**武汉纺织大学**

**《神经网络与深度学习》课程报告**

**题目： Weather Prediction**

**成 绩：**

**学 号： 2226010406**

**姓 名： 李圣璇**

**班 级： 应数12202班**

**指导教师： 杜小勤**

**报告日期：2024年11月21日**

# 目录

[目录 2](#_Toc185671658)

[一、课题的相关背景 3](#_Toc185671659)

[二、相关方法与框架 4](#_Toc185671660)

[2.1 Preprocessing 4](#_Toc185671661)

[2.2 Data Distribution 5](#_Toc185671662)

[2.3 Main 5](#_Toc185671663)

[2.4 Prediction Model 6](#_Toc185671664)

[三、设计与实现 6](#_Toc185671665)

[3.1 Preprocessing 6](#_Toc185671666)

[3.2 Data Distribution 6](#_Toc185671667)

[3.3 Main 7](#_Toc185671668)

[3.3.1自定义CNN神经网络设计架构如下（model）： 7](#_Toc185671669)

[3.3.2 MobileNetV2神经网络设计架构如下（model\_mo）： 8](#_Toc185671670)

[3.3.3 Resnet50神经网络设计架构如下（model\_resnet）： 10](#_Toc185671671)

[3.3.4 Densenet神经网络设计架构如下（model\_densenet）： 11](#_Toc185671672)

[3.4 Prediction Model 12](#_Toc185671673)

[四、分析与调试 12](#_Toc185671674)

[4.1评价指标： 12](#_Toc185671675)

[4.2结果分析： 14](#_Toc185671676)

[4.2.1对准确率的图像分析： 14](#_Toc185671677)

[4.2.2对混淆矩阵的图像分析： 17](#_Toc185671678)

[4.2.3Classificaton Report对比分析： 20](#_Toc185671679)

[4.2.4 综合分析 23](#_Toc185671680)

[五、运行结果 23](#_Toc185671681)

[六、 课程回顾与总结 24](#_Toc185671682)

[6.1前馈神经网络 (FNN) 26](#_Toc185671683)

[6.2卷积神经网络 (CNN) 26](#_Toc185671684)

[6.3循环神经网络 (RNN) 27](#_Toc185671685)

[6.4自编码器 (AE) 27](#_Toc185671686)

[6.5生成对抗网络 (GAN) 28](#_Toc185671687)

[6.6深度强化学习 (DRL) 28](#_Toc185671688)

[七、心得与展望 29](#_Toc185671689)

[八、参考文献 30](#_Toc185671690)

# 一、课题的相关背景

随着气候变化和极端天气事件的频发，准确地识别和预测天气变得愈加重要。传统的天气预报方法依赖于气象站的观测数据和数值天气预报模型，但这些方法往往无法及时反映瞬息万变的天气状况。近年来，计算机视觉和深度学习技术的迅猛发展为天气图像的自动分类和分析提供了新的可能性。

本项目旨在基于自定义卷积神经网络（CNN）及复合相关特征提取器（如：MobileNetV2、Resnet50、DenseNet）构建多个综合神经网络，以实现对天气图片进行分类，以识别当前的天气状况的效果。通过训练深度学习模型，我们能够从图像中提取特征，从而实现对不同天气类型（如晴天、阴天、雨天、雪天等）的自动识别。这种基于图像的天气分类方法不仅提高了天气监测的效率，还能够在特定场景下（如无人机监测、智能家居系统等）提供实时的天气信息。

项目的主要步骤包括数据收集、数据预处理、模型构建与训练、以及模型评估与优化。通过构建多个高效的深度学习模型，希望能够实现高准确率的天气分类，为未来的天气预报和相关应用提供有力支持。同时，该项目目前由于技术限制，最终得到的精度有待进一步优化。最终，这一技术有望在气象服务、农业、旅游、交通等多个领域发挥重要作用，帮助人们更好地应对天气变化带来的影响。

