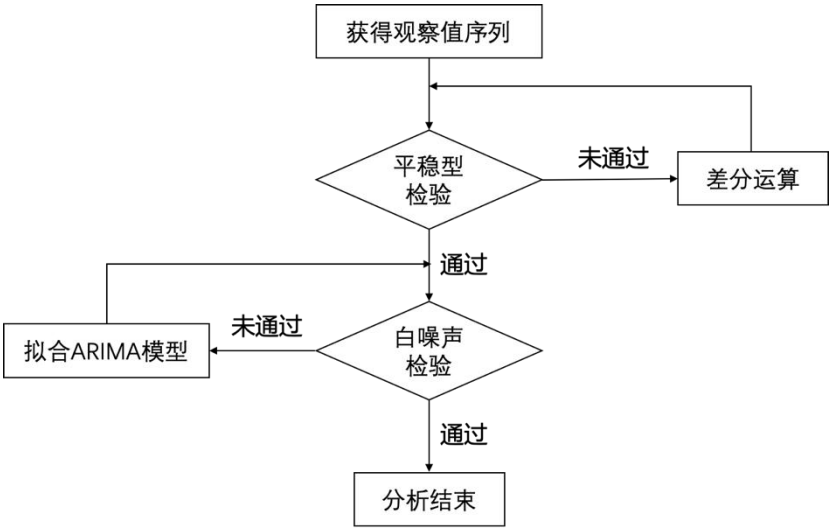

图 8-1 能见度预测流程图

8.2.1 数据平稳性判断及处理

根据时间序列图可以看出，原数据趋势向上，坡度较缓，直观上看出该时间序列具有一定的平稳性，从直方图显示出能见度集中在 110～120 的范围内，自相关图（ACF）和偏自相关图（PACF）均有明显拖尾特征，优先考虑 ARMA 模型。

