


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	华中科技大学
参赛队号	20104870012
队员姓名	1.桂丽婷
	2.王海华
	3.樊启润

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　　探索大雾演化规律，预测大雾变化趋势

摘　　要：

在气象观测、高速公路行驶、航班制定等场景中能见度一直都是不可或缺的指标之一。影响能见度的主要因素之一是雾。在此背景下，本文主要研究了在大雾情况下能见度主要影响因素和诸多估计方法，对给定数据进行了细致处理，并综合运用主成分分析、多元回归分析、预训练模型图像特征提取、随机森林深度学习算法、LSTM 神经网络、摄像机标定算法等统计与算法知识建立了相关问题的数学模型，并利用 Python、Matlab 等工具计算出了合理的结论。

针对问题一，为了保证时间序列的连续性以及能见度指标、气象勘测指标的完整性，采用插值法将缺失的部分数据进行补齐，同时通过加权插值方法对气象勘测值进行扩充。由于题目提供的数据量种类较多、且呈现时间序列，故采用**分区间的多元线性回归模型**来描述能见度与气象指标的关系。首先对 9 项气象勘测指标进行 **Pearson 相关性分析，**判断变量之间的相关性强弱。基于得到的相关系数矩阵，对 9 项气象观测数据进行**主成分分析**，实现降维的效果，得到量化的 **5 项主成分**。将主成分**代入多元线性回归模型进行求解**，分别在白天强光照、白天弱光照、黑夜强光照、黑夜弱光照四种条件中求得回归系数，并对其进行**显著性检验**，最终得到不同条件下的 MOR1A、RVR1A 和主成分之间的关系式。测试结果表明该模型对 RVR 和 MOR 的校正R^2,均在 **0.970 以上，**说明该模型能较好地描述能见度与气象勘测数据之间的关系。根据**主成分的实际含义解释来间接描述能见度与各个气象观测指标的内在联系**。最终推导出四种条件下的关系式，共计 **6 组多元一次方程，以表征能见度与气象观测指标的关系。**

针对问题二，首先，为了保留机场观测视频的连续信息，采用 OpenCV 按照 AMOS 能见度观测数据进行图像分割获取时间序列图像集合。接着，基于迁移学习理论使用 ResNet18 预训练模型提取图像高层特征，最终构建一个具有 **1410 条目、512 维特征向量**的时间序列数据集。然后，运用随机森林深度学习回归预测模型对图像特征向量时间序列进行建模，并且对该模型进行残差分析，其残差服从正态分布说明了模型的可靠性。接下来，使用 LSTM 长短期记忆神经网络对随机森林能见度估计模型的残差进行回归分析，该模型在 epoch 为 150 时对残差的估计准确率达到 **98%**。最后，构建**随机森林-LSTM 能见度估计模型**，该模型能见度估计值为随机森林模型能见度估计值与 LSTM 神经网络模型残差估计值之和，测试结果表明该模型对 RVR 和 MOR 的校正R^2值分别为 **0.979 和 0.958**，说明了**模型能见度估计效果良好、精度较高**。

针对问题三，为了估计 100 张高速公路图像对应的能见度，建立**临界亮度对比-摄像机标定能见度估计模型**估计每张图像的能见度。首先，该模型需要获取每张图像的最远可视

点，采用**基于临界亮度对比的最远视点确定算法**，通过图像预处理、图像天空区域和道路区域分割、天空亮度与道路亮度计算，最终结合基于 Koschmieder 定律与人眼识别亮度对比阈值所得的约束条件，获取每张视频截图中的图像最远视点。得到每张图像的最远可视点之后，需要建立图像坐标系与世界坐标系的关系，本文采用**基于摄像机标定的距离映射算法**，通过图像增强、Canny 算子边缘检测、基于 Hough 变换的车道线提取，并最终根据车道线和摄像机标定距离映射表达式拟合出图像坐标系到世界坐标系的距离映射公式。最后，结合每张图像的最远视点数据和摄像机标定距离映射公式计算 100 张高速公路图像的能见度预测值，并通过多项式回归分析拟合能见度随时间的变化规律，最终**绘制了表达式为$y = 0.0001483x^3 - 0.02221x^2 + 1.128x + 23.93$的能见度随时间变化曲线。**

针对问题四，基于问题三所得到的高速公路能见度随时间变化曲线，采用**多项式函数模型对曲线进行回归拟合**，对比曲线拟合效果，本文采取三次多项式函数曲线来描述高速公路能见度随时间的变化关系。根据能见度测量基本方程及人眼对比度阈值得到衡量大雾浓度的指标衰减系数与能见度之间的关系式，**结合能见度表达式获得大雾浓度指标衰减指数随时间的变化关系，结果发现大雾的变化趋势是减弱**。假定气象能见度 MOR 值为 150m，通过代入能见度表达式计算可知，**当时间为 08:08 时大雾散去**。

关键词：多元回归分析；随机森林-LSTM；临界亮度对比；摄像机标定

目 录

一：问题重述

1.1 问题背景

在气象观测、高速公路行驶、航班制定等场景中能见度一直都是不可或缺的指标之一。影响能见度的因素主要是雾和霾。高速公路上，能见度对于行驶安全而言是一项重要指标。当能见度很低时，管理人员为了保证车辆行驶的安全，通常会暂时封闭道路。而在航空领域，通常用跑道能见度来反映机场附近雾和霾的程度。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。通常情况下，机场能见度小于 400 米时会禁止航班起降。当能见度只有 600-800 米时，机场出于安全考虑会临时控制航班流量的措施，拉大航班起飞间隔。其结果就是容易造成航班的延误。因此，高速公路管理部门和航空公司对于能见度预测方面的问题一直都十分关注。

目前，我国高速路网已逐步形成。激光能见度仪虽然是一种常用的检测能见度的仪器，但是如果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。当下，机器视觉技术、图像处理算法、以及成像设备逐渐完善优化。基于视频来对能见度进行检测的方法也被广泛关注。它在使用成本、应用便捷性、设备维护等方面都要优于激光能见度仪。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等技术进行结合，通过对视频图像信号的分析处理，来建立视频图像与实际场景之间的联系。根据视频图像的特征变化，间接计算出能见度数值。

现有的基于视频图像的能见度检测方法，由于是间接计算的缘故，对于精确估计能见度还存在一定的困难。现有的方法中，大多数采用的方法是只选取少量视频、截取图像中的某些固有特征，基于 Koschmieder 定律进行估计。由于没有充分使用视频的连续信息的属性，所以估计精度受限，有较大的提升空间。

因为在高速公路行驶和飞机飞行的场景下，能见度是 2000 米还是 3000 米对交通影响并不大。一般在恶劣天气，尤其是大雾的条件下需要准确估计当前以及未来的能见度情况。所以对于大雾的演化规律的研究显得更为重要。

为了对不同大雾情况下的能见度进行估计并预测大雾的消散时间，我们需要根据实际监测的环境数据以及同一时段的视频图像信息来建立环境影响因素与能见度之间的关系，并结合视频数据信息构建深度学习模型来准确预测未来一段实际的能见度趋势。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。

1.2 问题重述

问题一：根据题目所提供的能见度与地面气象观测（温度、湿度和风速）数据，建立描述能见度与地面气象因素的关系模型，并且根据得出的能见度与气象因素关系导出具体关系表达式。

问题二：根据题目所提供的某机场视频数据，建立深度学习模型进行能见度的估计，并且利用机场 AMOS 观测的已知能见度数据对估计的能见度精度进行评估。

问题三：基于高速公路某路段监控视频截图数据，建立并实现不依赖于能见度仪观测数据的能见度估计算法，绘制监控视频截图所提供的时间段内高速公路能见度随时间变化的曲线。（提示：在有雾的情况下可以估计视频中物体的景深，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）

问题四：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加重或减弱的趋势，以及预测大雾在何时会散去，即何时达到指定的能见度（比如 MOR=150m）。

二：模型假设与符号说明

2.1 模型的基本假设

根据实际情况中能见度与大雾关系和文中所给条件，本文做出如下假设：

（1） 假设实际场景中的亮度与图像像素强度之间的转换符合线性关系；
（2） 假设题设给出的视频与图像是在非极端天气中拍摄；
（3） 假设空气为均质；
（4） 假设题目所给数据具有可靠性和合理性；
（5） 假设摄像机焦距为一定，且相机摆放高度为符合一般标准；

2.2 基本符号说明

表 2 - 1 符号说明

符号	说明
RVR, MOR	两项能见度指标
$PAINS$	站点气压
QNH	修正海平面气压
QEF	飞机着陆地区最高点气压
$TEMP$	温度
RH	相对湿度
$WS2A, WD2A, CW2A$	风速相关指标(
F_h	预训练模型提取的图像高层特征
$F_0, F_1 \cdots F_{511}$	图像特征向量 512 维特征
V	最终能见度估计值
R_v	随机森林模型能见度估计值
L_r	LSTM 神经网络残差估计值
B_0, L_0	目标物的固有亮度
B_0', L_b	背景的固有亮度
K	亮度的对比值
F	观测的光照强度
F_0	入射的光照强度
σ	衰减系数
ε	视觉对比阈值
$L(v)$	图像路面区域第v行的亮度
S	天空亮度
f	摄像机焦距
φ	旋转角度

三：总体技术流程图

本文各问题解决方案流程图如下：

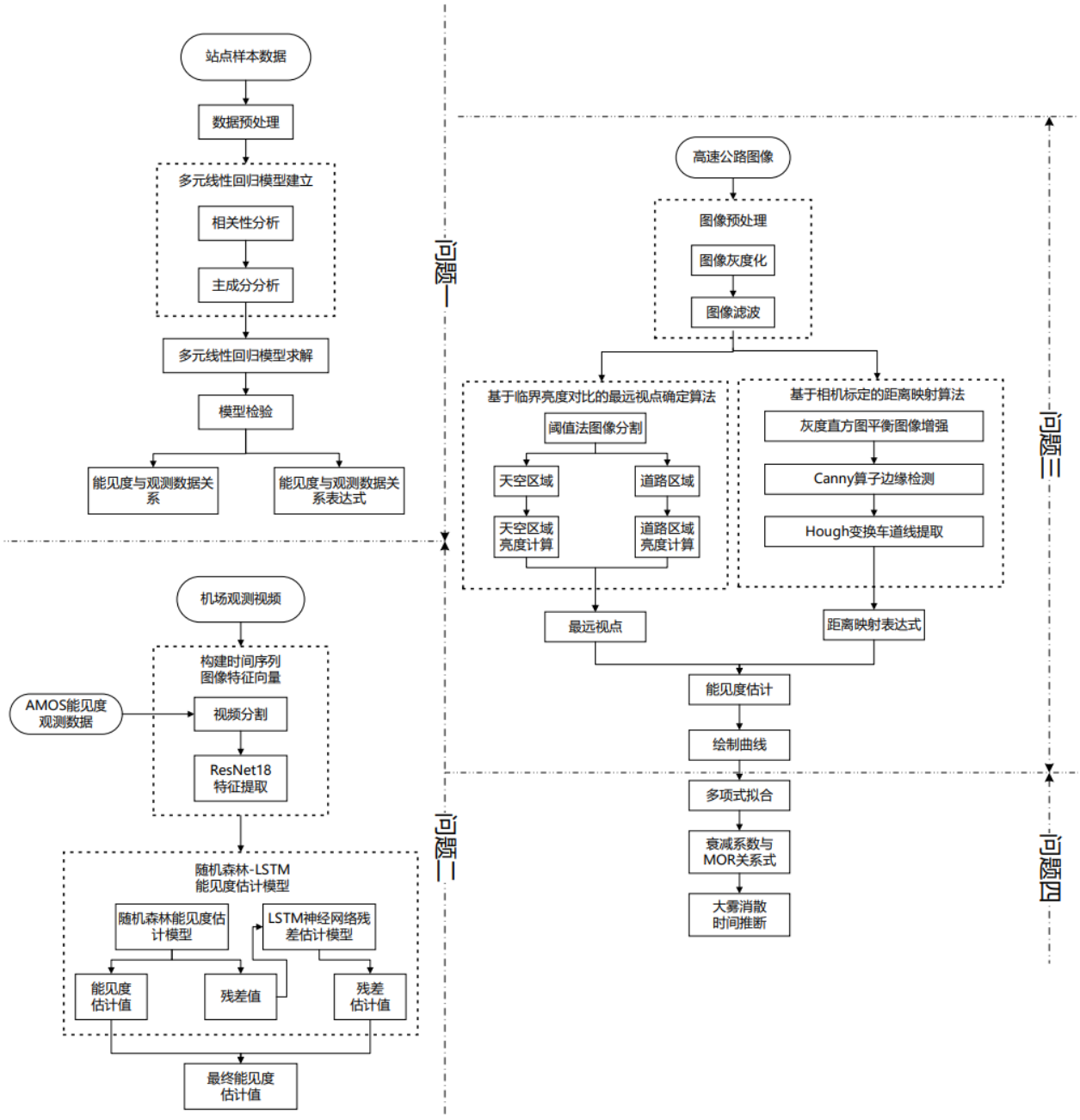

图 3 - 1 总体技术流程图

四：问题一的模型建立与求解

4.1 问题描述与分析

问题一分为两个小问，第一小问需要根据题目所提供的能见度与地面气象观测（温度、湿度和风速等）数据，来建立用于描述能见度与其他环境因素的关系模型。第二小问需要根据得出的能见度与环境因素关系导出准确的关系表达式。

根据题目所提出要求和地面观测数据，问题一解决流程如图 4-1 所示。问题一提供了两组分别在 2019 年、2020 年的某一时间段观测的气象数据。其中包括站点气压、飞机着陆地区最高点气压、修正海平面气压、温度、风速等信息。气象数据为时间序列数据，且数据采样密度在分钟级别。首先对题目所给出的能见度与环境因素的数据进行预处理，并根据清洗后的数据进行描述性统计分析。接着对数据进行归一化处理，以消除数据量纲不同而带来的影响。由于所提供的地面气象指标多达 9 种，首先通过 Pearson 相关系数对每一个变量之间的相关性作出统计。为了简化最后得出的能见度与环境因素的关系式子，采用主成分分析法对气象数据进行降维处理。得到 5 个主成分来替代原有的 9 种指标。

根据得到的主成分，建立多元线性回归模型，基于该模型进行量化，解释能见度与地面气象数据及其相关因素之间的关系。最后根据多元线性回归模型导出能见度与主成分之间的关系式，并对其相关指标进行验证。

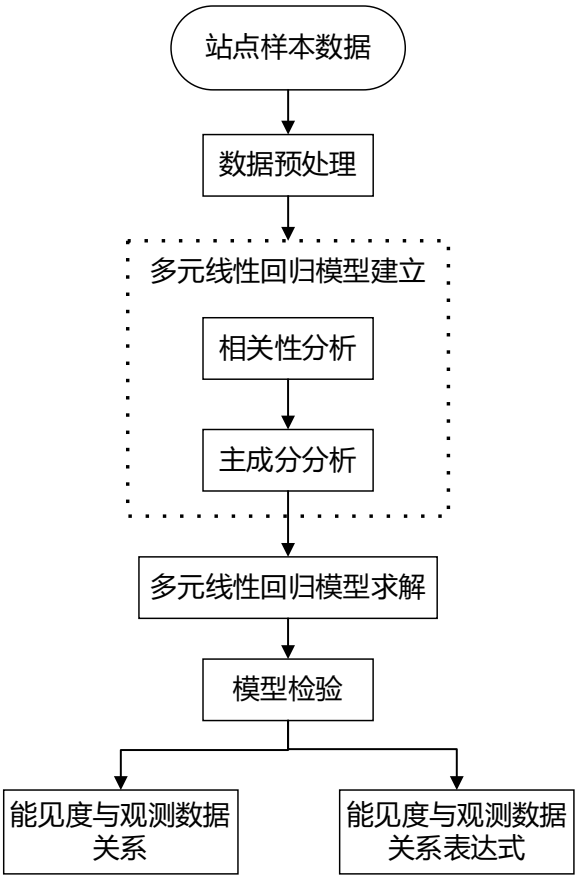

图 4 - 1 问题一流程图

4.2 数据预处理

问题一提供的数据有 2019 年和 2020 年中两个时间段内的两项能见度指标（RVR_1A、MOR_1A），以及站点气压(PAINS）、修正海平面气压(QNH)、飞机着陆地区最高点气压（QFE）、温度(TEMP)、相对湿度(RH)、3 项风速相关指标(WS2A、WD2A、CW2A)。其中对于指标 PAINS、QNH、TEMP、RH、DEWPOINT 每隔 1 分钟进行了一次采集，共持续 24 小时。对于指标 RVR_1A、MOR_1A、LIGHT_S、WS2A、WD2A、CW2A 每隔 15 秒进行一次采集，持续 24 小时。所以第一组数据采集密度是第一组的四倍，且第二组数据存在部分数据丢失的情况。

针对数据上存在的问题，首先通过拉格朗日插值法将第二组数据时间轴上缺失的数据进行补齐。插值公式如下：

$$Ln(x)=\sum_{i=0}^{n} f(x_i)l_i(x) \tag{4-1}$$

其中，$l_i(x)\ (i=0,1,2...,n)$ 是插值基函数，即$l_i(x)=\prod_{j=0}^{n}\frac{x-x_j}{x_i-x_j}$。拉格朗日插值多项式的余项是 $Rn(x)=f(x)-Ln(x)=\frac{1}{(n+1)!}f^{(n+1)}(\xi)\omega(x)$，且其中$\omega(x)=(x-x_0)(x-x_1)\dots(x-x_n)$。选取缺失时间附近相邻的 6 次采样数据代入朗格朗日插值方程，求解缺失值。

接着，需要对第一组数据的数据量进行扩充，使其数据的数量与第二组数据数量相同。数据填充方法采用加权插值方法，即补充的新数据不仅会受到附近数据值的影响，还会受到采集时间间隔长度的影响。插值公式如下，

$$D_i=\frac{\frac{i}{4}*X_n+\frac{(4-i)}{4}*X_{n+1}}{2},(i=1,2,3,4) \tag{4-2}$$

其中，D_i为某 1 分钟内需要插入的三个数据之一，$i=1,2,3,4$分别对应一分钟内的第 15 秒、30 秒、45 秒处的索引。X_n为当前这一分钟零时刻的数据，X_{n+1}为下一分钟的零时刻数据。X前的系数为加权值，使得新值受邻近值的影响更大。通过此方法，得到与第二组数据完全对应的数据组。对清洗后的数据进行描述性统计分析，分别求出 9 种气象勘测数据所对应的最大值、最小值、平均值，如表(4-1）所示。

表 4 - 1 描述性统计表

影响因素	最大值	最小值	平均值
PAINS	1023.3	1016.5	1019.275
QNH	1024.85	1018.04	1020.818
QFE	1023.28	1016.48	1019.255
TEMP	16.3	7.4	12.29646
RH	100	59	81.69071
DEWPOINT	10.91	7.06	9.033024
WS2A	6.8525	0.4325	3.467547
WD2A	354.25	2.75	97.77947
CW2A	6.39	-0.9725	2.215449

从表中可以看出 PAINS、QNH、QFE 三值的最大值、最小值以及平均值变化并不明显。反映到实际就是在一天时间内，当地环境气压变化不大，且三种不同的气压指标的最小值到最大值的变化幅度相近。由 TEMP 值可知一年之内昼夜温差最大达到 8.9 摄氏度，平均温度为 12.3 摄氏度。对于三个风速指标，一天之内风速的最大值和最小值以及风向的最大值和最小值变化都比较大，可以看出当地的水平风速和风向这两个指标的变化较为明显。