# 二、相关方法与框架

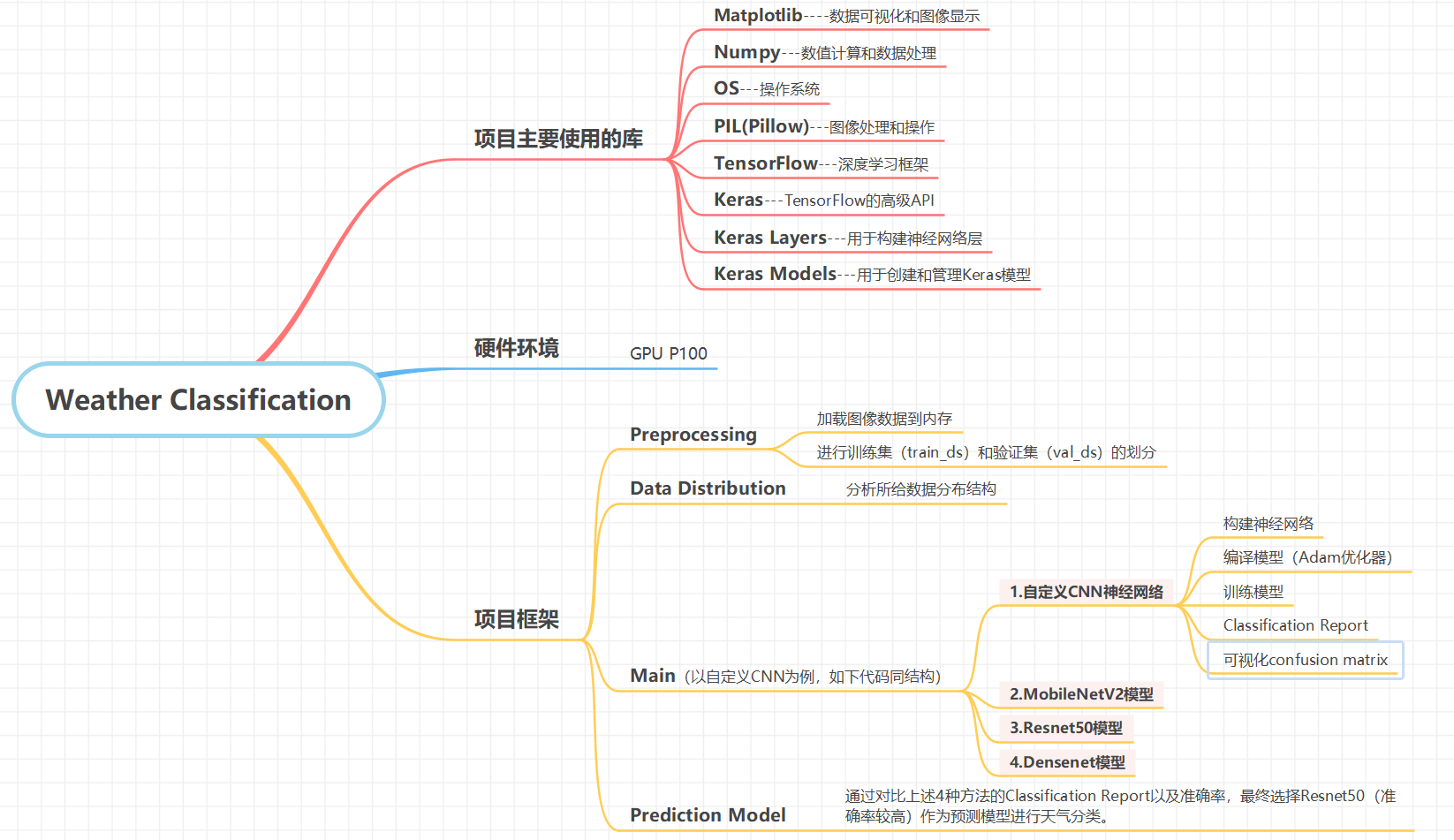


图 1项目框架图

该项目主要使用的库有：Matplotlib、Numpy、OS、PIL、TensorFlow、Keras、Keras Layers、Keras Models，分别的作用如上图所示。为了加快运行速率，选择GPU P100作为硬盘环境。该项目利用自定义CNN、MobileNetV2、ResNet50和 Densenet进行图像分类。所选模型经过量身定制，可捕捉数据集的特定特征，并且模型利用迁移学习，增强模型在不同图像中进行泛化的能力。