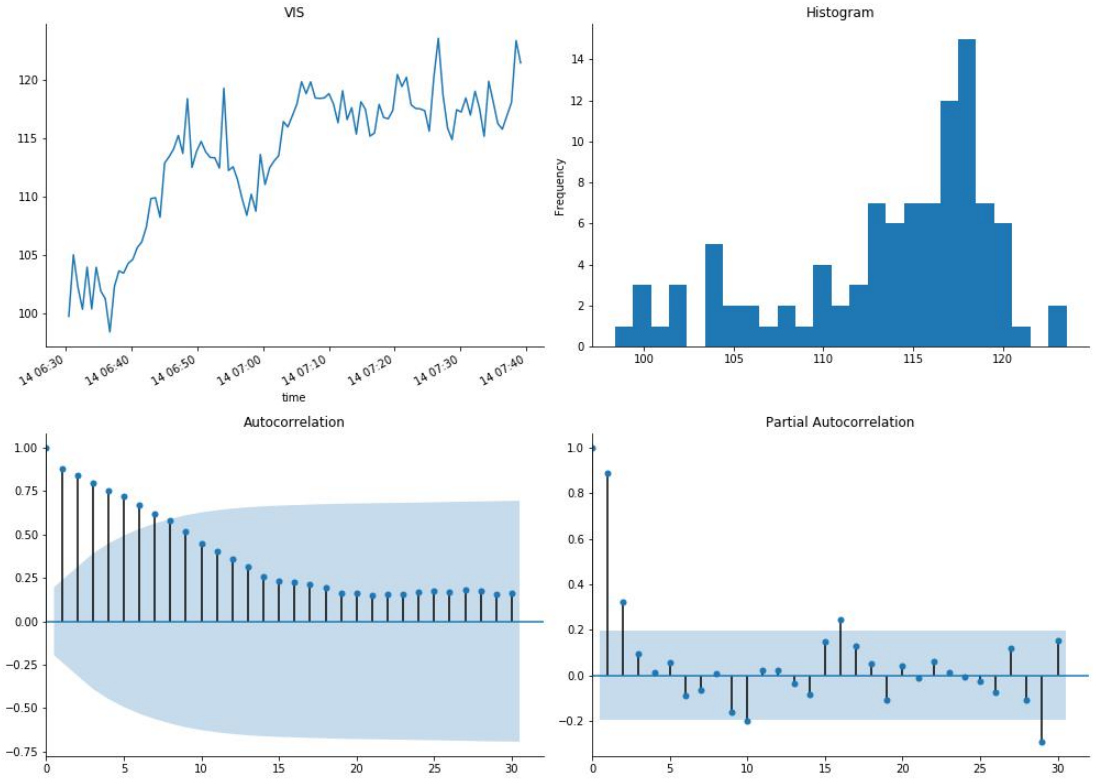

图 8-2 数据自相关图（ACF）和偏自相关图（PACF）

从数据的移动平均值&标准差有越来越大的趋势，是不稳定的，如下图所示

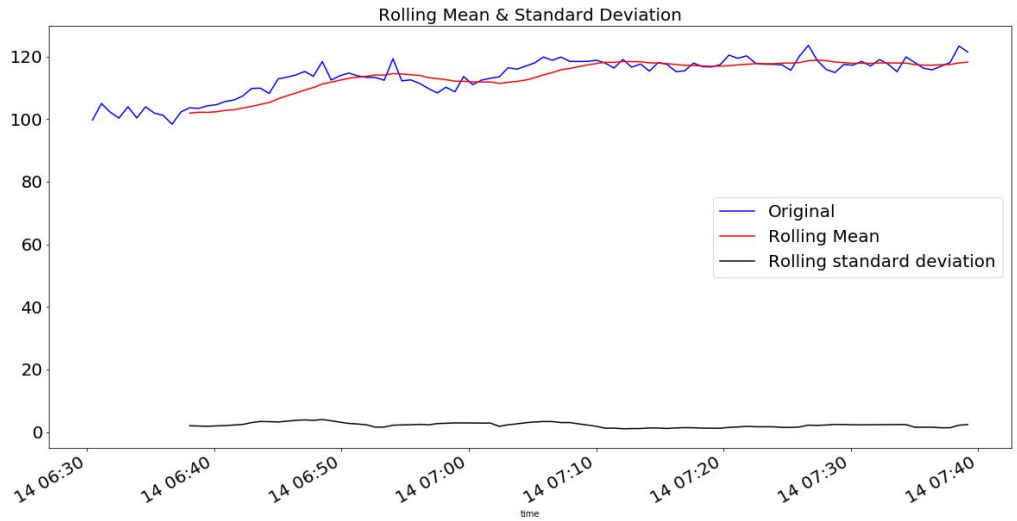

图 8-3 数据的移动平均值&标准差趋势图

接下来在看 Dickey-Fuller 的结果。此时 p 值为 0.496464，说明并不能拒绝原假设。通过 Dickey-Fuller 的数据可以明确的看出，在任何置信度下，数据都不是稳定的。

表 8-1 Dickey-Fuller 的数据

Test Statistic	-1.574351
p-value	0.496464
Number of Observations Used	98.000000
Critical Value (1%)	-3.498910
Critical Value (5%)	-2.582760

将原数据进行一阶差分，绘制移动平均值/标准差图，如下图所示。

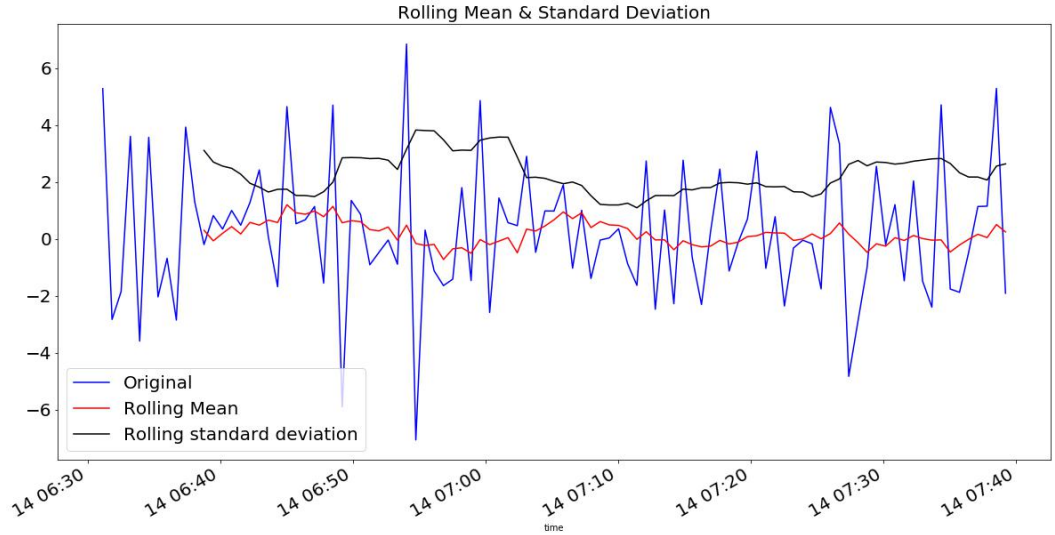

图 8-4 移动平均值/标准差图

可以看出原始数据的移动平均值和移动标准差趋于常数，几乎无波动，所以直观上可以认为是稳定的数据。接下来再分析 Dickey-Fuller 的结果，如下表所示。

表 8-2 处理后 Dickey-Fuller 的数据

Test Statistic	-1.368673e+01
p-value	1.367924e-25
Number of Observations Used	8.800000e+01
Critical Value (1%)	-3.506944e+02
Critical Value (5%)	-2.894990e+00

Dickey-Fuller 的结果显示，Statistic 值远小于 1%时的 Critical value，所以在 99%的置信度下，数据通过平稳性检验。通过对比，差分后，时间序列的平稳性明显提升，如下图所示。