最后，由于每一项指标的单位量纲不同，对后续计算与分析会产生不可忽视的影响，所以需要通过归一化的方式来消除不同指标的量纲。利用公式(4-3), 将数据变化到 0-1 之间，其中w_i表示归一化后结果，k序列最大数值。归一化结果如表（4-2）所示，

$$w_i = \frac{w_i - \min(w_i)}{\max(w_i) - \min(w_i)}, (i = 1,2, \dots, k) \qquad (4-3)$$

表 4 - 2 归一化数据结果

序列	PAINS	QNH	QFE	TEMP	RH	DEW PO-INT	WS2A	WD2A	CW2A	RVR1A	MOR1A
1	0.537	0.538	0.537	0.153	0.769	0.374	0.319	0.182	0.302	1.000	0.303
2	0.550	0.550	0.550	0.161	0.769	0.385	0.322	0.171	0.285	1.000	0.310
3	0.550	0.550	0.550	0.161	0.754	0.365	0.323	0.145	0.245	1.000	0.310
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1438	0.975	0.975	0.975	0.185	1.000	0.684	0.377	0.992	0.007	0.004	0.005
1439	0.987	0.987	0.987	0.194	1.000	0.695	0.383	0.989	0.000	0.009	0.005
1440	1.000	1.000	1.000	0.194	1.000	0.695	0.391	0.996	0.001	0.009	0.005

4.3 多元线性回归模型建立

根据题目所给信息，可以看出能见度数据和环境因素数据在时间层面上从黑夜跨度到白天。并且在黑夜期间和白天期间都存在强光照和弱光照的时间段，强光照和弱光照所对应的 LIGHT 值分别为 100 和 10。由于白天、黑夜，以及强光照、弱光照这四个条件的不同，会对能见度变化趋势造成比较大的差异。故为了降低分析难度，简化问题结构，将所提供的数据分类到以下四个板块之中，分别是黑夜强光照、黑夜弱光照、白天强光照、白天弱光照。根据中国气象网所提供的参考标准，定义北京时间 8:00-20:00 为白天，20:00-8:00 为黑夜。采取分块的方法，可以对 4 种不同环境下的能见度进行更加精确有针对性的多元回归分析。

当能见度指标不为定值的时期（四种不同条件下的定值时间段不同），设自变量$x_1, x_2, \cdots, x_5$的观测值$x_{1j}, x_{2j}, \cdots, x_{5j}$，及因变量$y$对应的观测值$y_i$满足关系式：

$$y_i = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \beta_3 x_{3j} + \beta_4 x_{4j} + \beta_5 x_{5j}, (j = 1,2, \cdots, p)$$

对主成分进行最小二乘法的参数估计。在对环境气象因素做主成分分析之前，首先进行相关系数分析。

4.3.1 Pearson 相关系数

Pearson 相关系数用来衡量两个变量之间相关性的大小，也就是衡量定距变量间的线性关系。对于两呈现线性关系的变量，通常可采用 Pearson 相关系数刻画二者的相关程度。其计算公式如下：

$$r=\frac{\sum_{k=1}^{n}(X_k-\bar{X})\,(Y_k-\bar{Y})}{\sqrt{\sum_{k=1}^{n}(X_k-\bar{X})^2}\cdot\sqrt{\sum_{k=1}^{n}(Y_k-\bar{Y})^2}} \qquad (4-4)$$

式中r表示相关系数，n为样本个数，X_k与Y_k分别表示第 i 个样本的两个不同的属性值。当$r=-1$时，两变量呈负相关性。$r=1$时，两变量呈正相关性。也就是当相关系数越接近±1时，两变量具有越强的相关性。由公式可计算出每一个环节因素之间的 Pearson 相关系数，其系数矩阵P如公式所示。

$$P=\begin{bmatrix} 1.000 & 1.000 & 1.000 & 0.455 & -0.601 & -0.313 & 0.745 & 0.281 & 0.739 \\ 1.000 & 1.000 & 1.000 & 0.455 & -0.602 & -0.314 & 0.745 & 0.281 & 0.740 \\ 1.000 & 1.000 & 1.000 & 0.455 & -0.601 & -0.313 & 0.745 & 0.281 & 0.739 \\ 0.455 & 0.455 & 0.455 & 1.000 & -0.947 & 0.205 & 0.702 & 0.281 & 0.686 \\ -0.601 & -0.602 & -0.601 & -0.947 & 1.000 & 0.112 & -0.744 & -0.294 & -0.719 \\ -0.313 & -0.314 & -0.313 & 0.205 & 0.112 & 1.000 & -0.029 & -0.027 & -0.030 \\ 0.745 & 0.745 & 0.745 & 0.702 & -0.744 & -0.029 & 1.000 & 0.267 & 0.901 \\ 0.281 & 0.281 & 0.281 & 0.281 & -0.294 & -0.027 & 0.267 & 1.000 & 0.401 \\ 0.739 & 0.740 & 0.739 & 0.686 & -0.719 & -0.030 & 0.901 & 0.401 & 1.000 \end{bmatrix}$$

根据所求得的每一个因素之间的 Pearson 相关系数可以绘制出如下颜色深度随相关系数变化的热力图。

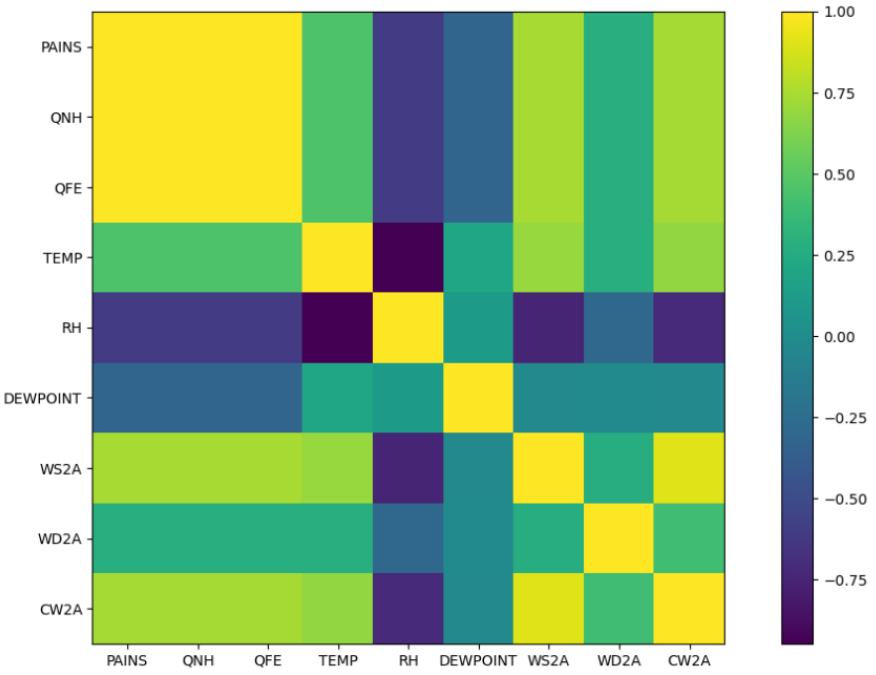

图 4 - 2 Pearson 相关系数热力图

如图（4-2）所示，图中纯黄色部分表示两者 Pearson 相关系数为 1，纯黑色表示两者 Pearson 相关系数为-1。中间颜色由黄色向黑色渐变，其相对应的相关系数的数值也越小。可以看出，2 分钟平均风速、2 分钟平均垂直风速和三种气压之间呈黄色，即是有着较高正相关性， Pearson 相关系数为 0.74。温度和三种气压之间呈浅色，呈现正相关性，Pearson 相关系数为 0.45。相对湿度与三种气压还有温度之间呈深色，呈现负相关性，Pearson 相关系数在-0.74 附近。也表明变量间存在一定的信息重叠。

若直接用这些指标数据进行多元回归分析，则会出现多重共线性的问题。所以需要先将这些指标进行主成分分析，利用各个主成分进行回归。

4.3.2 主成分分析

由题目所提供数据信息，并结合相关参考资料，可以推断能见度的高低与许多环境指标息息相关，例如气压、温度、风速、垂直风速等。我们选取题目中所提供的本站气压(PAINS)、飞机着陆地区最高点气压(QFE)、修正海平面气压(QNH)、温度(TEMP)、相对湿度(RH)、露点温度(DEWPOINT)、2 分钟平均风速(WS2A)、2 分钟平均风向(WD2A)、2 分钟平均垂直风速(CW2A)，共 9 个影响因素。以上因素之间具有一定联系，但并非每一项因素都对能见度的高低有显著影响。引入主成分分析不仅可以降低分析问题的复杂性，需要对其进行降维处理。此时主成分分析法可以用来确定影响能见度的主要因素或指标。

主成分分析的主要步骤如下：

(1) 计算相关系数矩阵

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix} \tag{4-5}$$

公式(4-5)中，$r_{ij}(i,j = 1,2,\dots,p)$为上一小节中所求得的两变量$X_k$与$X_k$之间的 Pearson 相关系数。所以矩阵$R$为标准化样本的协方差矩阵。将数据代入式（4-5）中，求得关系系数矩阵。

(2) 计算矩阵R的特征值与特征向量

首先解出特征方程$|\lambda * i - R| = 0$求得特征值$\lambda_i(i = 1,2,\dots,p)$，并使其按照从大至小的顺序排列，即$\lambda_1 \geq \lambda_1 \geq,\dots,\geq \lambda_p \geq 0$；接着求出与每一个特征值相对应的特征向量$e_i(i = 1,2,\dots,p)$。

(3) 计算主成分贡献率与累计贡献率

并由 Pearson 相关系数矩阵计算得出特征值和特征向量后，求解各个主成分的贡献率C_n与累计贡献率C_m，计算公式如下：

$$C_n = \frac{\lambda_i}{\sum_{k=1}^{p} \lambda_k}(i = 1,2,\dots,p) \tag{4-6}$$

$$C_m = \frac{\sum_{k=1}^{i} \lambda_k}{\sum_{k=1}^{p} \lambda_k}(i = 1,2,\dots,p) \tag{4-7}$$

经计算可得所有主成分的特征值、个体贡献率、累加贡献率，如表所示。由表可知，前三项主成分的累计贡献率达到 92.341%，前五项主成分的累计贡献率达到 99%，为了能在减少维度的同时，最大限度地表征原有的函数关系。故选择前五项作为主成分，求出第一、二、三、四主成分即可代替 9 项指标。

表 4 - 3 主成分参数表

主成分	特征值	贡献率(%)	累计贡献率(%)
PC_1	5.5632	73.647	73.647
PC_2	1.4863	12.027	85.674
PC_3	0.8693	6.667	92.341
PC_4	0.6744	4.260	96.602
PC_5	0.3195	2.493	99.095
PC_6	0.0867	0.813	99.909
PC_7	0.0006	0.090	99.999
PC_8	0.0000	0.000	99.999
PC_9	0.0000	0.000	99.999

根据表数据，做出图(4-3)，图 a 为每一个主成分所对应的贡献率，图 b 表示前i项主成分所对应的累计贡献率。从图中可以更直观地得出，当取前五项作为主成分项时，贡献率出于 99%接近，可以很好的表征原有因素对能见度结果的关系。

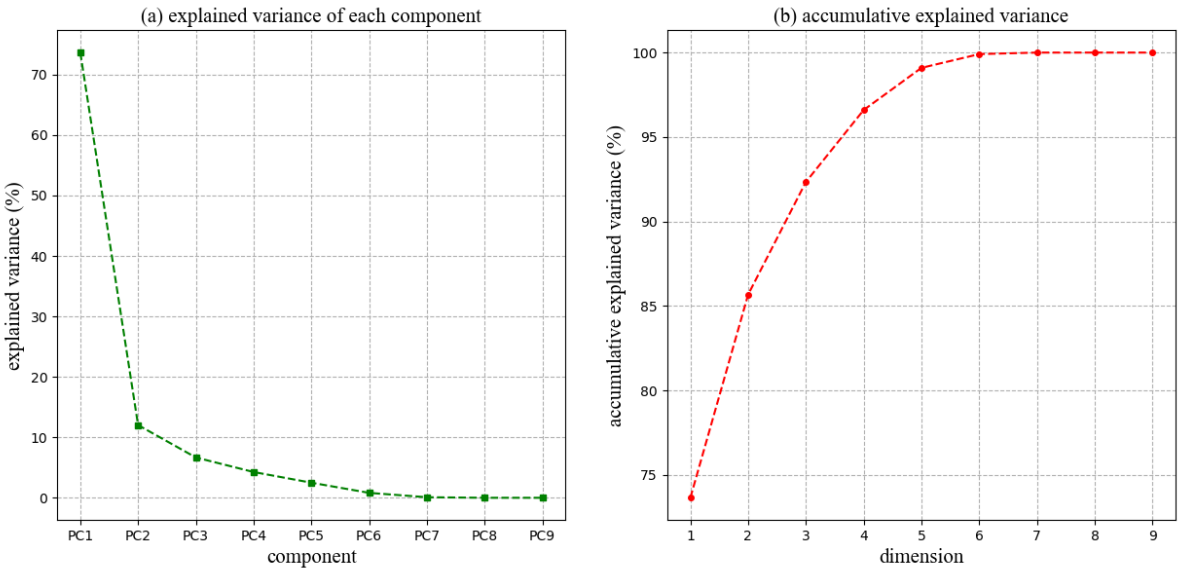

图 4 - 3 主成分贡献率

表 4 - 4 主成分特征向量表

主成分	特征向量
PC_1	$(0.388,0.388,0.388,0.312,-0.353,-0.081,0.381,0.170,0.382)^T$
PC_2	$(0.273,0.273,0.273,-0.484,0.265,-0.648,-0.135,-0.1192,-0.040)^T$
PC_3	$(0.051,0.051,0.051,0.124,-0.106,0.087,0.135,-0.969,-0.040)^T$
PC_4	$(0.219,0.217,0.219,-0.328,0.509,0.683,0.095,0.007,0.120)^T$
PC_5	$(0.241,0.240,0.241,0.268,-0.221,0.241,-0.556,0.065,-0.579)^T$

$$PC_i = \alpha_i \cdot X_i (i = 1,2,\dots,p) \qquad (4-8)$$

第一主成分PC_1对 PAINS、QNH、QFE06、WS2A、CW2A 为中等程度的正载荷；在 RH 上为中等程度的负载号。故第一主成分主要受气压、风速的正向影响，以及相对湿度的负向影响，其余因素对其几乎没有影响。

第二主成分PC_2对 PAINS、QNH、QFE06 为中低程度的正载荷；而在 TEMP、DEWPOINT 上呈现中等程度的负载荷。故第二主成分主要受温度指标的负向影响，以及中低程度的气压因素影响。

第三主成分PC_3对 WD2A 为高程度的负载荷；而其余因素对其几乎没有载荷。故第三主成分主要受到风向的影响，为风向成分。

第四主成分PC_4对 RH、DEWPOINT 为中程度的正载荷；在 PAINS、QNH、QFE06 上有低程度的正载荷。所以PC_4主要受到相对湿度和露点温度的正向影响，受到气压的轻微影响，其余因素对其几乎没有影响。

第五主成分PC_5对 WS2A、CW2A 为中程度的负载荷；在 PAINS、QNH、QFE06 上有低程度的正载荷。所以PC_5主要受到 2 分钟平均风速和 2 分钟平均垂直风速的负向影响，受到气压低程度的正向影响。

4.4 多元线性回归模型求解

如图 4-4 所示，(a)、(b)、(c)、(d)分别表示白天弱光照条件、黑夜弱光照条件、白天强光照条件、白天弱光照条件，四种不同情境下 2019、2020 年的两种能见度值随时间的变化趋势。（a）中只有 2019、2020 年的 MOR1A 值随时间发生变化，在一段时间内从最大值开始下降。其余能见度指标没有改变。(b)中四项能见度指标都在某一时间段内呈现下降趋势。(c)中为 2020 年白天时间段的能见度数据，RVR1A 值没有随之间变化，MOR1A 在其中一个时段呈现上升趋势。(d)中描述的是 2019 年黑夜时间段的能见度数据，期间 MOR1A 值并未变化，在一段时间内 RVR1A 呈现下降趋势。

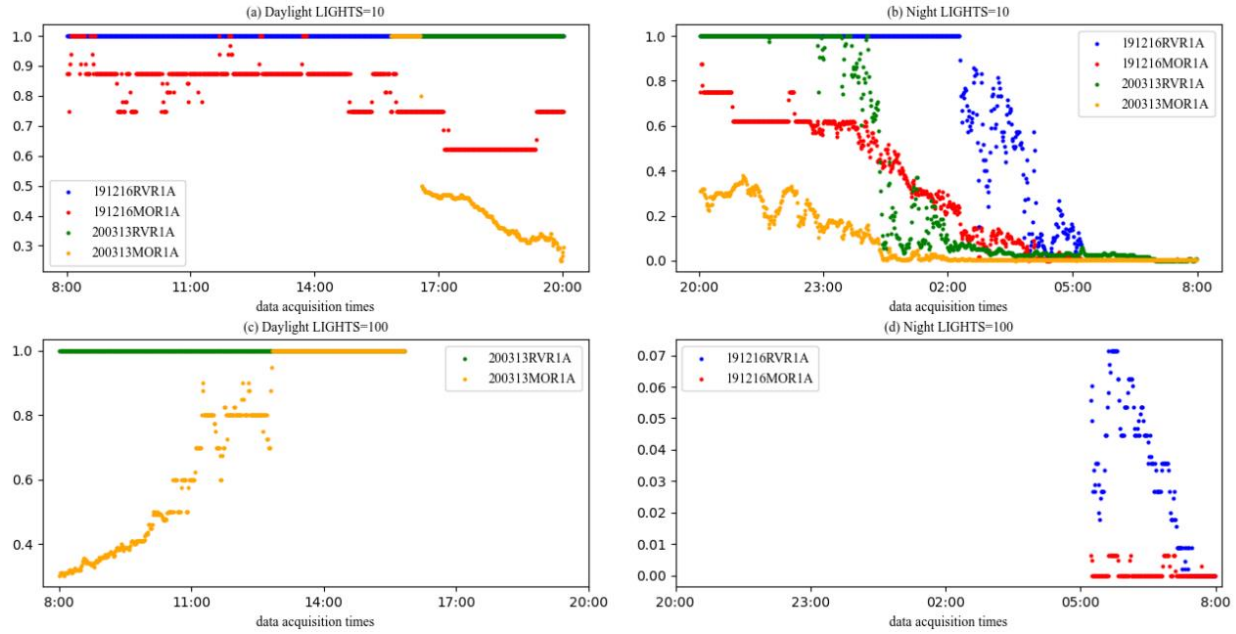

图 4 - 4 能见度时间序列散点图

(1)白天、强光照条件

利用各主成分数据进行多元线性回归模型的求解。将全部主成分引入多元回归线性模型，并对其进行最小二乘法的参数估计。通过 Stata 软件对主成分数据进行回归分析，得到如下表所示输出结果。

表 4 - 5 MOR1A 白天强光条件统计指标

主成分	Coef.	Std. Error	t	P>| t |	统计项	数值
PC1	0.4845	0.0103	47.00	0.000	SSR	32.205
PC2	-0.1791	0.0288	-6.21	0.000	SST	33.207
PC3	-0.2907	0.0213	-13.61	0.000	F	2984.16
PC4	-0.0040	0.0249	-0.16	0.871	Prob>F	0.0000
PC5	0.1643	0.0278	5.90	0.000	adj R^2	0.970