以下是简要的项目框架：

## 2.1 Preprocessing

首先调整图像大小以匹配模型的输入大小，由于MobileNetV2对于输入大小格式有限制（例：224\*224），于是统一设置为224\*224像素作为输入。其次将像素标准化为[0,1]的范围。最后划分数据集（train\_ds=0.8，val\_ds=0.2）。

## 2.2 Data Distribution

接着利用字典统计和可视化该多类别天气数据集中的图像分布，最终通过饼图呈现。

## 2.3 Main

在主体部分主要使用了4种不同的**自定义神经网络**进行训练，使用model.predict()方法在测试集上进行预测，并通过np.argmax()获取预测的类别标签。

为了评价模型的效果，选取如下的指标进行**模型评估**：accuracy\_score（准确率）、precision\_score（精确率）、recall\_score（召回率）和f1\_score（F1分数）计算相应的指标。之后对于每一种模型使用classification\_report打印详细的分类性能指标并计算混淆矩阵，使用 seaborn 和 matplotlib 可视化。之后部分将对这些指标进行详细的介绍。

以下是对4种模型的简要概括：

**自定义CNN**：构建和训练一个卷积神经网络（CNN）用于多类图像分类。使用 tensorflow.keras 相关模块构建和训练深度学习模型。

**MobileNetV2模型**：利用MobileNetV2进行特征提取，MobileNetV2是一种高效的卷积神经网络架构，专为移动和边缘设备设计，旨在实现高准确率和低计算成本。

**Resnet50模型**：利用Resnet50进行特征提取。ResNet50 由 50 层组成，包括多个残差块（Residual Blocks）。每个残差块通常包含两个或三个卷积层及其对应的残差连接。ResNet50 的核心思想是通过残差连接（skip connections）来解决深层网络训练中的退化问题。残差连接允许梯度直接流过网络，从而缓解梯度消失现象。

**DenseNet模型**：使用Densenet作为特征提取器。DenseNet 的核心思想是每一层都与之前的所有层相连接。这种密集连接方式允许特征在网络中更有效地传播，从而缓解了梯度消失的问题。同时，DenseNet 中使用了过渡层（Transition Layer），用于减少特征图的尺寸和通道数。这些层通常包括卷积和池化操作，有助于控制模型的复杂度。

## 2.4 Prediction Model

通过比较classification Report中各个模型结果，选取了各指标表现较为优异的Resnet50模型进行预测函数使用的主要模型，并对在网上随机找的2张图片进行预测，检验预测效果。

# 三、设计与实现

## 3.1 Preprocessing

该dataset总共分为6类，分为：alien\_test、cloudy、foggy、rainy、shine、sunrise，以及test.csv，由于数据本身属于图片类数据且数据完整，因此可以不用对原始数据进行数据预处理。因此在预处理这一步中，首先从指定目录加载图像数据，同时划分80%的数据用于训练，20%用于验证，同一设置图像的高度和宽度为224像素，并定义批量大小为16。统一定义为224\*224像素是因为后续用到的MobileNetV2模型的预训练权重为224，为了防止后续出现报错或警告，因此对所用模型统一使用该大小作为输入。接着将图像像素统一标准化为[0,1]的格式。为了使之后的代码模块看起来更整洁，在预处理处定义了用于绘制训练和验证数据的损失和准确率曲线的代码块。

## 3.2 Data Distribution

根据得到的结果发现：sunrise占比最大，rainy、foggy、cloudy占比相同，仅次于sunrise，这种分配比例有可能会对最终分类结果产生一定影响，因此数据集的优化思路可以是增大数据量的同时使得每个种类的占比均匀。