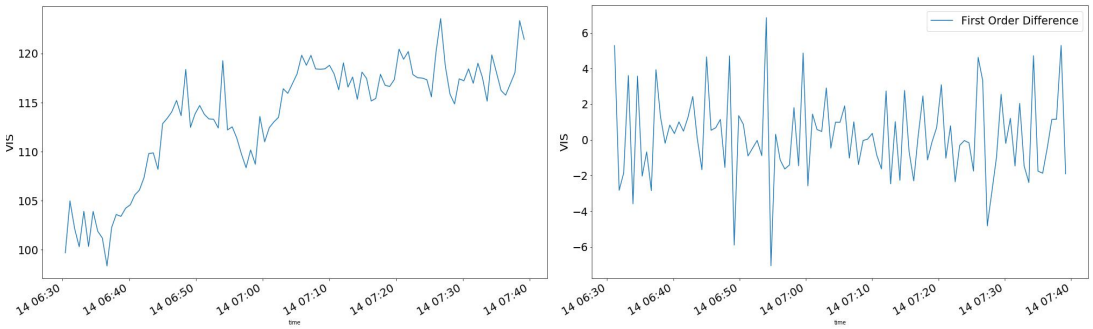

图 8-5 平稳性检验对比

8.2.2 设置 ARIMA 模型参数

现在我们已经得到一个平稳的时间序列，接来下就是选择合适的 ARIMA 模型，即 ARIMA 模型中合适的参数：阶层 p、差分 d、阶数 q。

由于原始数据经过一阶差分处理的情况就通过了平稳性检验，因此差分参数 d 设为 1。经过差分处理之后的数据的自相关图和偏自相关如下图所示。

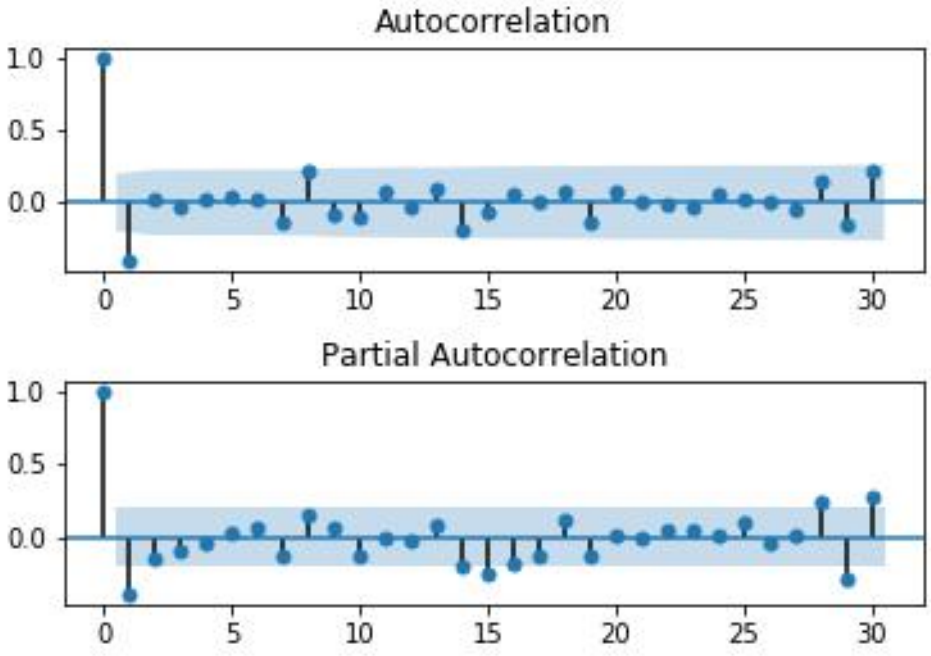

图 8-6 数据偏自相关图

样本自相关系数和样本偏自相关系数在最初的阶明显大于 2 倍标准差，而后几乎 95%的系数都落在 2 倍标准差的范围内，且非零系数衰减为小值波动的过程非常突然，通常视为 k 阶截尾；如果有超过 5%的样本相关系数大于 2 倍标准差，或者非零系数衰减为小值波动的过程比较缓慢或连续，通常视为拖尾。

从自相关图（ACF）和偏自相关图（PACF）可以看出，原数据 ACF 和 PACF 均具有拖尾性，因此我们优先考虑 ARMA 模型。经过观察 ACF 和 PACF，主观选择一个参数 p、q 的范围，再计算对应的 AIC 值、BIC 值进行进一步选择，以 AIC 值、BIC 值趋于小值为评判标准，以 RSME 最小化为最终目标，最终选取阶层 p = 3，阶数 q = 2 为最终参数