可以看出主成分 PC4 的系数对应的 P 值大于显著性水平 0.05，则接受原假设：该系数等于 0 的原假设，即主成分 PC4 没有通过系数显著性检验，说明 PC4 对于 MOR1A 没有显著性影响。故将 PC4 进行剔除。再次进行显著性检验后，剩下的主成分均通过检验。由于引入的自变量越多，拟合优度会越大。所以更倾向于使用调整后的拟合优度，白天强光照条件下$R^2_{adjusted} = 0.97$，非常接近 1。故最终模型包含了 4 个主成分。求得关系式如下：

$$y_1 = 0.4845 * PC1 - 0.1791 * PC2 - 0.2907 * PC3 + 0.1643 * PC5 + 0.40 \quad (4-9)$$

(2)白天、弱光照条件

表 4 - 6 MOR1A 白天弱光照条件统计指标

主成分	Coef.	Std. Error	t	P>| t |	统计项	数值
PC1	0.0306	.0239767	12.77	0.000	SSR	14.003
PC2	-0.5156	.027673	-18.63	0.000	SST	14.212
PC3	-0.3799	.0366274	-10.37	0.000	F	3289.15
PC4	0.1662	.0262708	6.33	0.000	Prob>F	0.0000
PC5	0.4683	.064841	10.22	0.000	adj R^2	0.9850

与(1)中同理，判断出 5 项主成分都通过显著性检验，可以求得 MOR1A 白天弱光照条件下的 MOR1A 关系式为：

$$y_2 = 0.0306 * PC1 - 0.5156 * PC2 - 0.3799 * PC3 + 0.1662 * \mathrm{PC4} + 0.4683 * PC5 + 0.484 \quad (4-10)$$

(3) 黑夜、强光照条件

与(1)中同理，只有 PC1、PC4 通过显著性检验。可以求得 RVR1A 黑夜强光照条件下的关系式为：

$$y_3 = 0.495 * PC1 - 0.1282 * \mathrm{PC4} + 0.464 \quad (4-11)$$

对于 MOR1A 只有 PC1、PC3、PC4 通过显著性检验。可以求得 MOR1A 黑夜强光照条件下的关系式为：

$$y_4 = 4.2205 * PC1 - 0.5335 * PC3 - 0.1282 * \mathrm{PC4} + 0.464 \quad (4-12)$$

(4) 黑夜、弱光照条件

与(1)中同理，黑夜、弱光照条件下 RVR1A 的 5 个主成分均通过显著性检验。且其他指标都在可接受范围内，故解得其关系式为

$$y_5 = -0.280 * PC1 + 1.7417 * PC2 - 0.4526 * PC3 + 0.219 * \mathrm{PC4} - 0.8391 * PC5 + 0.18 \quad (4-13)$$

MOR1A 的 5 个主成分均通过显著性检验。且其他指标都在可接受范围内，故解得其

关系式为

$$y_6 = 0.0461 * PC1 - 0.0252 * PC2 - 0.0651 * PC3 \\ +0.090 * \mathrm{PC4} - 0.344 * PC5 + 0.364 \quad (4-14)$$

4.5 多元线性回归模型总结

由于y是经过归一化后的结果，需要通过公式，求得实际单位对应的表达式，求得结果如式。

$$Y_{MOR1A} = (MOR1A_{max} - MOR1A_{min}) * y + MOR1A_{min} \quad (4-15)$$

$$Y_{RVR1A} = (RVR1A_{max} - RVR1A_{min}) * y + RVR1A_{min} \quad (4-16)$$

故对于能见度与主成分关系式如下所示：

(1) 白天强光照条件

$$\begin{cases} Y_{MOR1A} = (MOR1A_{max} - MOR1A_{min}) * y_1 + MOR1A_{min} \\ y_1 = 0.4845 * PC1 - 0.1791 * PC2 - 0.2907 * PC3 + 0.1643 * PC5 + 0.40 \end{cases} \quad (4-17)$$

由于数据显示 RVR1A 在白天强光条件下恒等于 3000，故

$$Y_{RVR1A} = 3000 \quad (4-18)$$

(2)白天弱光照条件

$$\begin{cases} Y_{MOR1A} = (MOR1A_{max} - MOR1A_{min}) * y_2 + MOR1A_{min} \\ y_2 = 0.0306 * PC1 - 0.5156 * PC2 - 0.3799 * PC3 \\ +0.1662 * \mathrm{PC4} + 0.4683 * PC5 + 0.484 \end{cases} \quad (4-19)$$

由于数据显示 RVR1A 在白天强光条件下恒等于 3000，故

$$Y_{RVR1A} = 3000 \quad (4-20)$$

(3)黑夜强光照条件

$$\begin{cases} Y_{MOR1A} = (MOR1A_{max} - MOR1A_{min}) * y_3 + MOR1A_{min} \\ y_3 = 0.495 * PC1 - 0.1282 * \mathrm{PC4} + 0.464 \end{cases} \quad (4-21)$$

$$\begin{cases} Y_{RVR1A} = (RVR1A_{max} - RVR1A_{min}) * y_4 + RVR1A_{min} \\ y_4 = 4.2205 * PC1 - 0.5335 * PC3 - 0.1282 * \mathrm{PC4} + 0.464 \end{cases} \quad (4-22)$$

(4)黑夜弱光照条件

$$\begin{cases} Y_{MOR1A} = (MOR1A_{max} - MOR1A_{min}) * y_5 + MOR1A_{min} \\ y_5 = -0.280 * PC1 + 1.7417 * PC2 - 0.4526 * PC3 \\ +0.219 * \mathrm{PC4} - 0.8391 * PC5 + 0.18 \end{cases} \quad (4-23)$$

$$\begin{cases} Y_{RVR1A} = (RVR1A_{max} - RVR1A_{min}) * y_6 + RVR1A_{min} \\ y_6 = 0.0461 * PC1 - 0.0252 * PC2 - 0.0651 * PC3 \\ +0.090 * \mathrm{PC4} - 0.344 * PC5 + 0.364 \end{cases} \quad (4-24)$$

五：问题二的模型建立与求解

5.1 问题描述与分析

针对问题二，要求根据题目提供的某机场视频数据，建立深度学习模型进行能见度的估计，并且利用机场 AMOS 观测的已知能见度数据对估计的能见度精度进行评估。在根据机场观测视频进行分析并估计能见度的过程中，提取视频中的特征数据信息是最为基础且关键性的一步。目前已有的方法只选取了少量视频、截取图像中的某些固有特征，未能充分利用视频的连续信息的估计精度是不高的。因此，本文将机场观测视频使用 OpenCV 分割为时间序列的图像集合，通过这种图像集合的时间序列保留机场观测视频数据的连续信息。传统图像特征提取方法是基于图像的颜色特征、纹理特征、形状特征以及空间特征等信息特征，虽然特征多样，但是对于要用深度学习算法分析的模型来说，数据量较小，容易导致模型的过拟合。为此，本文提出迁移学习的方法提取多维度机场观测图像特征，利用在 ImageNet 中表现优异的 ResNet18 神经网络预训练模型对机场观测图像集合提取特征得到 512 维关于图像特征的时间序列数据。获得多维度基于时间序列的图像特征之后，本文选择对超参数调优要求不高的随机森林深度学习算法对其进行和能见度回归分析。为进一步提高模型对能见度估计的精度，本文采用 LSTM 长短期记忆神经网络对经过随机森林深度学习算法能见度估计的残差进行回归分析，进一步增强对视频连续信息的分析。最终获得的能见度估计值为随机森林能见度估计值与 LSTM 残差估计值之和。

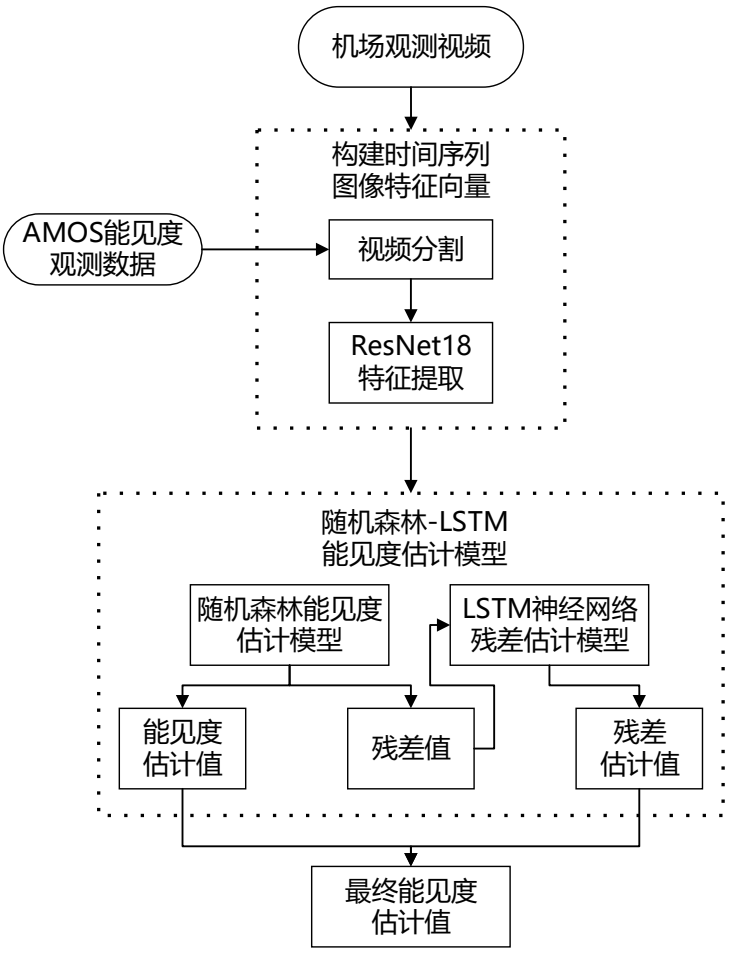

图 5 - 1 问题二流程图

5.2 基于迁移学习的图像特征提取

为了保留机场观测视频的连续信息，本文将该视频通过 OpenCV 进行视频分割，得到一个基于视频时间序列的图像视频集合，在此基础上对图像时间序列进行分析，以此建立精度更高的能见度估计模型。图像是多维数据很难直接对其进行分析，所以需要对其进行特征提取，构建特征向量。图像中包含了许多信息，主要包含颜色特征、纹理特征、形状特征以及空间特征等信息特征，虽然图像的特征多样，但是由于机场观测视频对应的能见度数据限制，可用的数据规模较小，无法有效发挥深度学习算法分析的模型的性能，并且容易导致能见度估计模型的过拟合和泛化能力的减弱。为此，本文提出使用迁移学习提取多维度机场观测图像特征，利用在 ImageNet 中表现优异的 ResNet18 神经网络预训练模型对机场观测图像集合提取特征得到 512 维关于图像特征的时间序列数据。

ResNet（Residual Neural Network）由微软研究院的 Kaiming He 等四名华人提出，作为残差网络，其基本思想是在网络中增加了直连通道，通过不断加深网络层数以丰富特征的层次，来提高分类精度。在 ResNet 网络结构中有两种残差模块，一种是以两个 3X3 的卷积网络串接在一起作为一个残差模块，另外一种是 1X1、3X3、1X1 的 3 个卷积网络串接在一起作为一个残差模块，其数学表达式分别为：

$$x_{l+1} = x_l + F(x_l, W_l) \qquad (5-1)$$

$$x_{l+1} = h(x_l) + F(x_l, W_l) \qquad (5-2)$$

其中x_l是输入，x_{l+1}是输出、$F(x_l, W_l)$ 即为残差部分。ResNet 有不同的网络层数，比较常用的是 18-layer、50-layer、101-layer 和 152-layer，他们都是由上述的残差模块堆叠在一起实现的。其中 ResNet18 模型较为轻便且性能具佳，本文使用 ResNet18 基于 ImageNet 训练集的预训练模型进行特征提取。在 ResNet18 预训练模型的多层结构中，中间层包含了图像的各种特征，其中前几层主要获取图像的颜色、边缘、纹理等低层特征，而后面的层可以产生用于直接识别、分类的高层特征。

迁移学习可以将预训练模型学习到的知识从原来的领域转移到目标任务上，为了提高 ResNet18 对机场观测图像的泛化能力，本文使用整流线性单元（ReLU）作为 ResNet18 的激活函数，避免其在训练过程中发生梯度消失；为了提高模型泛化能力，采用局部响应归一化来对特征图进行平滑处理。经过一系列模型微调，让 ResNet18 对机场观测图像泛化能力得到提高，最终本文将来自模型中间层的 512 维高层特征作为对应图像的特征，记为F_h

$$F_h = [F_0, F_1, \dots, F_{511}] \qquad (5-3)$$

本文利用 Python 在 Pytorch 环境中实现基于 ResNet18 神经网络预训练模型提取机场观测视频图像的高层特征，该 512 维图像纹理特征数据$[F_0, F_1, \dots, F511]$结果如下表所示。

表 5 - 1 ResNet18 预训练模型图像特征向量表

BeiJingDate	RVR1A	MOR1A	F_0	F_1		F_{510}	F_{511}
2020/3/13 00:01:30	2700	1300	3.698044	0.123245	…	1.516051	1.50217
2020/3/13 00:01:45	2800	1400	3.422071	0.055265	…	1.735844	1.47456
2020/3/13 00:02:00	3000	1500	3.358808	0.051911	…	1.600051	1.397367
2020/3/13 00:02:15	3000	1500	3.184369	0.053498	…	1.485236	1.269373
⋮	⋮	⋮	⋮	⋮	…	⋮	⋮
2020/3/13 05:50:00	175	50	0.524305	0.308521	…	0	0.104601
2020/3/13 05:50:15	175	50	0.664175	0.435253	…	0	0.015007
2020/3/13 05:50:30	175	50	0.675659	0.382431	…	0	0.000324
2020/3/13 05:50:45	175	50	0.537843	0.410633	…	0.006649	0.043148

5.3 随机森林能见度回归分析

在对机场观测视频进行按时间序列分割和 ResNet 预训练模型图像特征提取之后，数据条目有 1410 条，每个条目的特征向量维度为 512。这种数据条目较少而特征维度较大的数据集，随机森林深度学习算法能够发挥优异的性能；另外由于随机森林由多棵决策树集成而来，在普通情况下训练得到的模型精度就高于大多同类深度学习算法，同时随机森林算法在一般超参数调优下就能实现较好训练效果。因此，在针对上述图像特征向量数据集使用随机森林算法，以此提高模型训练效率并保证模型性能。

5.3.1 随机森林深度学习算法原理

随机森林回归模型是一种较新的且高度灵活的深度学习算法，该算法的基本单元是决策树，通过集成学习的思想将多棵 CART 树集成进行回归预测。随机森林算法包含训练集抽取、生成决策树、组合模型和模型验证四个步骤，具体内容如下：

抽取训练集，假设原始样本集中总共有N个样本，每轮从原始样本集中通过 Bootstraping（有放回抽样）的方式抽取 N 个样本，得到一个大小为 N 的训练集。在抽取过程中，可能存在某个样本被重复抽取或未被抽取到的情况发生。设一共进行k轮的训练集抽取，每轮抽取训练集中的样本个数分别为T_1，T_2，…，T_k。

生成决策树，假设训练集样本空间中一共有D个特征属性，从D个特征中随机选择其中的d个特征$(d < D)$组成一个新的特征集，利用新的特征集合生成决策树。设在k轮中一共生成k个决策树，由于该k个决策树在训练集的选择以及特征集的选择上都是随机的，因此这k个决策树之间是相互独立的。

组合模型，由于生成的k个决策树之间相互独立，每个决策树的重要性地位是相同的，因此在将每个决策树组合成模型时，可以认为它们具有相同的权重或者无需考虑它们的权值，对于回归问题，使用所有决策树输出的均值作为最终模型的输出结果。

验证模型，模型的验证需要验证集，由于训练集与特征集选取的随机性，有可能存在部分样本或特征未被选取，因此验证模型时无需额外专门获取验证集，只需选择未被选取过的样本集与特征集数据进行最终模型的验证即可。

基于随机森林算法的以上步骤，本文利用 CART 树建立模型，采用最小均方差原则，针对从机场视频图像中提取的特征A，对每个特征的叶子节点（划分点）s进行划分，完成建模。具体建模过程如下：对于划分特征A，对应的任意划分点s，父节点分裂出两个子节点（假设），两个节点的数据集为D_1和D_2，求出使D_1和D_2各自集合的均方差最小，同时D_1和D_2的均方差之和最小所对应的因素和因素值作为划分点。表达式为：

$$\min_{A,s}\left[\min_{c_1}\sum_{y_i\in D_1(A,s)}(y_i-c_1)^2+\min_{c_2}\sum_{y_i\in D_2(A,s)}(y_i-c_2)^2\right] \qquad (5-4)$$

其中c_1表示第一个节点的均值，c_2表示第二个节点的均值。以上式为原则进行特征划分点确定，最终建立起基于图像提取特征的能见度估计回归模型。

5.3.2 随机森林回归结果与分析

本文采用 Python 实现随机森林深度学习算法，对时间段为 00:00 到 05:59 的机场观测视频分割和预训练模型 ResNet18 图像特征提取得到的 1410 条、512 维的特征向量，分别对跑道视程 RVR 和气象学视程 MOR 两个能见度指标进行回归分析。随机森林算法各参数的设置分别为，树数量设置为 100 棵，决策树深度设置为 10，最小节点分裂所需的最小样

本数量设置为 50，叶子节点最少样本数量 10，分裂所需的最小增益设置为 0.1，列采样设置 sqrt 方法随机选择 sqrt 个特征，行采样比例设置为 0.8，最后设置随机种子保证每次生成的样本集不会变确保实验可重复。

根据题目提供的机场 AMOS 能见度观测值，现有两个衡量能见度值大小的指标跑道视程 RVR 和气象学视程 MOR。本文对 RVR、MOR 两个能见度值与对应时间节点 512 维图像特征向量的时间序列分别通过随机森林深度学习算法进行回归分析，两指标在时间序列上的实际值与预测值如下图所示。由图可以看出，红色能见度估计点与蓝色能见度真实值点大部分重合，总体上随机森林深度算法根据 ResNet18 预训练模型提取的图像特征对 RVR 和 MOR 能见度指标的回归分析效果良好，能够较好的估计能见度。其中图（a）和图（c）展示了 00:00 到 00:59 时段内波动幅度较大的 RVR 和 MOR 的估计，均能较好地估计能见度值。图（b）和图（d）展示了 01:00 到 05:59 时段内 RVR 和 MOR 的能见度估计结果，可以看到总体估计准确，尤其是在 02:00 到 05:59 时段内估计值几乎与真实值相同。