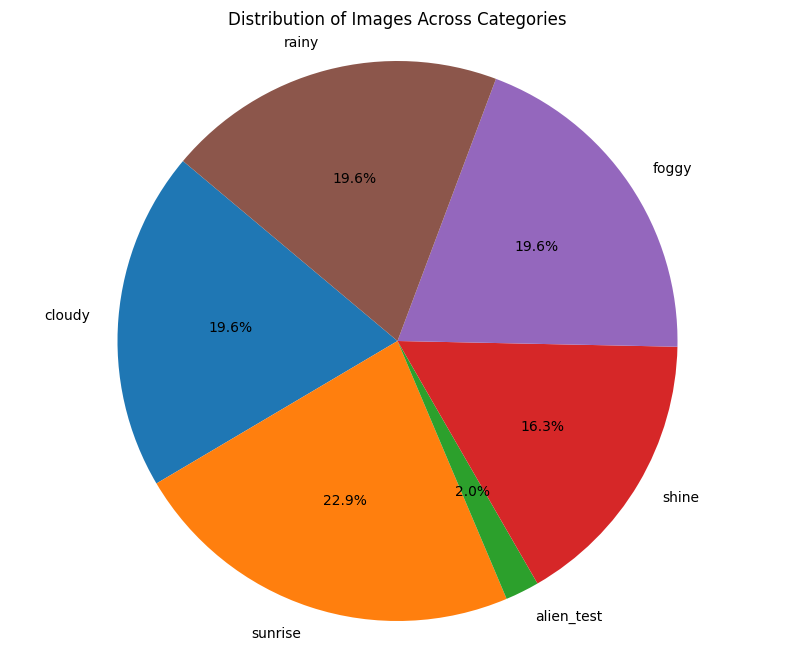


图 2数据分布图

## 3.3 Main

模型评估指标详见下一节。此处主要对4种神经网络的设计与实现进行说明。

### 3.3.1自定义CNN神经网络设计架构如下（model）：

1）输入层：定义输入图像的形状（高度、宽度和通道数）。

2）数据预处理：使用 Rescaling 将图像像素值缩放到 [0, 1] 之间。

3）卷积层和池化层：使用多个卷积层（Conv2D）提取特征，每个卷积层后面都有批归一化（BatchNormalization）以加速训练和提高稳定性。每个卷积块后面跟随一个最大池化层（MaxPooling2D）以减少特征图的尺寸，降低计算复杂度。最后使用全局平均池化层（GlobalAveragePooling2D）将特征图缩减为固定大小。

4）：全连接层：使用 Flatten 层将多维特征展平为一维。添加一个全连接层（Dense），具有 128 个神经元和 ReLU 激活函数。

5）：添加 Dropout 层以防止过拟合。

6）：输出层使用softmax激活函数，输出各个类别的概率分布。

接着对模型进行编译，这里选择Adam优化器，Adam的优势在于通过计算每个参数的自适应学习率，能够在训练过程中自动调整学习率，适应不同参数的更新需求。同时对于该多类分类问题，使用稀疏分类交叉熵作为损失函数。监控模型的准确率。之后进行模型的训练，使用fit方法训练模型，传入训练数据集和验证数据集，设置训练周期（epochs）为10。