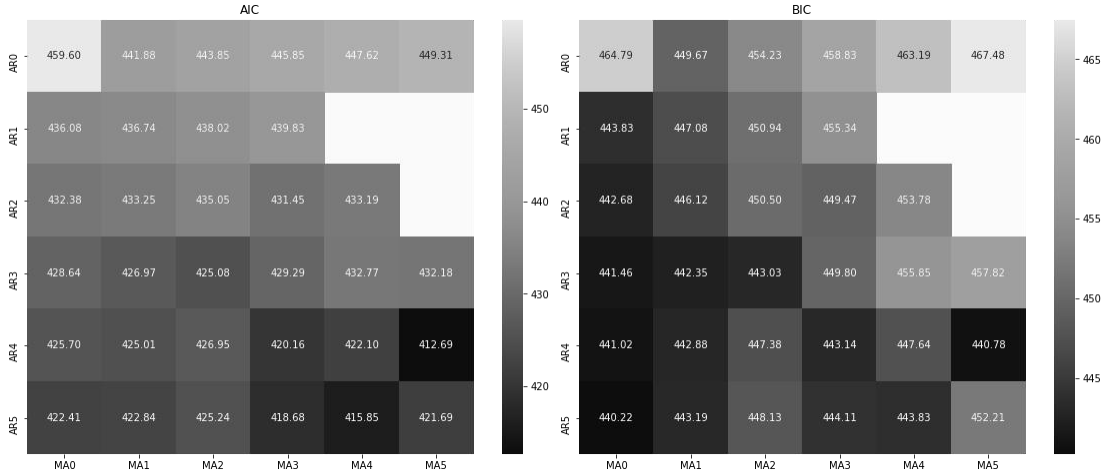

图 8-7 参数热力图

8.2.3 模型预测

通过 ARIMA 模型预测的图形及数据如下图。本题提供提供了 100 个数据样本，为保证预测的准确性，本文预测了 14 个数据，时间间隔约 40 秒。

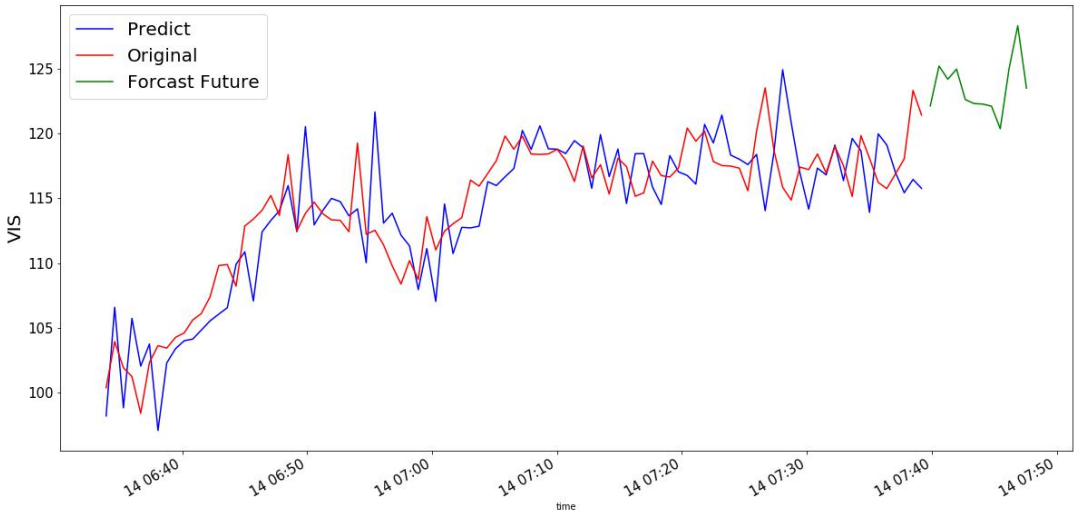

图 8-8 能见度预测趋势图

表 8-3 能见度预测趋势图

时间	能见度（米）	时间	能见度（米）
2016/4/14 7:39	122.1576705	2016/4/14 7:39	122.1576705
2016/4/14 7:40	125.2471591	2016/4/14 7:40	125.2471591
2016/4/14 7:41	124.2201705	2016/4/14 7:41	124.2201705
2016/4/14 7:41	125.0056818	2016/4/14 7:41	125.0056818
2016/4/14 7:42	122.6534091	2016/4/14 7:42	122.6534091
2016/4/14 7:43	122.3409091	2016/4/14 7:43	122.3409091
2016/4/14 7:44	122.3011364	2016/4/14 7:44	122.3011364

参考文献

[1] S. K. Nayar, S. G. Narasimhan, Vision in bad weather, ICCV'99.

[2] R. T. Tan, Visibility in bad weather from a single image, CVPR, 2008.

[3] N. Hautiere, J-P Tarel, J Lavenant D. Aubert, Automatic fog detection and estimation of visibility distance through use of onboard camera, Machine Vision and Application, 2006, 17 (1): 8-20.

[4] C. Sakaridis, D. Dai, L. V. Gool, Semantic foggy scene understanding with synthetic data, International J. Computer Vision, 2018, 3.

[5] E. Castilla,N. Martín,S. Muñoz,L. Pardo. Robust Wald-type tests based on minimum Rényi pseudodistance estimators for the multiple linear regression model[J]. Journal of Statistical Computation and Simulation,2020,90(14).

[6] Science - Chromatography; Recent Findings from Joaquin Hernandez-Fernandez and Co-Researchers Yields New Information on Chromatography (Quantification of oxygenates, sulphides, thiols and permanent gases in propylene. A multiple linear regression model to predict the ...)[J]. Chemicals & Chemistry,2020.

[7] Ahmadreza Rahbari,Tyler R. Josephson,Yangzesheng Sun,Othonas A. Moultos,David Dubbeldam,J. Ilja Siepmann,Thijs J.H. Vlugt. Multiple linear regression and thermodynamic fluctuations are equivalent for computing thermodynamic derivatives from molecular simulation[J]. Fluid Phase Equilibria,2020,523.