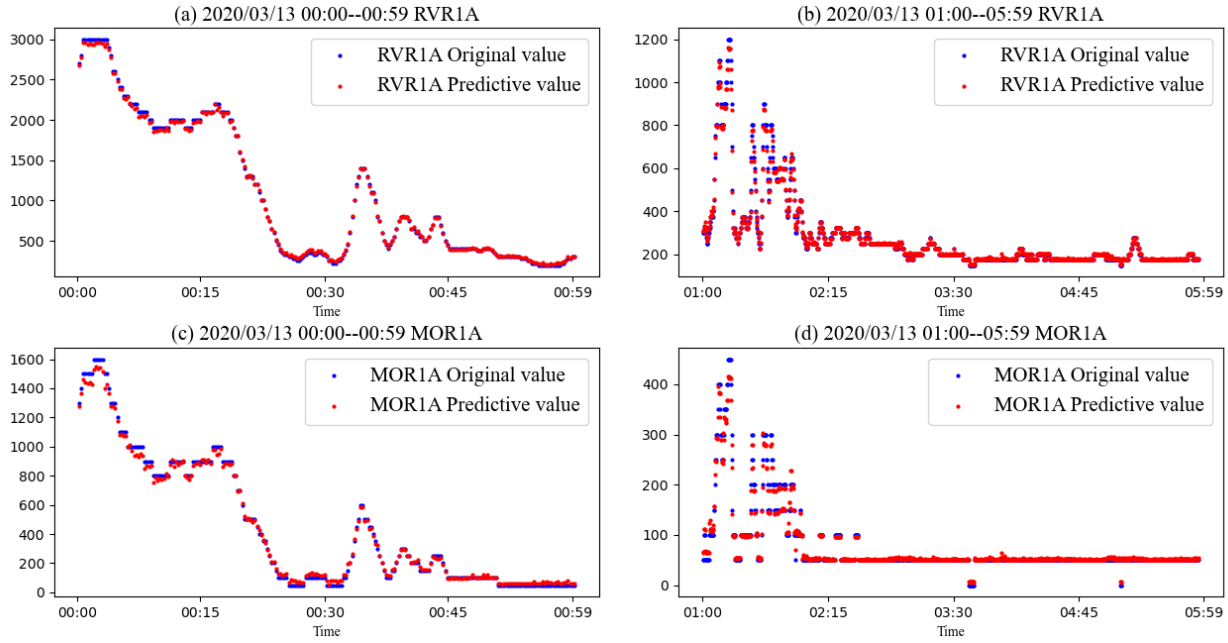

图 5 - 2 随机森林能见度估计模型估计结果图

为了检验随机森林能见度估计模型的可靠性，分析能见度估计值的可靠性、周期性或其他干扰。本文分别对两个能见度指标 RVR1A 和 MOR1A 经过随机森林深度学习算法预测后的残差进行分析。下图展示了 RVR 和 MOR 两种能见度指标各点能见度估计值的残差值与总体残差分布。其中图（a）和图（b）展示了是两个指标的每个能见度估计值对应的残差，可以看出集中在 0 值附近，经计算残差均值为 0。图（c）和图（d）分别是两个指标值能见度估计值残差的分布情况，可以看出两组预测值的残差分布均服从正态分布。由此可以总结基于随机森林深度学习算法的能见度估计模型是可靠稳定的。

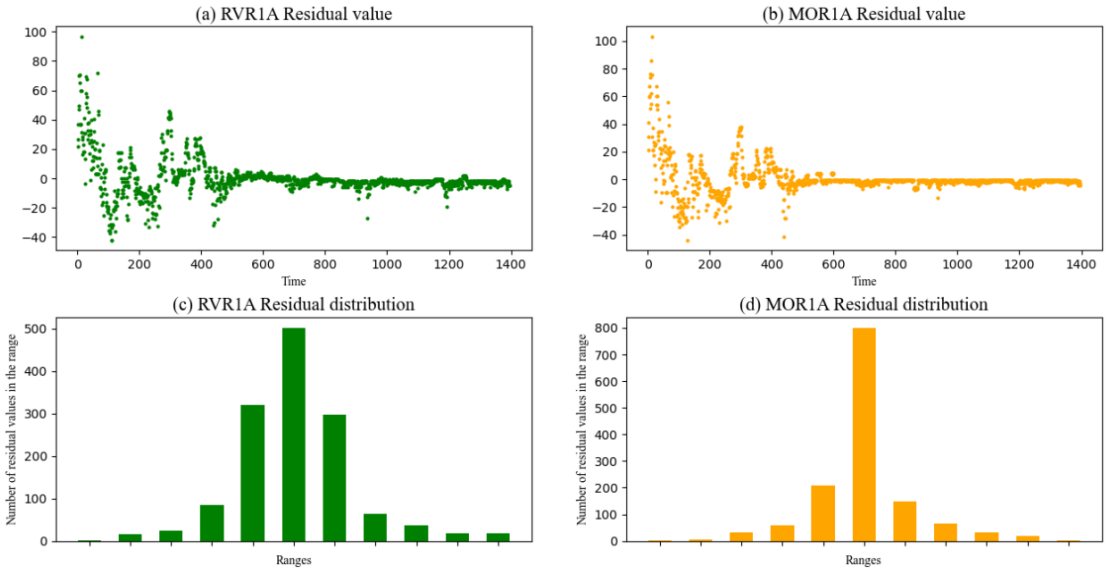

图 5 - 3 随机森林能见度估计模型残差值与分布图

5.4 LSTM 神经网络残差回归模型

在对随机森林深度学习算法的残差分析中得出其残差均值为 0 且残差分布服从正态分布，说明了随机森林能见度估计算法的可靠性与稳定性，同时也说明了该残差具有进一步分析的意义。为了减小随机森林能见度估计模型中能见度估计的误差，进一步提高整个能见度估计模型的估计精度，本文采用 LSTM 长短期记忆神经网络模型对经过随机森林深度学习算法能见度估计的残差部分进行回归分析，将最终的能见度估计值定义为随机森林回归模型的能见度估计值与 LSTM 神经网络残差回归模型的残差估计值之和。

5.4.1 LSTM 长短期神经网络原理

近年来，LSTM 神经网络在时间序列数据分析中得到广泛应用，它的出现极大的解决了长依赖的记忆推测问题，相比普通的 RNN，LSTM 能够在更长的序列中有更好的表现。和一般循环神经网络模型 RNN 一样，LSTM 也是重复链式结构，但是它的重复单元不同于标准 RNN 网络里的单元只有一个网络层，它的内部有四个网络层。LSTM 的核心思想是 LSTM 块状态，用贯穿各个 LSTM 块的水平线表示，该水平线在 LSTM 块中像传送带一样，贯穿整个 LSTM 块却只有很少的分支，这样能保证信息不变的流过整个深层网络。

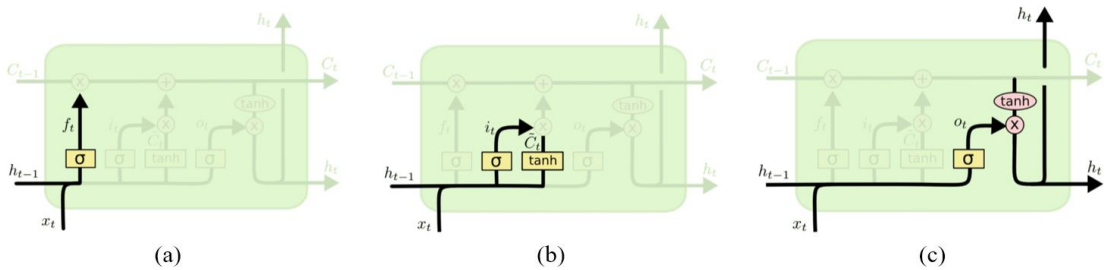

图 5 - 4 LSTM 块的三个组成部分

LSTM 网络能通过一种被称为阀门的结构对 LSTM 块状态进行删除或者添加信息，阀门能够有选择性的决定信息通过。阀门是一个 sigmoid 层和一个点乘操作的组合， sigmoid

层输出 0-1 的值，这代表有多少信息能够流过 sigmoid 层，其中 0 表示都不能通过，1 表示都能通过。一个 LSTM 块里面包含三个阀门来控制 LSTM 块状态，分别称为忘记阀门、输入阀门和输出阀门。

忘记阀门(a)，LSTM 块的忘记阀门就是决定 LSTM 块状态需要丢弃哪些信息。忘记阀门层结构如图 4.13（a）所示，它通过查看h_{t-1}和x_t信息来输出一个 0-1 之间的向量，该向量里面的 0-1 值表示 LSTM 块状态中的哪些信息保留或丢弃多少，其中 0 表示不保留，1 表示都保留。忘记阀门的数学表达式为：

$$f_t = \sigma * \left(W_f * [h_{t-1}, x_t] + b_f\right) \tag{5-5}$$

输入阀门(b)，LSTM 块在忘记阀门丢弃一些信息之后，下一步要决定把哪些信息存储在 LSTM 块状态中，这就是输入阀门的任务，它由两部分组成，分别是决定要更新的值和创造新的候选值向量。输入阀门层如图 4.13（b）所示，首先利用h_{t-1}和x_t通过输入阀门决定更新哪些信息；然后利用h_{t-1}和x_t通过 tanh 层创造新的候选 LSTM 块信息向量$\tilde{C}_t$，这些信息可能会被更新到 LSTM 块保存的信息中。该过程的数学表达式如下所示：

$$i_t = \sigma * (W_i * [h_{t-1}, x_t] + b_i) \tag{5-6}$$

$$\tilde{C}_t = tanh * (W_C * [h_{t-1}, x_t] + b_C) \tag{5-7}$$

在创建新的候选值后，通过忘记阀门选择 LSTM 块保存信息的一部分，再通过输入阀门选择添加候选信息向量$\tilde{C}_t$的一部分得到新的 LSTM 块信息C_t，这一过程就是将就信息C_{t-1}更新为新信息C_t，其数学表达式如下所示：

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5-8}$$

输出阀门(c)，LSTM 块更新完状态后需要根据输入的h_{t-1}和x_t来选择输出的信息状态特征，这里需要将输入经过如图 4.13（c）所示结构输出阀门的 sigmoid 层得到判断条件，然后将 LSTM 块状态经过 tanh 层得到一个-1~1 之间值的向量，该向量与输出阀门得到的判断条件相乘就得到了最终该 LSTM 块的输出。该过程的数学表达式如下所示：

$$o_t = \sigma * (W_o * [h_{t-1}, x_t] + b_o) \tag{5-9}$$

$$h_t = o_t * \tanh(C_t) \tag{5-10}$$

5.4.2 LSTM 神经网络残差回归结果与分析

本文采用 Python 在 Pytoch 环境中实现 LSTM 长短期记忆神经网络，对随机森林能见度估计模型的残差序列进行回归分析。LSTM 预测模型主要分为三层，分别是输入层、隐藏层和输出层。网络的输入层是将训练数据中由 ResNet18 预训练模型提取的特征量和对

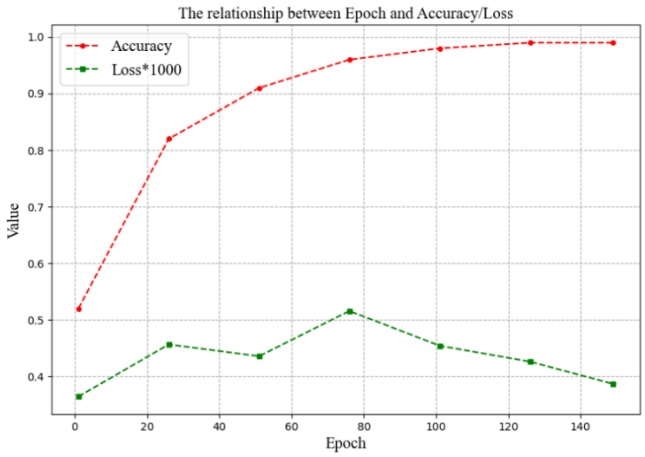

图 5 - 5　LSTM 神经网络 epoch 与 loss、accuracy 关系图

应残差值作为输入，将所有数据输入隐含层，隐含层由 L 个连接的同构 LSTM 细胞构成，经过隐含层后。网络模型结构为单层的 LSTM 层，优化器选择 Adam 算法，模型批量处理长度 epoch 为 150，损失函数 loss 使用均方误差函数，步长 epoch 与损失值和准确率结果如下图所示，可以看出损失值保持在很小的值，准确率 Accuracy 在 100 后收敛。

RVR、MOR 两个能见度指标经过随机森林能见度估计模型的残差在时间序列上的实际值与估计值如下图所示。由图可以看出，在分别展示两个能见度指标 RVR 和 MOR 残差值的图（a）和图（b）中，红色残差估计点与蓝色残差真实值点大部分重合，由此可以得出总体上 LSTM 长短期神经网络根据 ResNet18 预训练模型提取的图像特征对 RVR 和 MOR 能见度残差值的回归分析效果良好，能够较好的估计能见度残差。

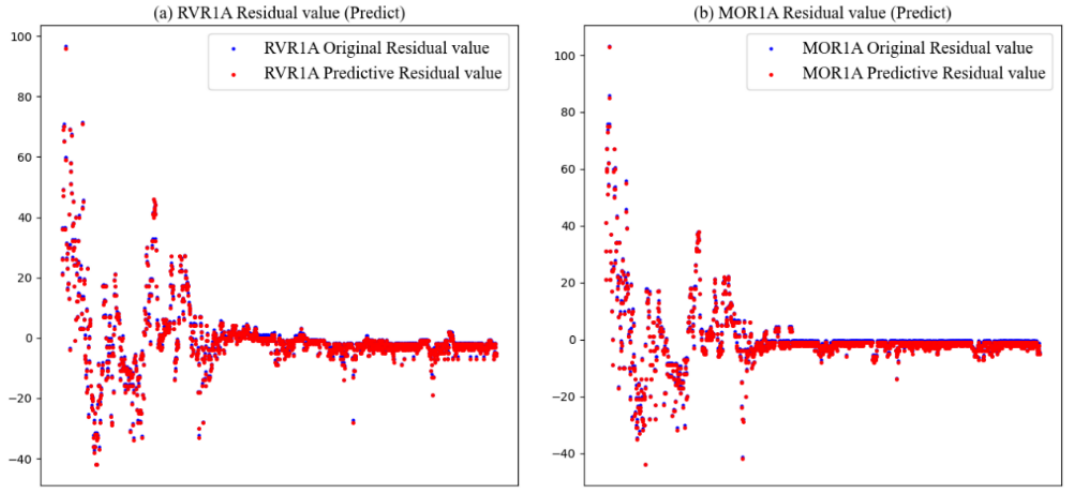

图 5 - 6 LSTM 神经网络残差估计模型残差估计结果图

5.5 随机森林-LSTM 能见度估计模型与精度评价

随机森林-LSTM 能见度估计模型是利用上文提出的机场观测视频图像分割、ResNet18 预训练模型图像特征提取、随机森林深度学习算法、LSTM 长短期记忆神经网络模型和 AMOS 机场能见度观测数据等；在对机场观测视频数据按时间序列分割并进行特征提取，最终与 AMOS 机场能见度观测数据匹配形成图像特征向量时间序列数据；然后使用随机森林深度学习算法依据数据训练能见度估计模型，并针对其残差使用 LSTM 神经网络进行回归分析，最终以随机森林模型能见度估计值与 LSTM 神经网络残差估计值之和作为最终能见度估计值。随机森林-LSTM 能见度估计模型能见度估计值计算公式如下：

$$V = R_v + L_r \qquad (5-11)$$

其中V为最终能见度估计值、R_v为随机森林模型能见度估计值、L_r为 LSTM 神经网络残差估计值。依据该公式与随机森林模型能见度数据和 LSTM 神经网络残差估计数据，在该时间序列上的能见度最终估计值与真实值如下图所示，由图可以看出相较于随机森林能见度估计模型结果图，下图中红色最终能见度估计点与蓝色能见度真实值点重合度更高，说明随机森林-LSTM 能见度估计模型在精度上更优于单独的随机森林能见度估计模型。其中图（a）和图（c）展示了 00:00 到 00:59 时段内波动幅度较大的 RVR 和 MOR 的最终估计，图（b）和图（d）展示了 01:00 到 05:59 时段内 RVR 和 MOR 的能见度最终估计结果，可以看出在各个时段和不同波动幅度情况下，随机森林-LSTM 能见度估计模型在能见度估计精度上表现优秀。

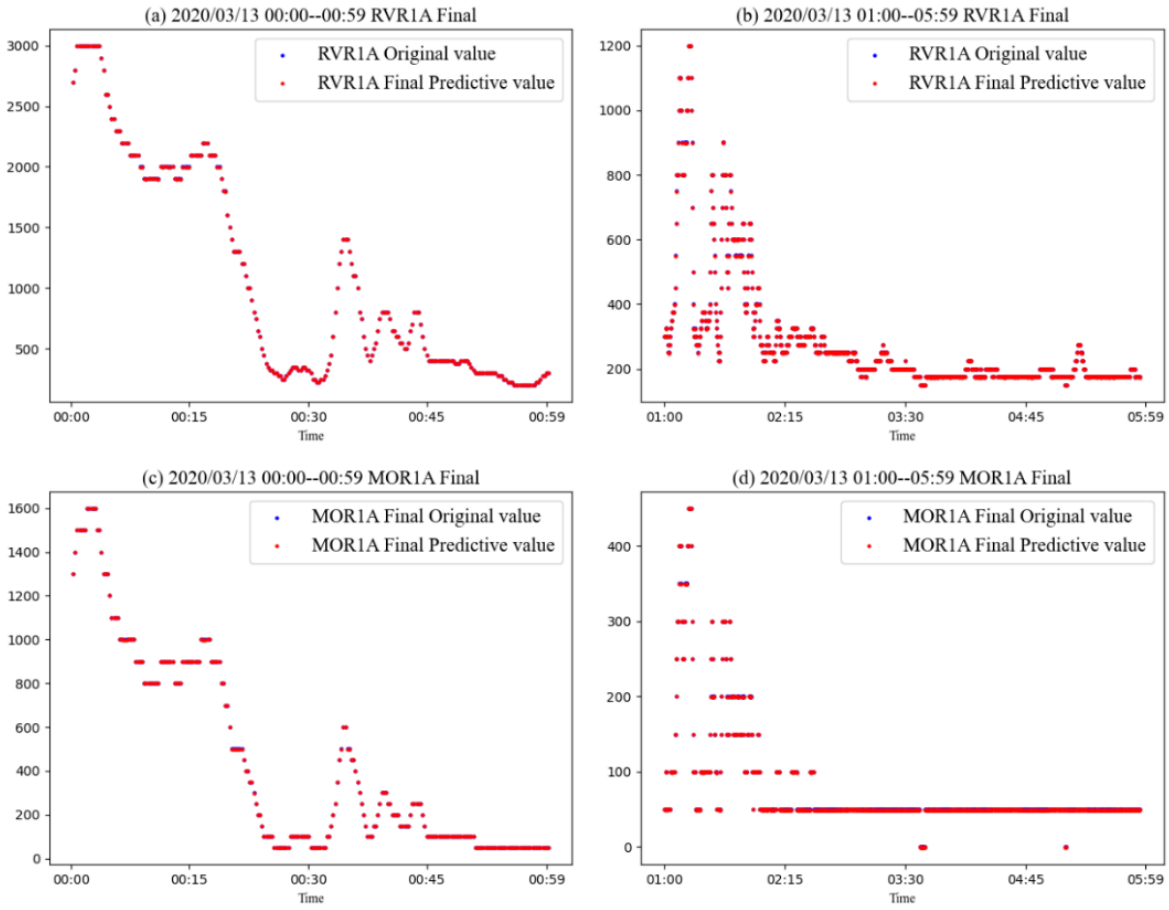

图 5 - 7 随机森林-LSTM 能见度估计模型能见度估计结果图

最后，本文采用均方误差（MSE）、均方根误差（RMSE）、平均绝对误差（MAE）、平均绝对百分比误差（MAPE）、对称平均绝对百分比误差（SMAPE）和校正决定系数（R-Squared）来评估随机森林-LSTM 能见度估计模型的精度，各项评价指标具体数值如下表所示，可以看出各项误差评价指标较小，说明了本文能见度估计模型的可靠性较好，另外两个能见度指标的 R-Squared 值分别为 97.9%和 95.8%，说明了本能见度估计模型精度较高。

表 5 - 2 随机森林-LSTM 能见度估计模型精度评价表

评价指标	RVR	MOR
MSE	0.291711908122376	0.33662464616430543
RMSE	0.5401036086922361	0.5801936281658955
MAE	0.4581356853240969	0.5237503901057319
MAPE	0.1785016604353095	0.2165025604363793
SMAPE	0.1787605298961361	3.127312186278788
R-Squared	0.9796988409926875	0.9589142340592004