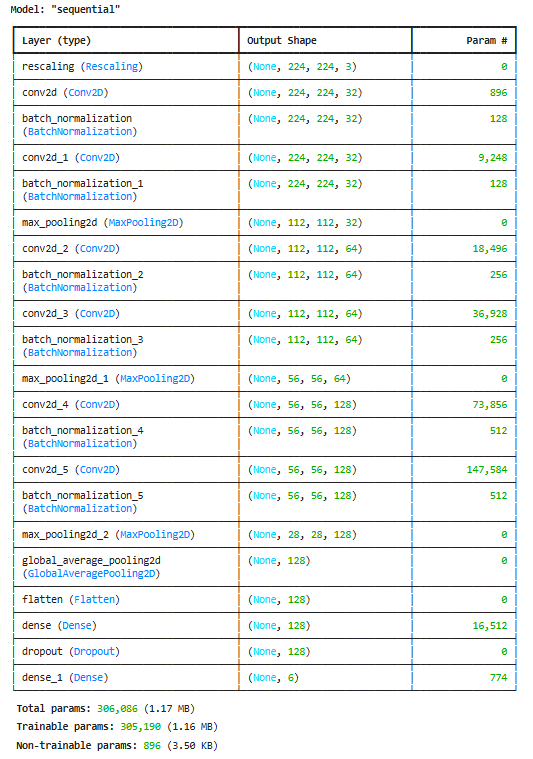


图 3 自定义CNN神经网络模型summary

### 3.3.2 MobileNetV2神经网络设计架构如下（model\_mo）：

1）输入输出：

输入形状: (img\_height, img\_width, 3)（例如，224x224x3）

输出层: include\_top=False（不包括顶部的全连接层）

预训练权重: weights='imagenet'（使用在 ImageNet 上预训练的权重）

2）全局平均池化层：GlobalAveragePooling2D，可以将特征图的空间维度平均化，生成一个一维特征向量。

3）全连接层：属于Dense层，具有神经元数量: 1024个，使用Leaky ReLU作为激活函数，可以提供非线性变换，增加模型的表示能力。

4）批归一化层：属于BatchNormalization层，加入该层可以规范化上一层的输出，加快训练速度并提高模型稳定性。

5）Dropout 层：属于Dropout层，丢弃率: 0.5，随机丢弃一部分神经元，以减少过拟合。

6）输出层：属于Dense层，设置神经元数量为num\_classes（分类的数量），设置激活函数为Softmax，最终通过输出层输出各类别的概率，进行多类分类任务。

接着设计模型，进行模型的训练。输入: base\_model.input（基础模型的输入），输出: predictions（最终输出层的预测结果）。为了增强训练效果，此处加入冻结和解冻层。冻结层：在初始训练阶段，冻结基础模型的所有卷积层，使其参数不参与训练。解冻层：在微调阶段，解冻基础模型的最后 20 层，使其参数参与训练，以进一步优化模型性能。

对于模型的编译选择Adam优化器，由于数据是整数类型存储的，因此选择sparse\_categorical\_crossentropy损失函数。最后对模型进行 50 轮训练。之后为了进一步增强模型效果，增加了微调过程：在解冻部分卷积层后，继续训练20轮以微调模型。

由于模型层数过多，此处不进行完整展示，但是在jupyter notebook中已经打印下来模型的整体summary，读者可自行进行浏览。

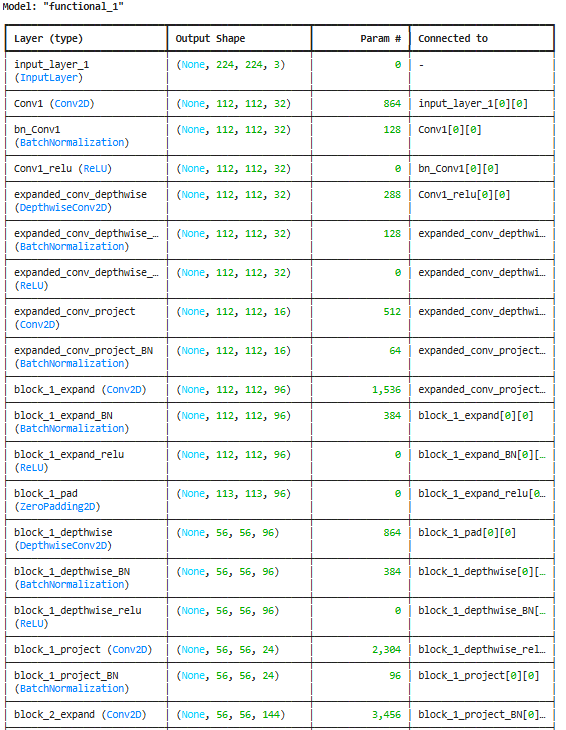


图 4 MobileNetV2模型summary

### 3.3.3 Resnet50神经网络设计架构如下（model\_resnet）：

1）模型概述：

ResNet50 是一种深度卷积神经网络，具有50层深度，专为解决深层网络训练中的梯度消失问题而设计。通过引入残差连接，ResNet50 允许信息在网络中更有效地流动，从而提升了模型的性能和训练效率。

2）模型组成：

Basemodel是基于ResNet50，输入形状为(224, 224, 3)（标准的 RGB 图像输入），输出层为include\_top=False（不包括顶部全连接层），预训练权重是weights='imagenet'（使用在 ImageNet 数据集上预训练的权重）

全局平均池化层 (GlobalAveragePooling2D)：可以将输入特征图的空间维度平均化，生成一个一维特征向量，减少参数数量并有效防止过拟合。

全连接层 (Dense)：该层神经元数量为1024个，激活函数选用ReLU，可以提供非线性变换，增强模型的表达能力。

输出层 (Dense)：该层神经元数量定义为num\_classes（根据具体任务设定的类别数量），定义激活函数是Softmax，可以输出各类别的概率，适用于多类分类任务。