[8] G.A. Farias-Basulto,P. Reyes-Figueroa,C. Ulbrich,B. Szyszka,R. Schlatmann,R. Klenk. Validation of a multiple linear regression model for CIGSSe photovoltaic module performance and Pmpp prediction[J]. Solar Energy,2020,208.

[9] Environmental Toxicology; Studies from University of Saskatchewan Provide New Data on Environmental Toxicology (Multiple Linear Regression Modeling Predicts the Effects of Surface Water Chemistry on Acute Vanadium Toxicity to Model Freshwater Organisms) [J]. Ecology Environment & Conservation,2020.

[10] 曹玉东,蔡希彪.基于并行小规模卷积神经网络的图像质量评价[J/OL].计算机工程与科学:1-8[2020-09-20]

[11] 储春洁,王佳雯,韩雅琪,陈胜.基于 Mask R-CNN 模型的胸片肺结节检测性能评估[J/OL].信息与控制:1-7[2020-09-20]

[12] 朱云鹏,黄希,黄嘉兴.基于 3D CNN 的人体动作识别研究[J].现代电子技术,2020,43(18):150-152+156.

[13] 于舒春,佟小雨.基于CNN特征提取的粒子滤波视频跟踪算法研究[J/OL].哈尔滨理工大学学报,2020(04):78-83[2020-09-20]

[14] E. Castilla,N. Martín,S. Muñoz,L. Pardo. Robust Wald-type tests based on minimum Rényi pseudodistance estimators for the multiple linear regression model[J]. Journal of Statistical Computation and Simulation,2020,90(14).

[15] 万齐斌,董方敏,孙水发.基于 BiLSTM-Attention-CNN 混合神经网络的文本分类方法[J].计算机应用与软件,2020,37(09):94-98+201.

[16] Anjali Gautam,Balasubramanian Raman. Towards effective classification of brain hemorrhagic and ischemic stroke using CNN[J]. Biomedical Signal Processing and Control,2021,63.

[17] Donghyeon Kim,Sangwook Park, David K. Han,Hanseok Ko. Multi-band CNN architecture

using adaptive frequency filter for acoustic event classification[J]. Applied Acoustics,2021,172.
[18] Zohaib Mushtaq, Shun-Feng Su,Quoc-Viet Tran. Spectral images based environmental sound classification using CNN with meaningful data augmentation[J]. Applied Acoustics,2021,172.
[19] 李长有,陈国玺,丁云晋.改进 Canny 算子的边缘检测算法[J].小型微型计算机系统,2020,41(08):1758-1762.
[20] Cesar Bustacara-Medina, Leonardo Florez-Valencia, Luis Carlos Diaz. Improved Canny Edge Detector Using Principal Curvatures[J]. Journal of Electrical and Electronic Engineering,2020,8(4).
[21] 李庆忠,刘洋.基于改进 Canny 算子的图像弱边缘检测算法[J].计算机应用研究,2020,37(S1):361-363.
[22] Zhang Xiao,Chen Fuen. Lane Line Edge Detection Based on Improved Adaptive Canny Algorithm[J]. Journal of Physics: Conference Series,2020,1549(2).
[23] Leonardo Bandeira Soares,Julio Oliveira,Eduardo Antonio César da Costa,Sergio Bampi. An Energy-Efficient and Approximate Accelerator Design for Real-Time Canny Edge Detection[J]. Circuits, Systems, and Signal Processing,2020(prepublish).
[24] 吕锟,姜帅,沈瑾,何振文,郭琳晶,陈捷雄,马泽义.惠州城区供电量与气象要素的关系及其预测模型[J].广东气象,2019,41(04):51-53+57.
[25] 马汉平, 兰州市典型传染病与气象因素的关系及其预测模型的研究. 甘肃省,兰州市疾病预防控制中心,2019-07-09.

附录

部分代码，全部代码详见支撑文件

1、基于 CNN 的能见度分类预测模型核心代码：

```
import keras
from keras.models import load_model
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D
import    numpy as np
import keras.backend.tensorflow_backend as KTF

import os
import tensorflow as tf
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

#设定为自增长
config = tf.compat.v1.ConfigProto() #tf.ConfigProto()
config.gpu_options.allow_growth=True
session = tf.compat.v1.Session(config=config) #tf.Session(config=config)
# KTF.set_session(session)
# tf.compat.v1.keras.backend.set_session(session)
tf.compat.v1.keras.backend.set_session



####输入######
version_name = '6'
num_classes = 100
#分类数
nppath = "./DataSet/"


#  加载数据
x_train = np.load(nppath + f"x_train{version_name}.npy")
x_test = np.load(nppath + f"x_test{version_name}.npy")
y_train = np.load(nppath + f"y_train{version_name}.npy")
y_test = np.load(nppath + f"y_test{version_name}.npy")

print(x_train.shape,y_train.shape,x_test.shape,y_test.shape)
#

x_train = x_train.astype('float64')/255
```

```
x_test = x_test.astype('float64')/255

# Convert class vectors to binary class matrices.
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)



model = Sequential()

model.add(Conv2D(32, kernel_size=(1,1),input_shape=x_train.shape[1:], activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, kernel_size=(5,5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

# beginning of fully connected neural network.
model.add(Dense(100, activation='relu'))
model.add(Dropout(0.5))
# Add fully connected layer with a softmax activation function
model.add(Dense(num_classes, activation='softmax'))

# Compile neural network
model.compile(loss='categorical_crossentropy', # Cross-entropy
                        optimizer='rmsprop', # Root Mean Square Propagation
                        metrics=['accuracy']) # Accuracy performance metric



# begin train the data
history = model.fit(x_train, # train data
                    y_train, # label
                    epochs=40, # Number of epochs
                    verbose=2,
                        batch_size=1000)
model.save(f"./cnnmodel{version_name}.h5")

# evaluate
loss, accuracy = model.evaluate(x_test, y_test)

print(loss, accuracy)
```