六：问题三的模型建立与求解

6.1 问题描述与分析

针对问题三，基于高速公路某路段监控视频截图数据，建立不依赖于能见度仪观测数据的能见度估计算法，并绘制监控视频截图所提供的时间段内高速公路能见度随时间变化的曲线。因此，问题三的关键在于设计仅根据每张监控视频截图的数据求解高速公路能见度的算法，并利用求解得到的该时间段内每个时间点的高速公路能见度值进行曲线拟合，得到高速公路能见度随时间变化的关系。本文利用阈值分割法进行道路区域与背景天空区域的识别分割，结合能见度定义中人眼可见的目标物和背景的亮度对比阈值约束条件，获取每个时间点对应的监控视频截图图像中的人眼识别最远视点，通过摄像机标定实现二维图像像素点到对应三维现实空间中位置点的映射，从而完成图像最远视点到高速公路能见度的对应。根据估计出的高速公路能见度数据，绘制出能见度随时间变化的曲线，并对该曲线进行多项式回归拟合。

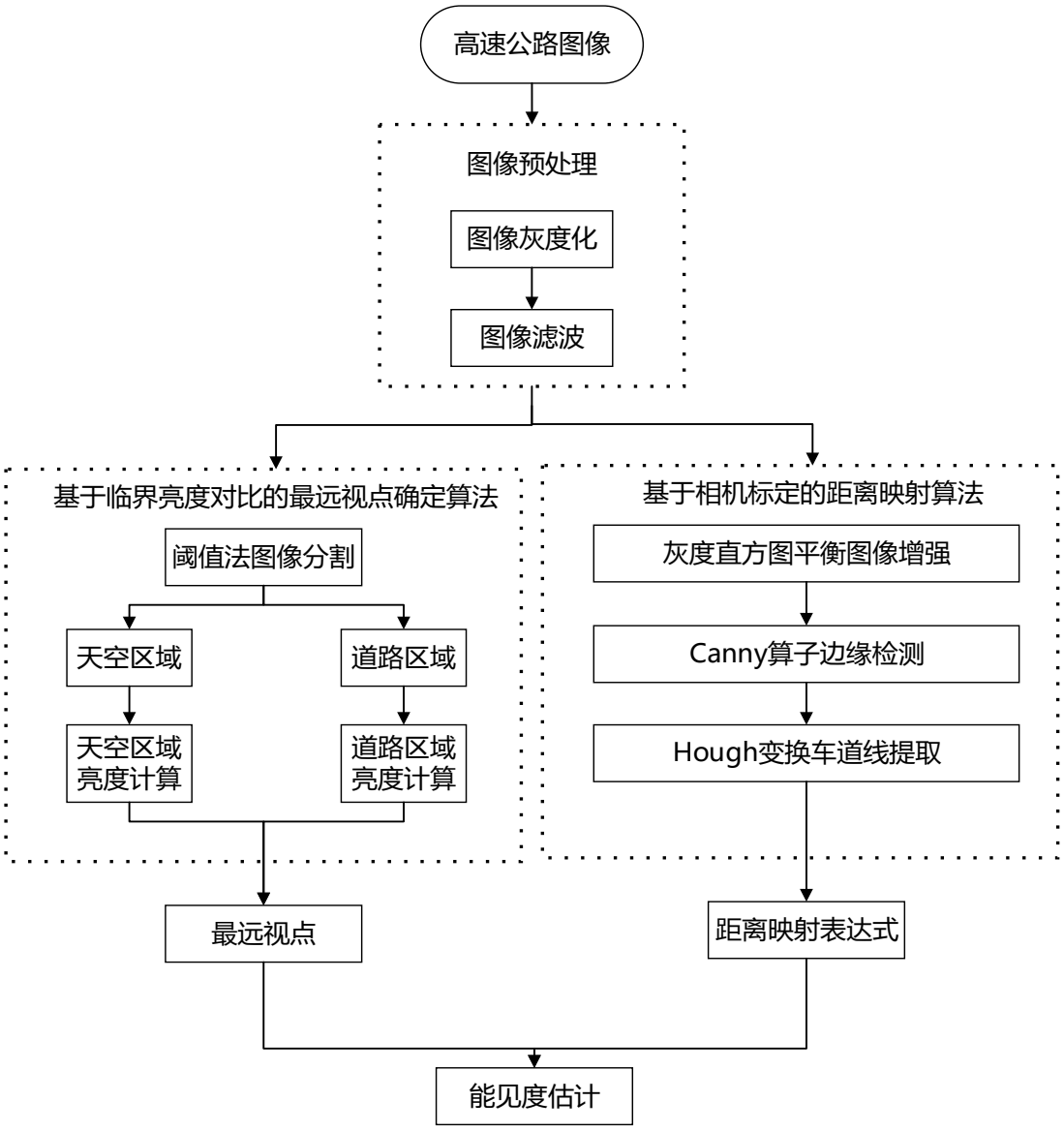

图 6 - 1 问题三流程图

6.2 基于临界亮度对比值的最远视点算法

6.2.1 最远视点估计原理

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。根据能见度的定义可知，能见度估计的重要决定因素之一为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。在此，本文考虑表示这种差异的指标，即亮度的对比值K。设B_0为目标物的固有亮度，$B_0{}'$为背景的固有亮度，则亮度的对比值定义为：

$$K=\begin{cases}\dfrac{|B_0{}'-B_0|}{B_0{}'}, & \text{if } B_0{}'\geq B_0\\ \dfrac{|B_0{}'-B_0|}{B_0}, & \text{if } B_0{}'<B_0\end{cases} \tag{6-1}$$

若目标物与背景的固有亮度一样，即$B_0=B_0{}'$，则$K=0$，说明不能分辨出目标物。若目标物为黑体，固有亮度为 0，即$B_0=0$，则$K=1$，说明目标物可以很容易地被识别出来。因此，人体对于目标物的识别会存在一个临界亮度对比值K，显然该临界亮度对比值是决定目标物是否能够被识别的关键，但同时该临界值与人体特征、识别环境特征、目标物特征等其他因素有关。

根据资料显示，若空气中不存在各种悬浮颗粒物，则人眼观测到的物体为物体的固有亮度，然而实际空气中存在大量悬浮微颗粒、气溶胶等，故从人眼的感官来讲，人眼看到的物体的亮度并非物体的固有亮度，而是由两部分组成，即空气亮度与经过削弱的物体固有亮度。空气亮度为大气光线进入人眼的亮度，一般可认为是观测者与物体之间的空气柱亮度，也可称空气亮度为目标物的背景亮度。而经过削弱的固有亮度是物体固有亮度经空气中颗粒物、气溶胶等散射与吸收后的亮度，该亮度的削弱可以用大气消光系数为度量计算。

Lamber-Beer 定律对大气的消光特性进行了描述，即解释了入射光与透射光、消光系数之间的关系，如下式所示：

$$F=F_0e^{-\sigma z} \tag{6-2}$$

其中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。

Koschmieder 在亮度衰减的研究基础之上，建立了目标物的视亮度与目标物表面的固有亮度、背景亮度以及大气消光系数之间的关系，如下所示：

$$L=L_0e^{-\sigma x}+(1-e^{-\sigma x})L_b \tag{6-3}$$

其中L为目标物的视亮度（cd/m^2），L_0为目标物的固有亮度（cd/m^2），L_b为背景亮度，σ为大气消光系数，x为观测点与目标物之间的距离（m）。

将上式进行变换之后得到

$$L-L_b=(L_0-L_b)e^{-\sigma x} \tag{6-4}$$

令$K=L-L_b$，$K_0=L_0-L_b$，于是上式可写成

$$K=K_0e^{-\sigma x} \tag{6-5}$$

其中K为目标物的视亮度与背景亮度之差，K_0为目标物的固有亮度与背景亮度之差。

进一步可得

$$\frac{K}{K_0} = e^{-\sigma x} \tag{6-6}$$

记$\varepsilon = \frac{K}{K_0}$，则有$\varepsilon = e^{-\sigma x}$.称$\varepsilon$为视觉对比阈值。根据国际照明委员会（CIE）的界定，其发布标准中规定ε为 0.05。

当摄像机为观察者时，那么对应的亮度会通过摄像机的感知元件映射到图像像素强度上。这里假设摄像机对图像像素的响应为线性函数，即实际场景中的亮度与图像像素强度之间的转换符合线性关系，并在此设两者之间的线性函数为F_{LI}，图像的像素强度为I，则图像像素强度I与亮度L之间关系符合：

$$I = F_{LI}(L) = F_{LI}(L_0 e^{-\sigma x} + (1 - e^{-\sigma x})L_b) \tag{6-7}$$

由假设F_{LI}为线性函数，可知F_{LI}满足线性可加性，即对于任意亮度L_1, L_2,以及任意常数α，都有

$$\begin{cases} F_{LI}(L_1 + L_2) = F_{LI}(L_1) + F_{LI}(L_2) \\ F_{LI}(\alpha L) = \alpha F_{LI}(L) \end{cases} \tag{6-8}$$

因此可得下式成立

$$I = e^{-\sigma x} F_{LI}(L_0) + (1 - e^{-\sigma x}) F_{LI}(L_b) \tag{6-9}$$

$F_{LI}(L_0)$表示目标物固有亮度映射到图像上为目标物固有像素强度，记为I_0，$F_{LI}(L_b)$为背景亮度映射到图像上为背景像素强度，记为I_b。由此可知，物体在图像上的像素强度也由两部分组成：一部分为经过削弱的物体固有像素强度，另一部分为空气像素强度，即可以表示为：

$$I = e^{-\sigma x} I_0 + (1 - e^{-\sigma x}) I_b \tag{6-10}$$

那么从理论角度来讲，我们可以认为基于图像的能见度与实际环境中人眼对于能见度的感知情况相同，即对于实际场景下能见度的计算方法适用于基于图像的能见度计算。

综上所述，人眼对目标识别所满足的临界条件为

$$\frac{K}{K_0} \geq 0.05 \tag{6-11}$$

接下来，考虑以路面为目标物、以天空为背景，通过路面亮度与背景亮度之间的对比计算图像中临界的目标行即可，也就是人眼识别的最远视点。

6.2.2 基于临界亮度对比值的最远视点算法的实现

（1）图像预处理

本文通过图像预处理来降低图像分析难度，该过程主要包含图像灰度化与图像滤波两个步骤。图像灰度化采用平均值法对彩色图像进行灰度化，将图像中的 R、G、B 三个分量变成了一个灰度值；图像滤波是对图像在其形成、传输记录过程中受到噪声的污染的处理，图像滤波实现了在尽量保留图像细节特征的条件下对目标图像的噪声进行抑制，本文利用中值滤波与高斯滤波预处理视频截图图像。图像预处理结果如下图 3.1 所示，Original、Gray、MedianBlur 和 GaussianBlur 分别展示了图像原图，灰度化图、中值滤波图和高斯滤波图，可以看出经图像滤波预处理后图像中的道路与背景天空区域的亮度对比度比原图的亮度对比度明显增大，更加便于观察高速公路能见度情况。

图 6 - 2 图像预处理

（2） 道路与天空区域划分

本文中采用临界亮度对比值确定最远视点，因此道路区域和天空背景区域的检测和识别工作是实现能见度估计算法的基础。从高速公路图向的特点来分析，摄像机一般布设在道路某一侧，且对交通检测呈俯视角度，天空位于图像上方，道路位于图像下方，并且图像中道路区域包含车道分割线。从视觉效果上来讲整个监控图像天空与路面之间的灰度呈现比较明显差异，故可采用阈值法进行处理。

阈值分割法基本思想：图像目标与背景像素灰度之间存在差异，反映到灰度直方图上分布上具有一定分离特征，故通过在直方图中选择适当的阈值将目标与背景进行区分。阈值分割是一种简单、效率高的图像分割方法，尤其是对目标与背景对比度较大的图像效果会更直观。设原始图像灰度为$f(x,y)$，图像分割的阈值为$T_{阈}$，分割后图像为$g(x,y)$，则采用阈值法进行图像分割满足以下条件：

$$g(x,y)=\begin{cases}1, & f(x,y)>T_{阈}\\ 0, & f(x,y)\le T_{阈}\end{cases} \qquad (6-12)$$

在此假设目标区域为道路区域，其余区域为天空背景区域，且满足低能见度条件下目标区域与背景域在灰度值存在较大差异，故需要确定合适的阈值对目标与背景分割，对图像二值化后进一步确定道路区域与天空区域。运用阈值分割法划分道路区域与背景天空区域，观察视频截图图像可知，图像涉及范围较小，每张图像仅由能见度范围内的道路区域与能见度范围以外的背景天空区域组成，划分结果如图 3.2 所示。其中图（a）为原图，图（b）为阈值确定后的二值化图，图（c）为逆二值化图，图（d）为全局阈值分割图，图（e）为天空区域图，图（f）为道路区域图。

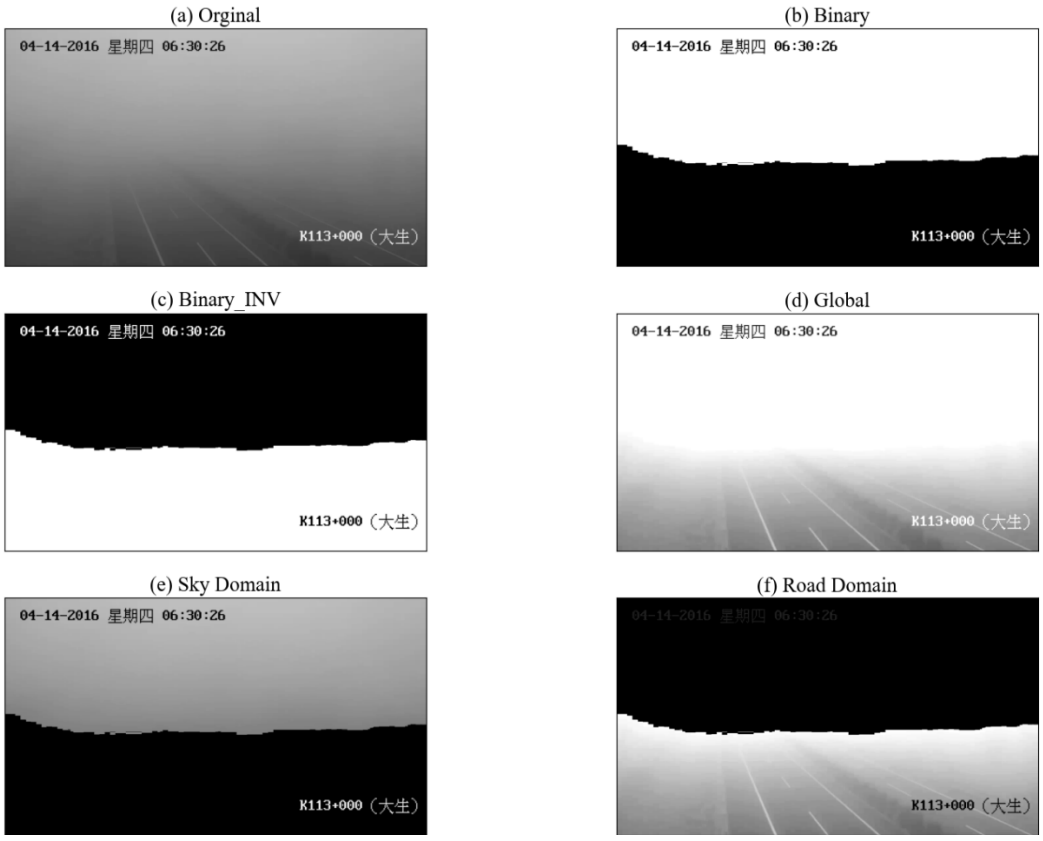

图 6 - 3 天空与道路区域识别图

（3） 道路与天空亮度计算

第一，考虑考虑路面亮度。理想情况下，对于路面同一行而言，亮度应该相同或具备连续性。而实际上对于路面区域而言，每一行的亮度也会出现不连续的状况，故本文先计算图像中路面第v行的连续像素点集，选取包含像素点数量最多的连续像素集$P_{cmax}(u,v)$，其对应的亮度为$L_{cmax}(u,v)$，设路面第v行的亮度为$L(v)$，则取连续像素集亮度的中值为路面第v行的亮度

$$L(v) = mid\ [L_{cmax}(u,v)] \qquad (6-13)$$

以一张高速公路图像为例，绘制道路亮度随图像中行的变化曲线如下：

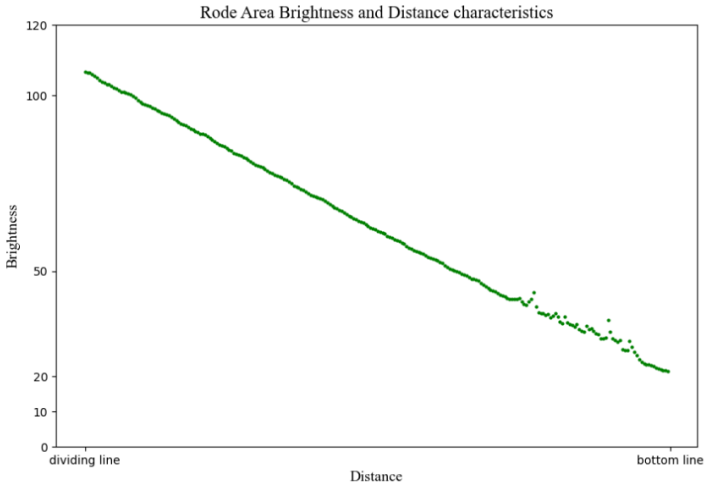

图 6 - 4 道路区域行亮度与距离关系图

第二 ，考虑天空背景亮度。在前一小节中对天空区域完成了检测与识别，假设空气为均质，故天空亮度的值可取为整个天空区域的亮度平均值，设天空亮度为S，包含天空区域的像素点共有n个，第j个像素点的亮度为$S_j(u_j, v_j)$，则天空的亮度为：

$$S = \frac{\sum_{j=1}^{n} S_j(u_j, v_j)}{n} \tag{6-14}$$

由上述能见度估计原理所得公式可知人眼对目标物的识别对比度为

$$\frac{K}{K_0} \geq 0.05 \tag{6-15}$$

该式为可视亮度满足的临界条件，因此本文以 0.05 为阈值提取图像中的满足该临界条件的可分辨像素，以路面为目标物、以天空为背景，通过计算亮度之间的对比值确定临界的目标行。在计算道路亮度时，为了降低噪声干扰等因素造成的误差，本文采用图像中的连续相邻行的路面宽度进行临界分辨率的判断，确定图像最远视点v_i，其中i代表图像像素中的第i行，$j \in \{0,1,2,3\}$.

$$\frac{|L(v_{i+j}) - S|}{S} \leq 0.05 \tag{6-16}$$

在上述条件下计算确定图像最远视点v_i，部分视频截图图像的天空亮度值与图像最远视点如下表所示：

表 6 - 1 天空亮度值与图像最远视点表

图像	天空亮度值	图像最远视点/行
Frame1	196.27	520
Frame2	196.39	520
Frame3	196.47	520
……	……	……
Frame98	195.22	496
Frame99	195.64	496
Frame100	195.89	496

6.3 基于摄像机标定的距离映射算法

高速公路监控系统通过摄像机获取视频图像，完成了三维现实世界场景到二维图像集合的投影，能见度估计算法中需要建立二维图像像素点与三维实际世界位置点的还原映射，这种映射关系是由摄像机的几何成像模型决定的，模型中包含摄像机的内参及外参等参数，而摄像机标定正是用来计算这些参数。本文在运用摄像机标定方法的过程中定义了四个参考坐标系，采用摄像机针孔成像模型完成了坐标系之间的转换，进而解决将图像像素点映射到实际位置点的问题，完成高速公路能见度的估算。