3）模型训练：

在初始训练阶段，所有基础模型的卷积层被设置为不可训练。这一策略有助于保持预训练权重不变，以便在新数据集上进行迁移学习。

4）编译模型：使用Adam优化器（自适应学习率优化器），选用：SparseCategoricalCrossentropy（适用于多类分类任务的稀疏标签）作为损失函数。

5）训练过程：设定10轮的训练轮数。

由于模型层数过多，此处不进行展示，但是在jupyter notebook中已经打印下来模型的整体summary，读者可自行进行浏览。

### 3.3.4 Densenet神经网络设计架构如下（model\_densenet）：

1）模型概述：

DenseNet121 是一种高效的卷积神经网络，旨在改善深度学习模型的性能和训练效率。通过密集连接的结构，DenseNet121 允许每一层都接收前面所有层的特征图，有效缓解了梯度消失问题，并促进了特征重用。

2）模型组成：基础模型上加入DenseNet121，输入形状为(224, 224, 3)（标准 RGB 图像输入），输出层为include\_top=False（不包括顶部全连接层，以便进行自定义分类）。预训练权重定义为weights='imagenet'（使用在 ImageNet 数据集上预训练的权重）。

3）层结构:

全局平均池化层 (GlobalAveragePooling2D)：将特征图的空间维度平均化，生成一个一维特征向量。这一层减少了参数数量，并有效防止了过拟合。

全连接层 (Dense)：神经元数量为1024，使用ReLU作为激活函数，可以提供非线性变换，增强模型的表达能力，便于进行复杂的分类任务。

输出层 (Dense)：神经元数量为num\_classes（设定为 6，表示任务的类别数量），激活函数为Softmax，可以输出各类别的概率，适用于多类分类任务。

4）模型训练：在初始训练阶段，所有基础模型的卷积层被设置为不可训练。这一策略有助于保持预训练权重不变，确保模型在新数据集上进行有效的迁移学习。在训练的后期，解冻 DenseNet 的最后 20 层，使这些层的权重能够在训练过程中更新。这一策略有助于模型在特定数据集上进行微调，从而提升性能。

5）编译模型：使用Adam（自适应学习率优化器）优化器，损失函数为：SparseCategoricalCrossentropy（适用于多类分类任务的稀疏标签）。

6）训练过程：设置20轮训练轮数。

由于模型层数过多，此处不进行展示，但是在jupyter notebook中已经打印下来模型的整体summary，读者可自行进行浏览。

## 3.4 Prediction Model

最终定义prediction函数进行预测。其中prediction选用评估效果最好的Resnet50作为模型（model\_rennet）。

# 四、分析与调试

## 4.1评价指标：

在机器学习和统计学中，准确率（accuracy）、精确率（precision）、召回率（recall）、F1 分数（F1-score）和混淆矩阵（confusion matrix）是评估分类模型性能的重要指标。以下是对它们的详细解释：

1）准确率 (Accuracy)：准确率是指模型正确预测的样本数量占总样本数量的比例。

2）精确率 (Precision)：精确率是指模型预测为正类的样本中实际为正类的比例，反映了模型的准确性。

3）召回率 (Recall)：召回率是指实际为正类的样本中被模型正确预测为正类的比例，反映了模型的敏感性。

4）F1 分数 (F1 Score)：F1 分数是精确率和召回率的调和平均值，综合考虑了模型的准确性和敏感性，适用于类别不平衡的情况。

其中：

TP（True Positive）：真正例

TN（True Negative）：真负例

FP（False Positive）：假正例

FN（False Negative）：假负例

5）混淆矩阵 (Confusion Matrix)：混淆矩阵是一个表格，用于总结分类模型的预测结果。它显示了每个类别的真正例、假正例、真负例和假负例的数量。其中：行表示实际类别，列表示预测类别。