2、线性回归预测主要代码：

```
## 导入数据

import pandas as pd
import numpy as np

ptu15 = pd.read_csv('PTU_R06_15.csv')
vis15 = pd.read_csv('VIS_R06_15.csv')
wind15 = pd.read_csv('WIND_R06_15.csv')

ptu12 = pd.read_csv('PTU_R06_12.csv')
vis12 = pd.read_csv('VIS_R06_12.csv')
wind12 = pd.read_csv('WIND_R06_12.csv')

# 修改 ptu15 列名
ptu15.columns = ['CREATEDATE', 'LOCALDATE (BEIJING)', 'SITE', 'PAINS',
        'QNH AERODROME', 'ST', 'QFE R06', 'ST.1', 'QFE R24',
        'ST.2', 'QFF AERODROME', 'TREND', 'TENDENCY', 'TEMP',
        'RH', 'DEWPOINT', 'TU DATA']

ptu15_select = ptu15[['CREATEDATE','LOCALDATE (BEIJING)','SITE','PAINS','QFE R06','QNH
AERODROME','TEMP','RH','DEWPOINT']]

ptu12.columns = ['CREATEDATE', 'LOCALDATE (BEIJING)', 'SITE', 'PAINS',
        'QNH AERODROME', 'ST', 'QFE R06', 'ST.1', 'QFE R24',
        'ST.2', 'QFF AERODROME', 'TREND', 'TENDENCY', 'TEMP',
        'RH', 'DEWPOINT', 'TU DATA']
ptu12_select = ptu12[['CREATEDATE','LOCALDATE (BEIJING)','SITE','PAINS','QFE R06','QNH
AERODROME','TEMP','RH','DEWPOINT']]

# 对 VIS、WIND 表同时间戳数据取平均
vis15_mean = vis15[['CREATEDATE','LOCALDATE
(BEIJING)','SITE','RVR_1A','MOR_1A','LIGHTS']].groupby(by = ['CREATEDATE','LOCALDATE
(BEIJING)','SITE']).mean().reset_index()
vis12_mean = vis12[['CREATEDATE','LOCALDATE
(BEIJING)','SITE','RVR_1A','MOR_1A','LIGHTS']].groupby(by = ['CREATEDATE','LOCALDATE
(BEIJING)','SITE']).mean().reset_index()
wind15_mean = wind15[['CREATEDATE','LOCALDATE (BEIJING)','SITE','WS2A
(MPS)','WD2A','CW2A (MPS)']].groupby(by = ['CREATEDATE','LOCALDATE
(BEIJING)','SITE']).mean().reset_index()
wind12_mean = wind12[['CREATEDATE','LOCALDATE (BEIJING)','SITE','WS2A
(MPS)','WD2A','CW2A (MPS)']].groupby(by = ['CREATEDATE','LOCALDATE
(BEIJING)','SITE']).mean().reset_index()

#并对可见度进行合成：VIS
```

```
vis15_mean['VIS']= np.where(vis15_mean['RVR_1A'] <=2000, vis15_mean['RVR_1A'], vis15_mean['MOR_1A'])
vis12_mean['VIS']= np.where(vis12_mean['RVR_1A'] <=2000, vis12_mean['RVR_1A'], vis12_mean['MOR_1A'])

data15 = ptu15_select.merge(vis15_mean,on = ['CREATEDATE','LOCALDATE (BEIJING)','SITE'] , how = 'inner')

data15 = data15.merge(wind15_mean,on = ['CREATEDATE','LOCALDATE (BEIJING)','SITE'] , how = 'inner')

data12 = ptu12_select.merge(vis12_mean,on = ['CREATEDATE','LOCALDATE (BEIJING)','SITE'] , how = 'inner')

data12 = data12.merge(wind12_mean,on = ['CREATEDATE','LOCALDATE (BEIJING)','SITE'] , how = 'inner')

# 重新保存
df1215 = data15.copy()
df1215.columns = ['CREATEDATE', 'LOCALDATE', 'SITE', 'PAINS', 'QFE',
        'QNH', 'TEMP', 'RH', 'DEWPOINT', 'RVR_1A', 'MOR_1A', 'LIGHTS',
        'VIS', 'WS2A', 'WD2A', 'CW2A']
# #删除 CW2A 为 0 的行
# df1215 = df1215[df1215['CW2A'] != 0]
# df1215.to_csv('df1215.csv')

# 重新保存
df0312 = data12.copy()
df0312.columns = ['CREATEDATE', 'LOCALDATE', 'SITE', 'PAINS', 'QFE',
        'QNH', 'TEMP', 'RH', 'DEWPOINT', 'RVR_1A', 'MOR_1A', 'LIGHTS',
        'VIS', 'WS2A', 'WD2A', 'CW2A']
# #删除 CW2A 为 0 的行
# df0312 = df0312[df0312['CW2A'] != 0]
# df0312.to_csv('df0312.csv')

## 初步数据可视化

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei'] # 步骤一（替换 sans-serif 字体）
plt.rcParams['axes.unicode_minus'] = False    # 步骤二（解决坐标轴负数的负号显示问题）
import seaborn as sns
import matplotlib.pyplot as plt

#灯光、可见度、时间
```

```
plt.figure(figsize=(20,10))
plt.plot(np.arange(len(df1215['CREATEDATE'])),df1215['LIGHTS'],'b',label="LIGHTS")
plt.plot(np.arange(len(df1215['CREATEDATE'])),df1215['VIS'],'r',label="VIS")
plt.legend(loc="upper right") #显示图中的标签
# plt.xlabel("the number of sales")
# plt.ylabel('value of sales')
plt.title('20191215')
plt.show()

#灯光、可见度、时间
plt.figure(figsize=(20,10))
plt.plot(np.arange(len(df0312['CREATEDATE'])),df0312['LIGHTS'],'b',label="LIGHTS")
plt.plot(np.arange(len(df0312['CREATEDATE'])),df0312['VIS'],'r',label="VIS")
plt.legend(loc="upper right") #显示图中的标签
# plt.xlabel("the number of sales")
# plt.ylabel('value of sales')
plt.title('20200312')
plt.show()

## 世界时间 17:00-23:59 与 00:00-17:00 分段处理
---
直观看来了 LIGHTS 和可见度没有什么关系

df,day = df1215.copy(),'2019/12/15'