6.3.1 基于摄像机标定的距离映射算法原理

摄像机标定算法的关键在于五个坐标系之间的映射关系，它们分别是图像坐标系、图像物理坐标系、摄像机坐标系、世界坐标系和辅助坐标系，各坐标系详细介绍如下：

图像坐标系$U_pO_pV_p$，摄像机采集图像信息以像素（pixel）为单位描述图像中的行与列，故以此建立图像坐标系$U_pO_pV_p$，使用（u，v）表示图像像素点，其中像素的行与列均为整

数，即$u,v \in Z$，坐标原点O_p为图像的左上角。

图像物理坐标系$X_iO_iY_i$，图像坐标系是以图像像素为基本点，无法反映实际物理距离信息，因此需建立图像物理坐标系$X_iO_iY_i$描述图像的物理位置。图像物理坐标系$X_iO_iY_i$以摄像机光轴与图像平面的交点为坐标原点O_i，一般为图像的中心点，图像物理坐标系中的点用$(x_i，y_i)$表示，单位为毫米(mm)。设每一个像素在图像物理坐标系所占的物理距离分别为dx与dy，则由几何关系知图像坐标系$U_pO_pV_p$与图像物理坐标系$X_iO_iY_i$之间存在如下关系：

$$\begin{cases} u = \dfrac{x_i}{dx} + u_0 \\ v = \dfrac{y_i}{dy} + v_0 \end{cases} \quad 即 \quad \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \dfrac{1}{dx} & 0 & u_0 \\ 0 & \dfrac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \tag{6-17}$$

其中(u，v)为图像坐标系中的像素点，$(x_i，y_i)$为图像物理坐标系中相应的物理位置，$(u_0，v_0)$为图像坐标系中的图像中心像素点。图像坐标系与图像物理坐标系的关系如下图所示。

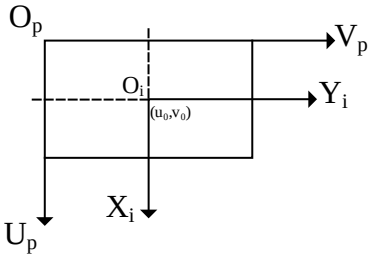

图 6 - 5 图像坐标系与图像物理坐标系

摄像机坐标系$X_cY_cZ_c$，摄像机坐标系是一个三维空间坐标系，其坐标原点O_c是摄像机光心，X_c轴与Y_c轴分别平行于图像物理坐标系$X_iO_iY_i$的X_i轴与Y_i轴，Z_c轴与摄像机光轴共线，与图像物理坐标系相交于图像物理坐标系原点O_i，O_cO_i为摄像机焦距f。如图所示。

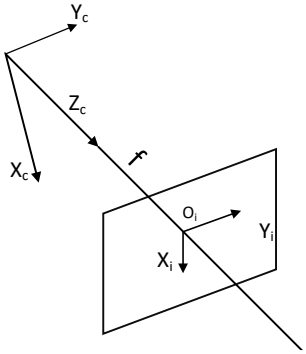

图 6 - 6 摄像机坐标系与图像物理坐标系的关系

根据摄像机针孔成像原理,存在如下关系

$$\frac{x_i}{x_c} = \frac{y_i}{y_c} = \frac{f}{z_c} \quad 即 \quad \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \frac{1}{z_c} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \tag{6-18}$$

其中(x_i, y_i)为图像物理坐标系下的点，(x_c, y_c, z_c)为摄像机坐标系中相应位置的点

综合以上两式，得到图像坐标系$U_pO_pV_p$与摄像机坐标系$X_cY_cZ_c$中点坐标的关系为：

$$\begin{cases} u = \dfrac{f}{z_c dx} x_c + u_0 \\ v = \dfrac{f}{z_c dy} y_c + v_0 \end{cases} \tag{6-19}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z_c} \begin{pmatrix} \dfrac{f}{dx} & 0 & u_0 \\ 0 & \dfrac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \frac{1}{z_c} \begin{pmatrix} \dfrac{f}{dx} & 0 & 0 \\ 0 & \dfrac{f}{dy} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \\ 0 \end{pmatrix} \tag{6-20}$$

世界坐标系$X_wY_wZ_w$，世界坐标系是用来表示现实世界中物体位置的坐标系。世界坐标系以摄像机坐标系Z_c光轴与地面的交点为原点O_w，Y_w与光轴Z_c在地面投影共线。由于假设路面为平面条件，因此有$Z_w = 0$。如下图所示。

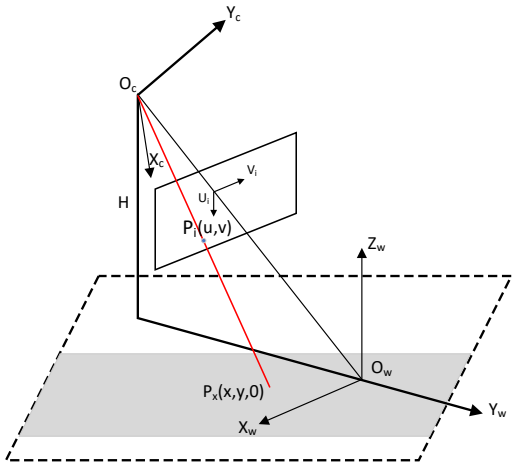

图 6 - 7 坐标系示意图

世界坐标系$X_wY_wZ_w$与摄像机坐标系$X_cY_cZ_c$之间存在刚性变换，即坐标系的位置与方向发生改变，而形状不变。世界坐标系与摄像机坐标系存在平移与旋转变换：

$$\begin{cases} x_c = r_1 x_w + t_1 \\ y_c = r_2 y_w + t_2 \\ z_c = r_3 z_w + t_3 \end{cases} \text{ 即 } \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} + T \tag{6-21}$$

其中为$r_i(i = 1,2,3)$旋转变换系数，$t_i(i = 1,2,3)$为平移转换系数。R为r_i组成的旋转矩阵，T为t_i组成的平移矩阵。摄像机坐标系以摄像机安装点为原点O_c，与光轴Z_c共线，设三维坐标系间满足右手定则，摄像机俯角为φ，即摄像机光轴Z_c与世界坐标系Z_w之间的夹角为φ。设摄像机安装距离地面的高度为H，则知摄像机坐标系与世界坐标系之间存在平移$H/\cos\varphi$ 。

辅助坐标系$X_fY_fZ_f$，在世界坐标系的基础上建立辅助坐标系$X_fY_fZ_f$，世界坐标系与辅助坐标系共原点O_w，X_f轴与世界坐标系的X_w轴重合，$Y_fO_fZ_f$为$Y_wO_wZ_w$绕原点O_w旋转角度φ后得到的结果。如图所示。

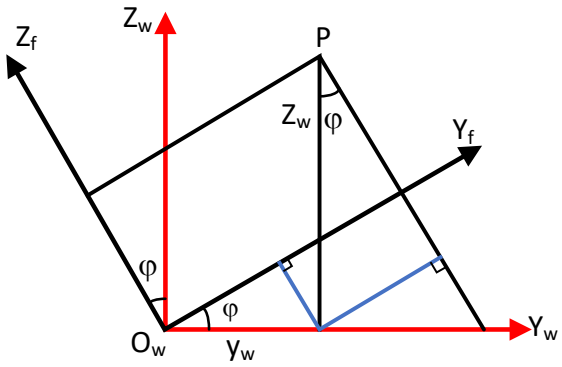

图 6 - 8 世界坐标系与辅助坐标系变换示意图

辅助坐标系$X_fY_fZ_f$与世界坐标系$X_wY_wZ_w$之间的坐标满足关系

$$\begin{cases} x_f = x_w \\ y_f = \cos\varphi\, y_w + \sin\varphi\, z_w \\ z_f = -\sin\varphi\, y_w + \cos\varphi\, z_w \end{cases} \quad 即 \quad \begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} \tag{6-22}$$

此时辅助坐标系$Y_fO_fZ_f$中Z_f轴与摄像机坐标系中的Z_c轴平行，平面$X_fO_fY_f$与平面$X_cO_cY_c$平行。综合两式，得到摄像机坐标系$X_cY_cZ_c$与辅助坐标系$X_fY_fZ_f$之间的关系

$$\begin{cases} x_c = -y_f \\ y_c = -x_f \\ z_c = -z_f + \dfrac{H}{\cos\varphi} \end{cases} \tag{6-23}$$

进一步地可得摄像机坐标系$X_cY_cZ_c$与世界坐标系$X_wY_wZ_w$之间的关系

$$\begin{cases} x_c = -\cos\varphi\, y_w - \sin\varphi\, z_w \\ y_c = -x_w \\ z_c = \sin\varphi\, y_w - \cos\varphi\, z_w + \dfrac{H}{\cos\varphi} \end{cases} \tag{6-24}$$

由于世界坐标系建立在路面上，满足约束条件$z_w = 0$，则在此条件下图像坐标系$U_pO_pV_p$与世界坐标系$X_wY_wZ_w$中点的坐标满足：

$$\begin{cases} u = \dfrac{-\cos\varphi\, y_w f}{\sin\varphi\, y_w dx + \dfrac{H}{\cos\varphi} dx} + u_0 \\ v = \dfrac{-x_w f}{\sin\varphi\, y_w dy + \dfrac{H}{\cos\varphi} dy} + v_0 \end{cases} \tag{6-25}$$

摄像机内部参数有f, dx, dy，对于一般的摄像机而言，由于制作工艺的约束可近似假设参数$dx = dy$，这里可以令$\rho = \frac{f}{dx}$，则上式可以表示为

$$\begin{cases} u = \dfrac{-\cos\varphi\, y_w \rho}{\sin\varphi\, y_w + \dfrac{H}{\cos\varphi}} + u_0 \\ v = \dfrac{-x_w \rho}{\sin\varphi\, y_w + \dfrac{H}{\cos\varphi}} + v_0 \end{cases} \tag{6-26}$$

将上式进行变换，得到现实世界中任意一点$(x_w，y_w)$使用图像像素点$(u，v)$表示为

$$\begin{cases} x_w = \dfrac{(v_0 - \mathrm{v})\mathrm{H}}{(u-u_0)\sin\varphi + \rho\cos\varphi} \\ y_w = \dfrac{(u-u_0)H}{[(u-u_0)\sin\varphi + \rho\cos\varphi]} \end{cases} \quad (6-27)$$

根据《高速公路监控技术要求》（2012 年 3 号公告），高速公路上摄像机采用立柱安装，安装高度在 8~12m 范围内，在此情况下未知参数为ρ与φ，。

标定几何建模，为了标定摄像机的内参和外参等相关参数，本文利用高速公路固有的路面车道线构造尺寸数据选取了若干关键点。根据《公路交通安全设施设计细则》（JTG/TD-2006）对车道分界线线长的规定，在设计速度为 100km/h 的高速公路上车道分界线之间的距离关系如下图所示，每一条车道分界线的长度为 6m，相邻车道分界线之间的距离为 9m。

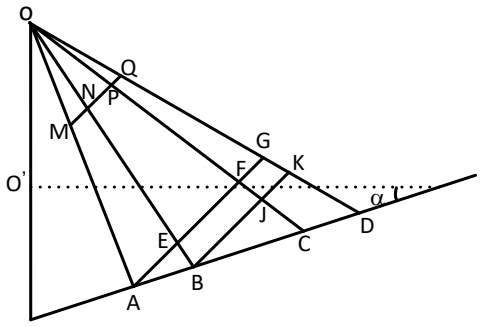

图 6 - 9 几何推导模型图

在摄像机标定时选取车道分界线为标定特征，从几何视角上来看，分界线近似位于一条直线上，本文基于几何建模对摄像机标定进行分析。通过观察高速公路视频截图图像发现，摄像机光轴与道路方向并不一致，如图所示，假设点O为摄像机所在位置，监控图像上的点M、N、P、Q，映射到道路上各点分别为A、B、C、D，其中AB所在直线为车道分界线所在直线，MN所在直线为图像中的车道分界线所在直线，OA、OB、OC、OD为摄像机进行透射时的光线。以A为端点，做辅助线段AG，满足$AG \parallel MQ$，分别交线段OB、OC于点E、F；以B为端点，做辅助线段BK，满足$BK \parallel MQ$，交线段BK于点J。

由图中条件$AG \parallel MQ$且$BK \parallel MQ$可知，$\Delta AGD \sim \Delta BKD$，于是

$$\frac{BD}{AD} = \frac{BK}{AG} \quad (6-28)$$

另外

$$\frac{MN}{AE} = \frac{MQ}{AG}, \text{即 } AG = \frac{AE \cdot MQ}{MN} \quad (6-29)$$

$$\frac{NP}{BJ} = \frac{NQ}{BK}, \text{ 即 } BK = \frac{NQ \cdot BJ}{NP} \quad (6-30)$$

则由以上三式可得

$$\frac{BD}{AD} = \frac{MN}{MQ} \cdot \frac{NQ}{NP} \cdot \frac{BJ}{AE} \quad (6-31)$$

由$\Delta AGD \sim \Delta BKD$可得

$$\frac{AF}{BJ}=\frac{AC}{BC},\text{即 } B'J=\frac{AF\cdot BC}{AC} \tag{6-32}$$

将式（6-32）代入，可得

$$\frac{BD}{AD}=\frac{MN}{MQ}\cdot\frac{NQ}{NP}\cdot\frac{AF}{AE}\cdot\frac{BC}{AC} \tag{6-33}$$

由$AG \parallel MQ$可知，

$$\frac{AF}{AE}=\frac{MP}{MN} \tag{6-34}$$

将式（6-34）代入，得到

$$\frac{BD}{AD}=\frac{MP}{MQ}\cdot\frac{NQ}{NP}\cdot\frac{BC}{AC} \tag{6-35}$$

$$AD=\frac{AB\cdot MQ\cdot NP\cdot AC}{MQ\cdot NP\cdot AC-MP\cdot NQ\cdot BC} \tag{6-36}$$

假设实际世界距离为$AB=l_1$，$BC=l_2$，$AD=l$，图像上像素点为$M(u_m,v_m)$，$N(u_n,v_n)$，$P(u_p,v_p)$，$Q(u_q,v_q)$，设每一个像素的行与列所对应的物理距离分别为dx与dy，即可得到

$$\begin{cases}MQ=(v_q-v_m)dy\\MP=(v_p-v_m)dy\\NP=(v_p-v_n)dy\\NQ=(v_q-v_n)dy\end{cases} \tag{6-37}$$

将式（6-37）代入，可得

$$l=\frac{l_1l_2(v_p-v_m)(v_q-v_n)}{l_2(v_p-v_m)(v_q-v_n)-(l_1+l_2)(v_q-v_m)(v_p-v_n)} \tag{6-38}$$

根据上式可知，对于沿摄像机光轴投影方向的任意一点$Q(u_q,v_q)$，可求得点Q对应的现实世界位置点到摄像机的距离l，在这种情况下，上式符合函数模型

$$y=\frac{ax}{b+cx}\text{，其中}a>0\text{，}b>0\text{，}c\neq0 \tag{6-39}$$

因此，本文将该函数作为摄像机标定的曲线拟合模型。

在现实世界环境中，两条平行直线是不相交的，但经过摄像机投影到二维图像之后，相互平行的道路在某处会出现相交的情况，我们称相交点为消失点，即实际平行线经过摄像机投影后在图像中汇聚形成的点。如下图所示。

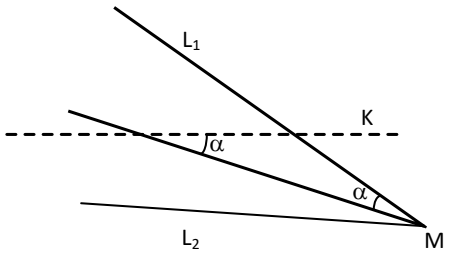

图 6 - 10 道路图像消失点示意图

其中，L_1，L_2为实际环境中道路的两条相互平行边缘，直线K为摄像机光轴在道路所在平面上的投影，在图像上L_1与L_2相交于点M，此时M为道路消失点。夹角α为以道路消失点为顶点的两条平行道路边缘直线所形成夹角的平分角。

6.3.2 基于摄像机标定的距离映射算法实现

（1）图像增强

由于在低能见度的高速公路图像中，图像羽化严重，很难检测高速公路的边缘和车道线的重要信息。为了更好的识别高速公路车道线等特征，本文首先对高速公路图像进行图像增强。采用图像灰度直方图平衡的方法，通过对图像灰度直方图进行修正来获得图像增强效果，主要是进行对比度增强，让亮的更亮，让暗的更暗。

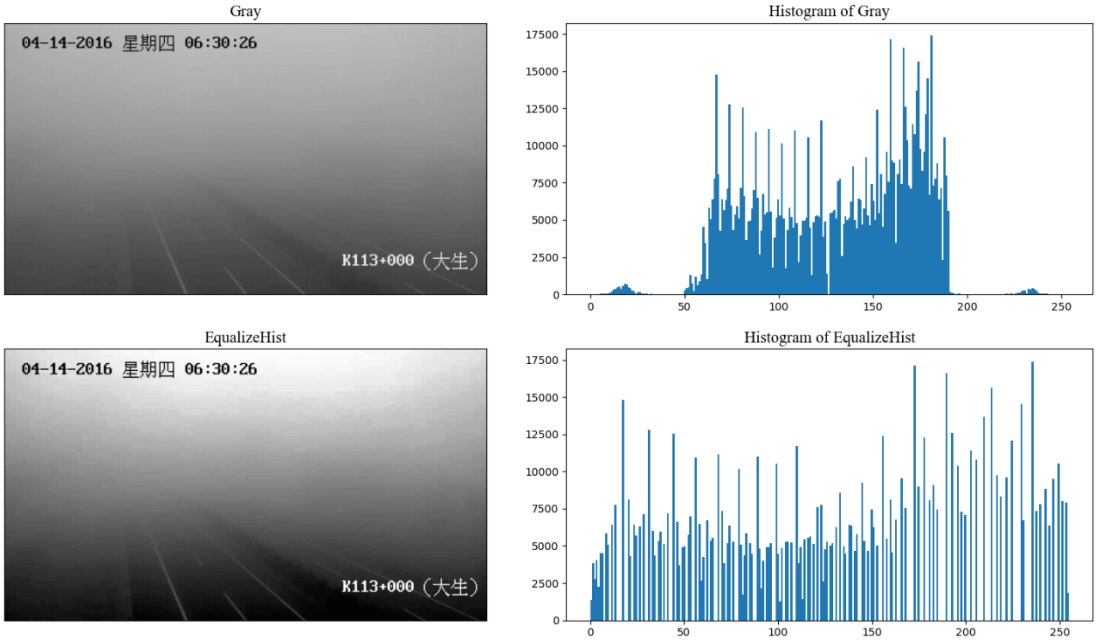

图 6 - 11 图像灰度直方图平衡

图像灰度直方图平衡图像增强结果如图 3.5 所示，图 Gray 和图 Histogram of Gray 分别为原图与其对应的灰度直方图，从直方图可以看出，其灰度分布较为不均，大多数集中在 50 到 200 灰度级之间。通过灰度直方图平衡后，如图 EqualizeHist 和图 Histogram of EqualizeHist 所示，平衡后的直方图中灰度明显比未平衡时较均匀，图像道路区域与天空区域亮度对比度明显增强。

（2）边缘检测

本文采用的摄像机标定的距离映射算法的关键在于使用车道线上的点估计实际距离从而拟合摄像机标定的实际距离映射模型。为了更好地获取高速公路车道线信息，本文分别使用一阶导数算子 Roberts、Prewitt、Sobel 和二阶导数算子 Log、Canny 对高速公路图像进行边缘检测，并选择效果最好的算子完成接下来的图像边缘检测。利用不同的边缘检测算子对图像进行处理，检测效果如图 3.6 所示。从 6 张图中可以看出，图（f）展示的 Canny 算子检测出来的道路边缘准确、边缘遗漏与误检率较低，检测的边缘信息比较全面丰富，因此本文选择边缘检测效果较好的 canny 算子完成图像边缘检测。

图 6 - 12 边缘算子检测效果图

（3）高速公路车道线的提取

监控视频图像中高速公路基本呈直线与曲线，且根据在能见度研究的需求，只考虑道路线性比较良好的，采用道路的直线模型处理。本文采用 Hough 变换获取道路直线的参数值，并在此基础上拟合道路直线，定位道路区域。Hough 变换属于图像特征提取方法，可实现图像空间与参数空间的映射关系，基本思想是假设图像中存在一条直线，并设该直线满足 $y = kx + b$，参数 k 为直线的斜率，b 为截距，故该直线主要参数为 k 与 b。

图 6 - 13 Hough 变换图像

若直线上任意两点(x_1,y_1)、 (x_2,y_2)，则利用两点式可求解直线方程。而 Hough 变换是将$y=kx+b$转换成$b=-kx+y$，则代入直线任意两点(x_1,y_1)、 (x_2,y_2)，即直线 $b=-kx_1+y_1$ 与 $b=-kx_2+y_2$ 会在参数空间内相交于一点，交点坐标为所求参数。Hough 变换的实质是将直线的求解问题转换成寻找参数空间的交点问题。经 Hough 变换所得图像如图 3.7 所示，通过对比 Canny 边缘检测可知 Hough 变换较为准确地提取了车道线信息。

（4）基于摄像机标定的距离映射

使用 Hough 变换后获得的车道分界线的检测结果对摄像机进行标定，利用标定拟合曲线将图像中的原始像素扩展成实际物理尺寸。根据摄像机标定算法原理所得公式，要完成摄像机标定需要知道若干参数，在此假设摄像机焦距为 6mm，相机摆放高度为 10m，相机摆放角度为 30 度，相机视角为 30 度，已知车道虚线长度为 6m，虚线间隔为 9 米，以此作为图像比例尺推算实际距离。下面将选取若干个车道线点，依据估算实际距离，标定拟合曲线，摄像机标定模型满足公式

$$y=\frac{ax}{b+cx}，\text{其中}a>0，b>0，c\neq 0 \qquad (6-40)$$

选取左侧车道线上的 8 个点，经曲线拟合可得各参数分别为

$$a=1.13864612，\mathrm{b}=-203.51593329，\mathrm{c}=0.42949565$$

$$y=\frac{1.13864612x}{-203.51593329+0.42949565x} \qquad (6-41)$$

.拟合曲线结果如图 3.8 所示，其中车道线端点纵坐标为标定拟合曲线的 x 轴。

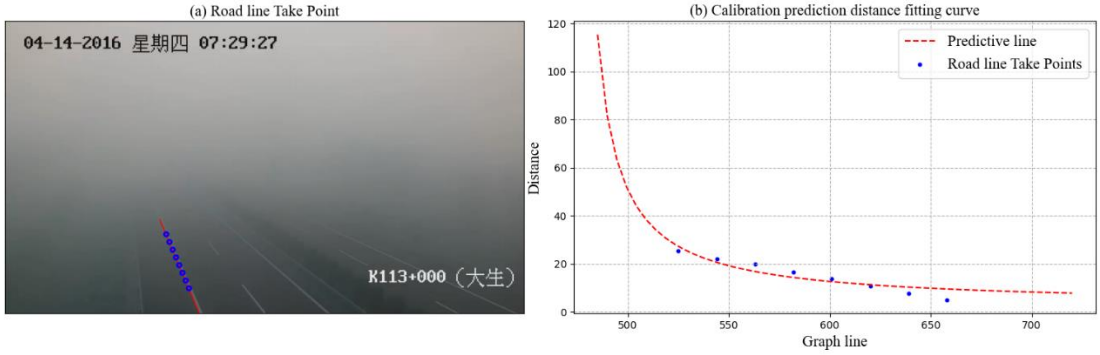

图 6 - 14 道路取点与标定拟合曲线

选取车道线点测试结果如下表所示，从表中可知图像上的像素点与实际距离的映射关系较为准确。

表 6 - 2 车道线标定测试结果

车道线点标号	车道线点坐标	标定实际距离/m	标定预测距离/m
1	[397, 525]	25.45	27.210228667381124
2	[405, 544]	21.88	20.558567888390655
3	[413, 563]	19.82	16.742120545366838
4	[421, 582]	16.65	14.266618063652738
5	[429, 601]	13.74	12.530935435970113
6	[437, 620]	10.56	11.246537994335569
7	[445, 639]	7.55	10.25767009175364
8	[453, 658]	4.78	9.472857060300516

6.4 基于临界亮度对比与摄像机标定的能见度估计算法实现

在基于临界亮度对比的最远视点确定算法实现中，获得了本题所给 100 张高速公路图像的图像最远视点所对应的图像坐标系行坐标。在基于摄像机标定的距离映射算法实现中，得到了拟合后的图像坐标系到世界坐标系的距离映射公式。结合两个算法的实现依据图像的最远视点和距离映射公式，对各图像能见度估计部分展示如下表所示：

表 6 - 3 图像能见度估计

图像	图像最远视点/行	能见度估计/m
Frame1	520	29.87094218
Frame2	520	29.87094218
Frame3	520	29.87094218
Frame4	525	27.21023
Frame5	525	27.21023
……	……	……
Frame98	496	59.3624
Frame99	496	59.3624
Frame100	496	59.3624

将表 3.10 绘制成能见度估计值随时间变化的曲线，并且运用一次、二次、三次、四次多项式函数对曲线进行拟合，发现三次多项式函数大体上符合能见度估计值随时间变化的趋势，因此我们选取三次多项式拟合能见度随时间变化的曲线。如下图所示。

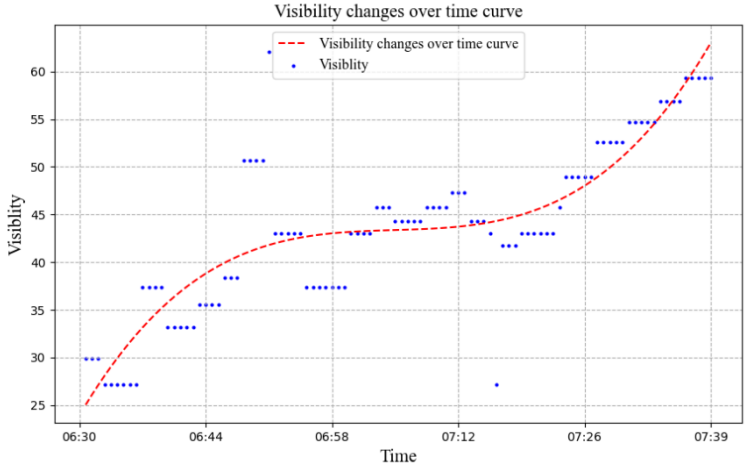

图 6 - 15 能见度随时间变化曲线

进行多项式回归分析拟合结果为：

$$y = 0.0001483\,x^3 - 0.02221x^2 + 1.128\,x + 23.93 \qquad (6-42)$$

七：问题四的模型建立与求解

7.1 问题描述与分析

针对问题四，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加重或减弱的趋势，以及预测大雾在何时会散去，即何时达到指定的能见度。根据题目中提及的能见度测量基本方程，预测大雾的变化趋势问题也就转化为预测能见度的变化趋势问题。在问题三中，文中已经得到能见度估计值随时间变化的拟合曲线，因此，只需通过对能见度曲线进行多项式函数拟合，获取能见度随时间变化的表达式，然后根据表达式预测未来能见度变化规律，基于 Koschmieder 定律及人眼对比度阈值，得到代表大雾浓度的指标衰减系数随时间变化的规律，以此预测大雾的变化趋势及规律。

7.2 模型的建立

7.2.1 基于多项式模型的能见度变化曲线拟合

在问题三中，已完成能见度估计值随时间变化的曲线绘制工作，本文采用多项式函数模型对能见度变化曲线进行拟合，在此分别运用一次、二次、三次、四次多项式函数对曲线进行拟合，发现三次多项式函数大体上符合能见度估计值随时间变化的趋势，而四次多项式和三次多项式拟合曲线大致相当，且高次多项式易导致模型过拟合，因此我们选取三次多项式拟合能见度随时间变化的曲线，如下图（a）所示。

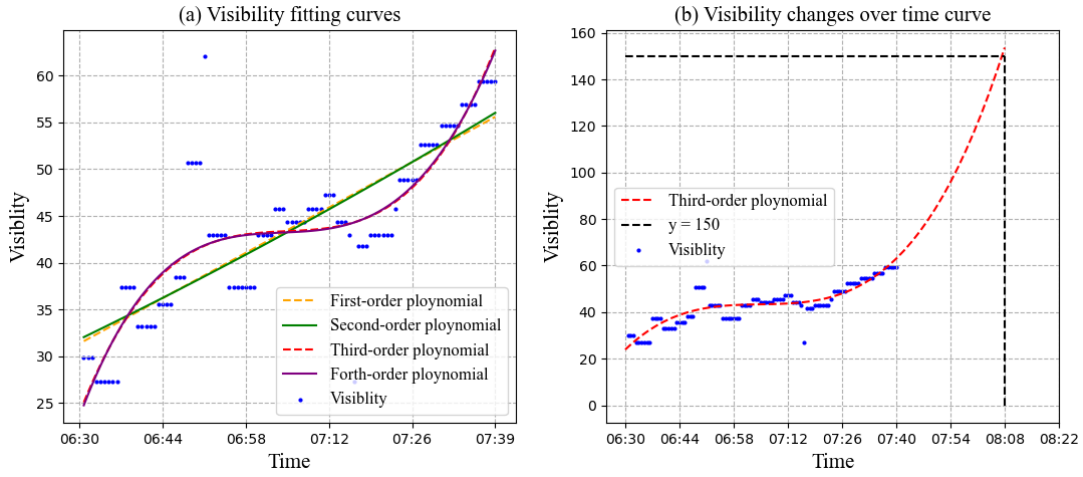

图 7 - 1 能见度随时间变化曲线

其中，一次多项式函数的拟合结果为

$$y = 0.2426\,x + 31.35 \qquad (7-1)$$

二次多项式函数的拟合结果为

$$y = 0.0002646\,x^2 - 0.02221\,x + 31.8 \qquad (7-2)$$

三次多项式函数的拟合结果为

$$y = 0.0001483\,x^3 - 0.02221x^2 + 1.128\,x + 23.93 \qquad (7-3)$$

四次多项式函数的拟合结果为

$$y = -2.445 * 10^{-7}x^4 + 0.0001977\,x^3 - 0.02543x^2 + 1.201x + 23.55 \qquad (7-4)$$

本文选取的能见度变化曲线拟合表达式为

$$y = 0.0001483\,x^3 - 0.02221x^2 + 1.128\,x + 23.93 \quad (7-5)$$

依据能见度变化曲线拟合表达式，计算气象能见度 MOR 值为 150m 时，对应时间为 08:08 如图（b）所示,因此依据能见度变化曲线高速公路雾将在 08:08 时散去。

7.2.2 基于 Koschmieder 定律的大雾浓度计算

Koschmieder 定律建立了目标物的视亮度与目标物表面的固有亮度、背景亮度以及大气消光系数之间的关系，如下所示：

$$L = L_0 e^{-z} + (1 - e^{-\sigma z}) L_b \quad (7-6)$$

其中L为目标物的视亮度（cd/m^2），L_0为目标物的固有亮度（cd/m^2），L_b为背景亮度，σ为大气消光系数，x为观测点与目标物之间的距离（m）。将上式进行变换之后得到

$$L - L_b = (L_0 - L_b) e^{-\sigma x} \quad (7-7)$$

令$F = L - L_b$，$F_0 = L_0 - L_b$，于是上式可写成

$$F = F_0 e^{-\sigma z} \quad (7-8)$$

其中F为目标物的视亮度与背景亮度之差，F_0为目标物的固有亮度与背景亮度之差，可得

$$\frac{F}{F_0} = e^{-\sigma z} \quad (7-9)$$

记$\varepsilon = F/F_0$，则有$\varepsilon = e^{-\sigma z}$.称$\varepsilon$为视觉对比阈值。根据国际照明委员会（CIE）的界定，其发布标准中规定ε为 0.05。根据题目所提供的信息，参数σ也称为衰减系数，与雾的厚度有关：σ越大表明雾越浓。能见度对应的指标气象光学视程(MOR)满足

$$\mathrm{MOR} = \frac{\log\left(\frac{F}{F_0}\right)}{-\sigma} = \frac{\log(0.05)}{-\sigma} \quad (7-10)$$

因此，衡量大雾浓度的指标σ满足

$$\sigma = \frac{2.996}{MOR} \quad (7-11)$$

结合能见度变化曲线拟合表达式，可得雾随时间的变化规律如公式（7-12）其中t表示时间。

$$\sigma = \frac{2.996}{MOR} = \frac{2.996}{0.0001483\,x^3 - 0.02221x^2 + 1.128\,x + 23.93} \quad (7-12)$$

依据表达式衰减系数的变化趋势如下图所示，随时间递减，说明大雾的变化趋势是减弱。

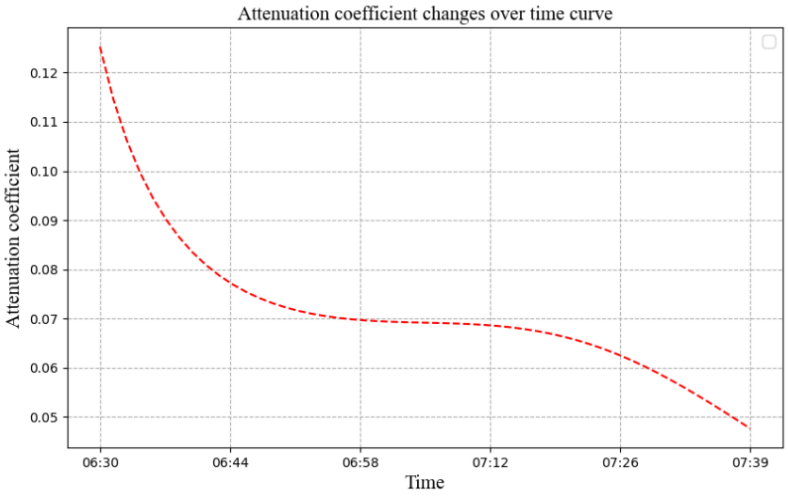

图 7 - 2 衰减系数随时间变化图

八：模型评价

8.1 模型的优点

问题一：在数据预处理方面，采用加权插值法对数据量较少的气象数据进行扩充，使得后续回归过程中，能更精确地反映真实函数关系。在多元线性回归模型中采用主成分分析方法，降低了对众多气象数据进行分析的难度。同时，某种程度上也对最终得到的能见度关系式进行了降维处理。在进行多元线性回归之前，首先多不同条件进行了划分，这样也使得模型可以运用在该多变量的问题之中。

问题二：在数据处理方面，采用基于时间序列的视频分割方案，保留视频的连续信息。在特征提取方面，采用基于迁移学习的 ResNet18 预训练模型提取图像的 512 维高层特征，极大了数据规模，让深度学习算法能够发挥更好的性能。在能见度估计模型构建方面，采用随机森林-LSTM 神经网络模型，使用随机森林深度学习算法初步训练能见度估计模型，接着使用 LSTM 神经网络对其残差进行回归分析，最终能见度估计值为随机森林模型能见度估计值与 LSTM 模型残差估计值之和，采用这种集成方法极大提高了能见度估计模型的精度。

问题三：在图像处理方面，采用图像灰度化和图像滤波预处理抑制噪声污染，运用图像灰度直方图平衡进行图像增强，对比多种梯度算子的边缘检测效果，选取检测效果最好的完成道路边缘检测，Hough 变换提取车道线信息，皆有助于提高后续图像处理和分析工作的有效性和可靠性。在摄像机标定中，标定方法对于原始视频图像拍摄方面的要求不多，因此。摄像机标定不失为一种处理高速公路视频图像的有效方式。结合摄像机部分参数假定以及提取的车道线信息数据拟合标定曲线，使摄像机标定过程更加合理化与简便化。

问题四：结合测量基本方程与人眼对比度阈值获得大雾浓度指标衰减系数与能见度之间的关系，简化且定量描述了大雾随时间的变化规律。

8.2 模型的缺点

问题一：由于对数据集进行了区间的划分，将其划分在 4 种不同的条件之下，也使得最后给出的关系式数量较多，对于能见度与气象勘测数据间的关系表达不够直观，有待在不同区间进行优化整合。

问题二：由于采用 ResNet18 预训练模型进行图像的高层特征提取，导致提取的 512 维图像特征向量的可解释性较差。另外，采用随机森林深度学习算法与 LSTM 神经网络模型集成的能见度估计模型，导致模型训练过程资源消耗较多，训练效率较低。

问题三：在摄像机标定过程中，标定结果受到摄像机部分参数假定条件的影响较大，摄像机参数的确定缺乏一定的客观性。

问题四：由于时间仓促等原因，高速公路能见度随时间变化的曲线拟合模型有待完善，并且高速公路能见度是受外在环境中各种因素影响的，仅考虑拟合能见度随时间变化趋势的预测效果是有待考量的。

8.3 模型的展望

在问题一中，少数变量与能见度存在非线性的关系，故还需要引入适用于刻画非线性关系的模型来加以补充。

在问题二中，在使用随机森林深度学习算法时，由于时间有限没有进行超参数调优，

只是依据一般情况进行参数选择，在模型训练结果上可能没有达到最好，在未来可以考虑更加细致的超参数调优，训练更加可靠有效的分析模型。

在问题三中，解决能见度估计问题的方法还包括双亮度差法、暗通道法等方法，但不同方法的适用条件稍有差别，可考虑将本文中的摄像机标定方法推广到不同的摄像场景得到的图像。

在问题四中，由于高速公路能见度受到多种外在因素的影响，可以考虑建模估计预测外在因素影响下的大雾浓度。

参考文献

[1] 张旭,王卫,白婷,吉莉莉,康俊,张佳敏.四川浅发酵香肠加工进程中挥发性风味物质测定及其主成分分析[J/OL].现代食品科技:1-10[2020-09-21].https://doi.org/10.13982/j.mfst.1673-9078.2020.10.0383.

[2] 洪昕妍.基于主成分分析的新疆水资源承载力分析[J].水利规划与设计,2020(09):39-41+60.

[3] 苗晓颖,胡继连.山东棉花种植面积变动及影响因素分析——基于 14 变量的因子回归分析[J].当代经济,2020(09):85-89.

[4] 彭润龙,张亚如,郭荣伟.基于多元回归全球气候变化的模型分析[J].齐鲁工业大学学报,2020,34(04):61-68.

[5] 李莹. 基于时间序列与多元线性回归综合模型的农村卷烟销量预测[D].云南大学,2015.

[6] Nayar S K , Narasimhan S G . Vision in bad weather[C]// Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. IEEE, 1999.

Tan R T . Visibility in bad weather from a single image[C]// 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008.

[7] Sakaridis, Christos, Dai, et al. Semantic Foggy Scene Understanding with Synthetic Data[J]. INTERNATIONAL JOURNAL OF COMPUTER VISION, 2018.