idx1 = df[df['CREATEDATE'] == day+ ' 6:00'].index
idx2 = df[df['CREATEDATE'] == day +' 17:00'].index
print(idx1,idx2)

def dayCut(df):
    dfd = df.iloc[360:1021,:]
    return dfd

df1d = dayCut(df1215)
df2d = dayCut(df0312)

## 将两天白天的数据合并

df = pd.concat([df1d,df2d])

## 数据预处理
---
RVR 和 MOR 分别为两种定义下的能见度，其中 RVR≤2000 米，MOR≤10000 米

def dataPre(data15):
```

```
    #去重 和去空值
    data15.dropna(inplace = True)
    data15.drop_duplicates(inplace = True)

    #去极值
    def filter_extreme_3sigma(series,n=3): #3 sigma
        mean = series.mean()
        std = series.std()
        max_range = mean + n*std
        min_range = mean - n*std
    #     print(max_range)
    #     print(min_range)
        return np.clip(series,min_range,max_range)
    for col in data15.columns:
        if isinstance(data15[col].iloc[0],str):
    #         print(col)
            pass
        else:
            data15[col] = filter_extreme_3sigma(data15[col])

    #标准化
    from sklearn.preprocessing import StandardScaler

## 标准化(使特征数据方差为 1,均值为 0)

    # 使用 sklearn 的包
    scaler = StandardScaler()
    data_notstr = data15.iloc[:,3:].copy()
    scaler.fit(data_notstr)                                    # 使用 transfrom 必须要用
fit 语句
    data_notstr = scaler.transform(data_notstr)              # transfrom 通过找中心和缩放
等实现标准化
#     fit_trans_data_2 = scaler.fit_transform(data_2)    # fit_transfrom 为先拟合数据,然后
转化它将其转化为标准形式
    data15.iloc[:,3:] = data_notstr
#     print('使用 fit,transform 标准化的数据:\n', data15)
#     print('使用 fit_transform 标准化的数据:\n', fit_trans_data_2)
    return data15

# temp = df1215[(df1215.VIS!=5000) & (df1215.VIS!=6000) & (df1215.VIS!=7000) ].copy()
df_pro = dataPre(df)

## 相关性分析
```

```
df_pro.columns = ['CREATEDATE',
  'LOCALDATE',
  'SITE',
  'PAINS',
  'QFE',
  'QNH',
  'TEMP',
  'RH',
  'DEWPOINT',
  'WS2A',
  'WD2A',
  'CW2A',
  'RVR_1A',
  'MOR_1A',
  'LIGHTS',
  'VIS']

aa = df_pro.corr()
aa.apply(lambda x: x.round(2))

plt.figure(figsize=(20, 15))
sns.set(font_scale=2) # font size 2
sns.heatmap(df_pro.corr(), annot=True)
plt.savefig("./结果/heatmap.png")

plt.figure(figsize=(20, 15))
sns.set(font_scale=2) # font size 2
sns.pairplot(df_pro[[
  'PAINS',
  'QFE',
  'QNH',
  'TEMP',
  'RH',
  'DEWPOINT',
  'WS2A',
  'WD2A',
  'CW2A',
  'VIS']],kind='reg')
plt.savefig("./结果/pairplot.png")
plt.show()

##  多元线性回归

import seaborn as sns
```

```
import matplotlib.pyplot as plt
data = df_pro.copy()
# visualize the relationship between the features and the response using scatterplots
feature_cols = ['PAINS','TEMP','RH','WS2A'] #'PAINS',
flag_col = 'VIS'

sns.set(font_scale=2) # font size 2
sns.pairplot(data, x_vars=feature_cols[:2], y_vars='VIS', height=10, aspect=0.8, kind='reg')
plt.savefig('./结果/关键变量相关性 1.png')
plt.show()#注意必须加上这一句，否则无法显示。

sns.set(font_scale=2) # font size 2
sns.pairplot(data, x_vars=feature_cols[2:], y_vars='VIS', height=10, aspect=0.8, kind='reg')
plt.savefig('./结果/关键变量相关性 2.png')
plt.show()#注意必须加上这一句，否则无法显示。

#seaborn 的 pairplot 函数绘制 X 的每一维度和对应 Y 的散点图。通过设置 size 和 aspect 参数来调节显示的大小和比例。
#可以从图中看出，TV 特征和销量是有比较强的线性关系的，而 Radio 和 Sales 线性关系弱一些，Newspaper 和 Sales 线性关系更弱。
#通过加入一个参数 kind='reg'，seaborn 可以添加一条最佳拟合直线和 95%的置信带。

#create a python list of feature names
# equivalent command to do this in one line
X = data[feature_cols]
y = data[flag_col]

##构造训练集和测试集
#default split is 75% for training and 25% for testing
from sklearn.model_selection import train_test_split    #这里是引用了交叉验证
X_train,X_test, y_train, y_test = train_test_split(X, y,test_size=0.25, random_state=1)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

from sklearn.linear_model import LinearRegression
linreg = LinearRegression()
model=linreg.fit(X_train, y_train)
linreg.fit(X_train, y_train)
print(model)
print(linreg.intercept_)
print(linreg.coef_)
```

```
# pair the feature names with the coefficients
print(flag_col+' = ',end = '')
for i,j in zip(feature_cols, linreg.coef_):
    print(f'({j.round(4)}) * {i} + ',end = '')
print(f'({linreg.intercept_})')
```

评价回归问题

(1)平均绝对误差(Mean Absolute Error, MAE)

(2)均方误差(Mean Squared Error, MSE)

(3)均方根误差(Root Mean Squared Error, RMSE)

```
# #模型评估

# print(type(y_pred),type(y_test))
# print (len(y_pred),len(y_test))
# print (y_pred.shape,y_test.shape)
# from sklearn import metrics
# import numpy as np
# sum_mean=0
# for i in range(len(y_pred)):
#     sum_mean+=(y_pred[i]-y_test.values[i])**2
# sum_erro=np.sqrt(sum_mean/50)
# # calculate RMSE by hand
# print("RMSE by hand:",sum_erro)
#RMSE 越小越好
r_sq = model.score(X_test, y_test)
print('TEST')
print('coefficient of determination(R^2) :', r_sq)

y_pred = model.predict(X_test)
from sklearn import metrics
MSE = metrics.mean_squared_error(y_test, y_pred)
RMSE = np.sqrt(metrics.mean_squared_error(y_test, y_pred))

print('MSE:',MSE)
print('RMSE:',RMSE)

r_sq = model.score(X_train, y_train)
print('TRAIN')
print('coefficient of determination(R^2) :', r_sq)
y_train_pred = model.predict(X_train)
```

```
from sklearn import metrics
MSE = metrics.mean_squared_error(y_train, y_train_pred)
RMSE = np.sqrt(metrics.mean_squared_error(y_train, y_train_pred))

print('MSE:',MSE)
print('RMSE:',RMSE)

#  训练集绘图
plt.figure(figsize=(20,10))
x = np.arange(len(X_train))
y_train_hat = model.predict(X_train)
plt.scatter(x,y_train)
plt.plot(x, y_train_hat)
plt.show()

font = {'weight' : 'normal',
'size' : 30,
}

plt.rcParams['font.sans-serif']=['SimHei']#黑体
#  测试集绘图
y_pred = linreg.predict(X_test)
import matplotlib.pyplot as plt
plt.figure(figsize=(20,10))

plt.plot(range(len(y_pred)),y_pred,'b',label="预测值")
plt.plot(range(len(y_pred)),y_test,'r',label="真实值")
plt.legend(loc="upper right",prop = font) #显示图中的标签
# plt.title('测试集对比图')
plt.ylabel("能见度",font)
plt.xlabel('时间点',font)
plt.savefig('./结果/测试集对比.png')
plt.show()
```