[8] 张帅. 用于轮胎花纹分类的图像特征提取算法研究[D]. 2019.

[9] 邓拓. 基于 LSTM 神经网络的机场能见度预测[D]. 2019.

[10] Zhu L , Zhu G , Han L , et al. The Application of Deep Learning in Airport Visibility Forecast[J]. Atmospheric & Climate ences, 2017, 07(3):314-322.

[11] Wen C , Liu S , Yao X , et al. A novel spatiotemporal convolutional long short-term neural network for air pollution prediction[J]. Science of the Total Environment, 2019, 654(MAR.1):1091-1099.

[12] Salman. A. G, Heryadi. Y，Abdurahman. E. Wayan SupartaWeather forecasting using merged Long Short-Term Memory Model[J]. Bulletin of Electrical Engineering and Informatic. 2018, 7(3):377-385

[13] 龙科军, 李超群, 毛学军, et al. 高速公路雾天能见度预测方法[J]. 徐州工程学院学报(自然科学版), 2017(32):31-37.

[14] 李红梅. 基于图像处理的高速公路能见度检测技术研究[D].

[15] Nicolas Hautiére, Tarel J P , Lavenant J , et al. Automatic fog detection and estimation of visibility distance through use of an onboard camera[J]. Machine Vision and Applications, 2006, 17(1):8-20.

[16] Hallowell R G, Matthews M P, Pisano P A. Automated extraction of weather variables from camera imagery[C]. Proceeding of the 2005 Mid Continent Transportation Research Symposium, Ames, IA, 2005: 1 13

[17] Nicolas Hautière, Babari R , Dumont E , et al. Estimating Meteorological Visibility Using Cameras: A Probabilistic Model-Driven Approach[C]// Asian Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010.

[18] 陈文艺, 来庆盈, 杨辉. 基于相位检测的双平面摄像机标定[J]. 计算机工程与设计, 2020(7).

[19] Tarel J P , Nicolas Hautière. Fast visibility restoration from a single color or gray level

image[C]// IEEE International Conference on Computer Vision. IEEE, 2010.
[20] Wenqi Ren, Si Liu, Hua Zhang. Single Image Dehazing via Multi-scale Convolutional Neural Networks[M]// Computer Vision – ECCV 2016. Springer International Publishing, 2016.
[21] Johannes Kopf, Boris Neubert, Billy Chen,等. Deep photo: model-based photograph enhancement and viewing[C]// Acm Siggraph Asia. ACM, 2008.

附录

问题一数据归一化代码

```
import pandas as pd

def MaxMinNormalization(x,Max,Min):
    x = (x - Min) / (Max - Min);
    return x

def WriteData(X,length,Max,Min,filename):
    for i in range(length):
        with open(filename,'a') as f:
            f.write(str(MaxMinNormalization(X[i],Max,Min))+'\n')

'''
data = pd.read_excel('amos191216.xlsx')
attri = ['PAINS','QNH','QFE','TEMP','RH','DEWPOINT','WS2A','WD2A','CW2A','RVR1A','MOR1A']
filename = ['data1.txt','data2.txt','data3.txt','data4.txt','data5.txt','data6.txt','data7.txt','data8.txt','data9.txt','data10.txt','data11.txt']
length = len(data['DEWPOINT'])
Max = [10.91,6.8525,354.25,6.39,3000,8000]
Min = [7.06,0.4325,2.75,-0.9725,200,50]
for i in range(len(filename)):
    WriteData(data[attri[i]],length,Max[i],Min[i],filename[i])
'''

data = pd.read_excel('amos200313.xlsx')
attri = ['PAINS','QNH','QFE','TEMP','RH','DEWPOINT','WS2A','WD2A','CW2A','RVR1A','MOR1A']
filename = ['data1.txt','data2.txt','data3.txt','data4.txt','data5.txt','data6.txt','data7.txt','data8.txt','data9.txt','data10.txt','data11.txt']
length = len(data['DEWPOINT'])
Max = [1020.6,1022.14,1020.58,20.2,100,12.94,4.635,358.25,4.5975,3000,10000,100]
Min = [1012.6,1014.14,1012.58,7.8,35,3.96,0.0775,2.5,-1.6375,100,0,10]
for i in range(len(filename)):
    WriteData(data[attri[i]],length,Max[i],Min[i],filename[i])
```

问题一 PCA 数据降维代码

```
from sklearn.decomposition import PCA
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
# data = pd.read_excel('amos191216.xlsx',sheet_name="Sheet2",usecols="D:L")
data = pd.read_excel('amos200313.xlsx',sheet_name="Sheet2",usecols="C:K")
pca = PCA(n_components=5)
components = pca.fit_transform(data)

f = open('comp.txt','a')
for component in components:
    f.write(','.join([str(i) for i in component])+'\n')
f.close()

evr = pca.explained_variance_ratio_
ev = pca.explained_variance_
sv = pca.singular_values_
```

问题一多元回归分析代码

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import datasets
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

import pandas as pd

def evaluate(y_test,y_predict):
    mse_predict = mean_squared_error(y_test, y_predict)
    mae_predict = mean_absolute_error(y_test, y_predict)
    return mse_predict, mae_predict

data = pd.read_excel('amos191216.xlsx',sheet_name="ye100")
x = data[['PC1','PC2','PC3','PC4','PC5']]
y = data['RVR1A']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1)

reg = LinearRegression()
reg.fit(x_train,y_train)
y_predict = reg.predict(x_test)

print('coefficient: ',reg.coef_) # 系数
print('intercept: ',reg.intercept_) # 截距
print('accuracy: ',reg.score(x_test,y_test))
mse,mae = evaluate(y_test,y_predict)
print("MSE,MAE: ",mse,mae)
```

问题二 ResNet18 预训练模型图像特征提取代码

```
import torch
import torch.nn as nn
import torchvision.models as models
import torchvision.transforms as transforms
from torch.autograd import Variable
from PIL import Image
import os

def get_vector(image_name):
    img = Image.open(image_name)
    t_img = Variable(normalize(to_tensor(scaler(img))).unsqueeze(0))
    my_embedding = torch.zeros(512)
    def copy_data(m, i, o):
        my_embedding.copy_(o.data.reshape(o.data.size(1)))
    h = layer.register_forward_hook(copy_data)
    model(t_img)
    h.remove()
    return my_embedding.numpy()

if __name__ == "__main__":
    model = models.resnet18(pretrained=True)
    layer = model._modules.get('avgpool')
    model.eval()
    scaler = transforms.Scale((224, 224))
    normalize = transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.
225])
    to_tensor = transforms.ToTensor()

    # dir = './image/fog2/'
    dir = './image/fog1/'
    list = os.listdir(dir)
    # f = open('features2.txt','w')
    f = open('features1.txt','w')
    for i in range(0,len(list)):
        path = os.path.join(dir,list[i])
        print(path)
        vet = get_vector(path)
        f.write(list[i]+',')
        for j in vet:
            f.write(str(j)+',')
        f.write('\n')
    f.close()
```

问题二随机森林能见度估计模型代码

```
import pandas as pd
import numpy as np
import random
import math
from sklearn.externals.joblib import Parallel, delayed


class Tree(object):
    def __init__(self):
        self.split_feature = None
        self.split_value = None
        self.leaf_value = None
        self.tree_left = None
        self.tree_right = None

    def calc_predict_value(self, dataset):
        if self.leaf_value is not None:
            return self.leaf_value
        elif dataset[self.split_feature] <= self.split_value:
            return self.tree_left.calc_predict_value(dataset)
        else:
            return self.tree_right.calc_predict_value(dataset)

    def describe_tree(self):
        if not self.tree_left and not self.tree_right:
            leaf_info = "{leaf_value:" + str(self.leaf_value) + "}"
            return leaf_info
        left_info = self.tree_left.describe_tree()
        right_info = self.tree_right.describe_tree()
        tree_structure = "{split_feature:" + str(self.split_feature) + \
                         ",split_value:" + str(self.split_value) + \
                         ",left_tree:" + left_info + \
                         ",right_tree:" + right_info + "}"
        return tree_structure


class RandomForestRegression(object):
    def __init__(self, n_estimators=10, max_depth=-1, min_samples_split=2, min_sample
s_leaf=1,
                 min_split_gain=0.0, colsample_bytree=None, subsample=0.8, random_
state=None):
        self.n_estimators = n_estimators
```

```
        self.max_depth = max_depth if max_depth != -1 else float('inf')
        self.min_samples_split = min_samples_split
        self.min_samples_leaf = min_samples_leaf
        self.min_split_gain = min_split_gain
        self.colsample_bytree = colsample_bytree
        self.subsample = subsample
        self.random_state = random_state
        self.trees = None
        self.feature_importances_ = dict()

    def fit(self, dataset, targets):
        targets = targets.to_frame(name='label')
        if self.random_state:
            random.seed(self.random_state)
        random_state_stages = random.sample(range(self.n_estimators), self.n_estimators)

        if self.colsample_bytree == "sqrt":
            self.colsample_bytree = int(len(dataset.columns) ** 0.5)
        elif self.colsample_bytree == "log2":
            self.colsample_bytree = int(math.log(len(dataset.columns)))
        else:
            self.colsample_bytree = len(dataset.columns)

        self.trees = Parallel(n_jobs=-1, verbose=0, backend="threading")(
            delayed(self._parallel_build_trees)(dataset, targets, random_state)
                for random_state in random_state_stages)

    def _parallel_build_trees(self, dataset, targets, random_state):
        subcol_index = random.sample(dataset.columns.tolist(), self.colsample_bytree)
        dataset_stage = dataset.sample(n=int(self.subsample * len(dataset)), replace=Tru
e,
        random_state=random_state).reset_index(drop=True)
        dataset_stage = dataset_stage.loc[:, subcol_index]
        targets_stage = targets.sample(n=int(self.subsample * len(dataset)), replace=True,

        random_state=random_state).reset_index(drop=True)

        tree = self._build_single_tree(dataset_stage, targets_stage, depth=0)
        print(tree.describe_tree())
        return tree

    def _build_single_tree(self, dataset, targets, depth):
        if len(targets['label'].unique()) <= 1 or dataset.__len__() <= self.min_samples_s
plit:
```

```
                tree = Tree()
                tree.leaf_value = self.calc_leaf_value(targets['label'])
                return tree

            if depth < self.max_depth:
                best_split_feature, best_split_value, best_split_gain = self.choose_best_featu
re(dataset, targets)
                left_dataset, right_dataset, left_targets, right_targets = \
                    self.split_dataset(dataset, targets, best_split_feature, best_split_value)

                tree = Tree()
                if left_dataset.__len__() <= self.min_samples_leaf or \
                        right_dataset.__len__() <= self.min_samples_leaf or \
                        best_split_gain <= self.min_split_gain:
                    tree.leaf_value = self.calc_leaf_value(targets['label'])
                    return tree
                else:
                    self.feature_importances_[best_split_feature] = \
                        self.feature_importances_.get(best_split_feature, 0) + 1

                    tree.split_feature = best_split_feature
                    tree.split_value = best_split_value
                    tree.tree_left = self._build_single_tree(left_dataset, left_targets, depth+
1)
                    tree.tree_right = self._build_single_tree(right_dataset, right_targets, dep
th+1)
                    return tree
            else:
                tree = Tree()
                tree.leaf_value = self.calc_leaf_value(targets['label'])
                return tree

        def choose_best_feature(self, dataset, targets):
            best_split_gain = float("inf")
            best_split_feature = None
            best_split_value = None

            for feature in dataset.columns:
                if dataset[feature].unique().__len__() <= 100:
                    unique_values = sorted(dataset[feature].unique().tolist())
                else:
                    unique_values = np.unique([np.percentile(dataset[feature], x)
                                               for x in np.linspace(0, 100, 100)])
                for split_value in unique_values:
```

```
                    left_targets = targets[dataset[feature] <= split_value]
                    right_targets = targets[dataset[feature] > split_value]
                    split_gain = self.calc_r2(left_targets['label'], right_targets['label'])

                    if split_gain < best_split_gain:
                        best_split_feature = feature
                        best_split_value = split_value
                        best_split_gain = split_gain
        return best_split_feature, best_split_value, best_split_gain

    @staticmethod
    def calc_leaf_value(targets):
        return targets.mean()

    @staticmethod
    def calc_r2(left_targets, right_targets):
        r2 = 0
        for targets in [left_targets, right_targets]:
            mean = targets.mean()
            for dt in targets:
                r2 += (dt - mean) ** 2
        return r2

    @staticmethod
    def split_dataset(dataset, targets, split_feature, split_value):
        left_dataset = dataset[dataset[split_feature] <= split_value]
        left_targets = targets[dataset[split_feature] <= split_value]
        right_dataset = dataset[dataset[split_feature] > split_value]
        right_targets = targets[dataset[split_feature] > split_value]
        return left_dataset, right_dataset, left_targets, right_targets

    def predict(self, dataset):
        res = []
        for _, row in dataset.iterrows():
            pred_list = []
            for tree in self.trees:
                pred_list.append(tree.calc_predict_value(row))
            res.append(sum(pred_list) * 1.0 / len(pred_list))
        return np.array(res)


if __name__ == '__main__':
    df = pd.read_excel("features.xlsx").fillna(-1)
    df = df.rename(columns={'RVR1A': 'label'})
```

```
    headers = pd.read_excel("features.xlsx",usecols="E:SV",nrows=0)
    feature_list = []
    for i in headers:
        feature_list.append(i)
    clf = RandomForestRegression(n_estimators=100,
                                 max_depth=10,
                                 min_samples_split=50,
                                 min_samples_leaf=10,
                                 min_split_gain=0.0,
                                 colsample_bytree="sqrt",
                                 subsample=0.8,
                                 random_state=66)
    train_count = int(len(df)-1)
    clf.fit(df.loc[:train_count, feature_list], df.loc[:train_count, 'label'])

    y_pred = clf.predict(df.loc[:train_count, feature_list])
    print(y_pred)
    f = open("ypred.txt",'w')
    for i in y_pred:
        f.write(str(i)+'\n')
    f.close()
```

问题二 LSTM 神经网络模型代码

```
import torch
import torch.nn as nn

import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler

df = pd.read_excel("features.xlsx",sheet_name="result").fillna(-1)
residuals1 = df.loc[:len(df)-1,"RVRResi"].values.astype(float)
residuals2 = df.loc[:len(df)-1,"MORResi"].values.astype(float)
x = df.loc[:len(df)-1,"NUM"]

train_data = residuals1
test_data = residuals1
scaler = MinMaxScaler(feature_range=(-1, 1))
train_data_normalized = scaler.fit_transform(train_data .reshape(-1, 1))

train_data_normalized = torch.FloatTensor(train_data_normalized).view(-1)
train_window = 8
```

```
def create_inout_sequences(input_data, tw):
    inout_seq = []
    L = len(input_data)
    for i in range(L-tw):
        train_seq = input_data[i:i+tw]
        train_label = input_data[i+tw:i+tw+1]
        inout_seq.append((train_seq ,train_label))
    return inout_seq
train_inout_seq = create_inout_sequences(train_data_normalized, train_window)
print(train_inout_seq[:5])

class LSTM(nn.Module):
    def __init__(self, input_size=1, hidden_layer_size=100, output_size=1):
        super().__init__()
        self.hidden_layer_size = hidden_layer_size
        self.lstm = nn.LSTM(input_size, hidden_layer_size)
        self.linear = nn.Linear(hidden_layer_size, output_size)
        self.hidden_cell = (torch.zeros(1,1,self.hidden_layer_size),
                            torch.zeros(1,1,self.hidden_layer_size))
    def forward(self, input_seq):
        lstm_out, self.hidden_cell = self.lstm(input_seq.view(len(input_seq) ,1, -1), self.
hidden_cell)
        predictions = self.linear(lstm_out.view(len(input_seq), -1))
        return predictions[-1]
model = LSTM()
loss_function = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
print(model)
epochs = 150
files = open("epochLoss.txt",'w')
for i in range(epochs):
    for seq, labels in train_inout_seq:
        optimizer.zero_grad()
        model.hidden_cell = (torch.zeros(1, 1, model.hidden_layer_size),
                             torch.zeros(1, 1, model.hidden_layer_size))
        y_pred = model(seq)
        single_loss = loss_function(y_pred, labels)
        single_loss.backward()
        optimizer.step()
    if i%25 == 1:
        print(f'epoch: {i:3} loss: {single_loss.item():10.8f}')
        files.write(f'epoch: {i:3},loss: {single_loss.item():10.8f}'+'\n')
print(f'epoch: {i:3} loss: {single_loss.item():10.10f}')
```

```
files.write(f'epoch: {i:3},loss: {single_loss.item():10.8f}'+'\n')
files.close()
test_inputs = train_data_normalized[-train_window:].tolist()
model.eval()

for i in range(len(test_data)):
    seq = torch.FloatTensor(test_inputs[-train_window:])
    with torch.no_grad():
        model.hidden = (torch.zeros(1, 1, model.hidden_layer_size),
                        torch.zeros(1, 1, model.hidden_layer_size))
        test_inputs.append(model(seq).item())
print(test_inputs[len(test_data):])
actual_predictions = scaler.inverse_transform(np.array(test_inputs[train_window:] ).reshape
(-1, 1))
print(actual_predictions)
pre = open("pred.txt",'w')
for i in actual_predictions:
    pre.write(str(i)+'\n')
pre.close()
```

问题三区域划分代码

```
import cv2
import matplotlib.pyplot as plt

path = './image/orginal/original_frame1.bmp'
img = cv2.imread(path,0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)
titles = ['(a) Orginal','(b) Binary','(c) Binary_INV','(d) Global','(e) Sky Domain','(f) Road
Domain']
images = [img,thresh1,thresh2,thresh3,thresh4,thresh5]
for i in range(6):
  plt.subplot(3,2,i+1),plt.imshow(images[i],'gray')
  plt.title(titles[i],fontdict={'family' : 'Times New Roman', 'size' : 18})
  plt.xticks([]),plt.yticks([])
plt.show()
```

问题三边缘检测代码

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

```
from skimage.data import camera
from skimage.filters import roberts, sobel, scharr, prewitt
from skimage import feature

path = './image/orginal/original_frame1.bmp'
img=cv2.imread(path)
imgG=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
imgH = cv2.equalizeHist(imgG)
image = cv2.GaussianBlur(imgH, (9, 9),0)

edge_roberts = roberts(image)
edge_sobel = sobel(image)
edge_scharr = scharr(image)
edge_prewitt = prewitt(image)
edge_canny = feature.canny(image,0,15)
```

问题三 Hough 变换代码

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread('test.bmp')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
hist = cv2.equalizeHist(gray)
blurred = cv2.GaussianBlur(hist, (9, 9),0)
edges = cv2.Canny(blurred,0,50)

plt.subplot(1,2,1)
plt.imshow(edges,'gray')
plt.xticks([])
plt.yticks([])
plt.tick_params(labelbottom=False, labelleft=False)
plt.title('(a) Canny', fontdict={'family' : 'Times New Roman', 'size' : 18})
#hough transform
lines = cv2.HoughLinesP(edges,1,np.pi/180,30,minLineLength=60,maxLineGap=10)
lines1 = lines[:,0,:]#提取为二维
path = './image/orginal/original_frame86.bmp'
img_ori = cv2.imread(path)
for x1,y1,x2,y2 in lines1[:]:
    cv2.line(img_ori,(x1,y1),(x2,y2),(255,0,0),1)
plt.subplot(1,2,2)，plt.imshow(img_ori)，plt.xticks([])，plt.yticks([])
plt.tick_params(labelbottom=False, labelleft=False)
plt.title('(b) Hough', fontdict={'family' : 'Times New Roman', 'size' : 18})
```

```
plt.show()
```