## 4.2结果分析：

### 4.2.1对准确率的图像分析：

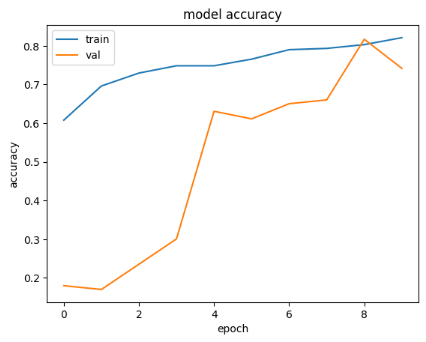


图 5 CNN比较图

从图中可以看出，随着训练轮次（epoch）的增加，训练集的准确率逐步提升，从初始的约0.3增加到接近0.8。这表明模型在训练数据上的学习效果较好，并随着训练轮次的增加而持续优化。验证集的准确率开始较低，约为0.2，但在训练到第4轮左右时有大幅提升，之后呈现波动并最终接近0.7。该模型在训练数据上表现出持续的提升，说明CNN结构能够很好地拟合训练数据。验证集准确率逐步上升，表明模型的泛化能力在逐步改善。验证集的波动可能与训练集数据分布或验证集数据复杂性有关，需要进一步优化模型或数据预处理。

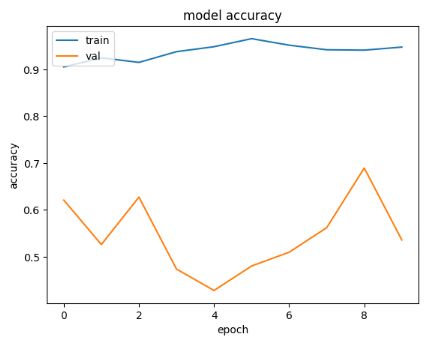


图 6 MobileNetV2比较图

训练集的准确率从初始的约0.9稍有提升，最终趋于平稳，达到接近0.95。这说明MobileNetV2在训练数据上有很强的拟合能力。验证集的准确率波动较大，初始约为0.6，在训练中期一度下降到0.4以下，随后逐步回升，但最终波动在0.6到0.7之间。训练集准确率稳定在较高水平，表明MobileNetV2模型能够很好地捕获训练数据中的模式。验证集准确率波动较大，可能反映出模型在验证数据上的泛化能力较差，存在一定的过拟合问题。可能需要通过正则化、数据增强或调整学习率等方法缓解过拟合现象，提升模型在验证集上的稳定表现。

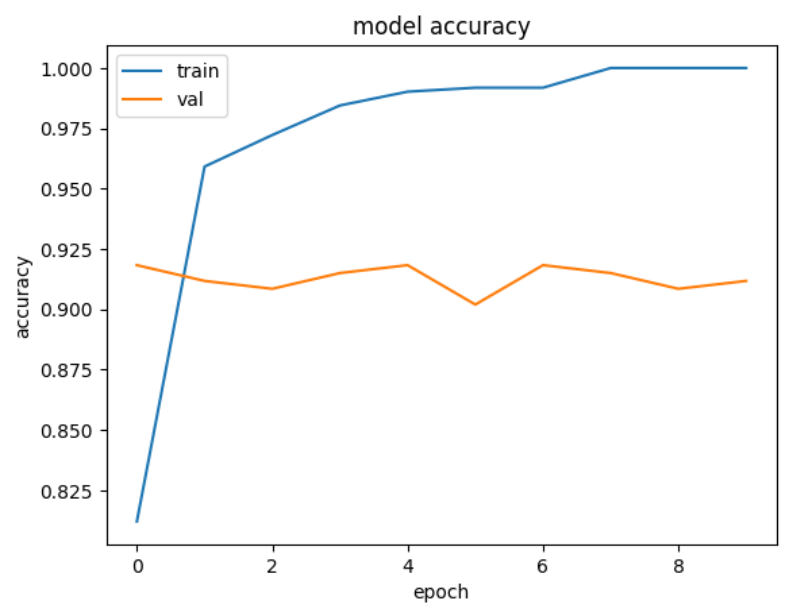


图 7 Resnet50比较图

训练集准确率（蓝色线）：在第一个epoch，准确率迅速上升，从大约0.82增加到0.95以上。随后，准确率继续稳步上升，直到接近1.0，显示出模型在训练集上表现非常好。

验证集准确率（橙色线）：验证集的准确率在第一个epoch后保持相对稳定，大约在0.92左右。

整体上，验证集的准确率没有显著提升，且波动较小，这可能表明模型在验证集上的表现没有随着训练的进行而显著改善。训练集和验证集之间的准确率差距较大，尤其是在训练后期，这可能表明模型在训练集上过拟合。

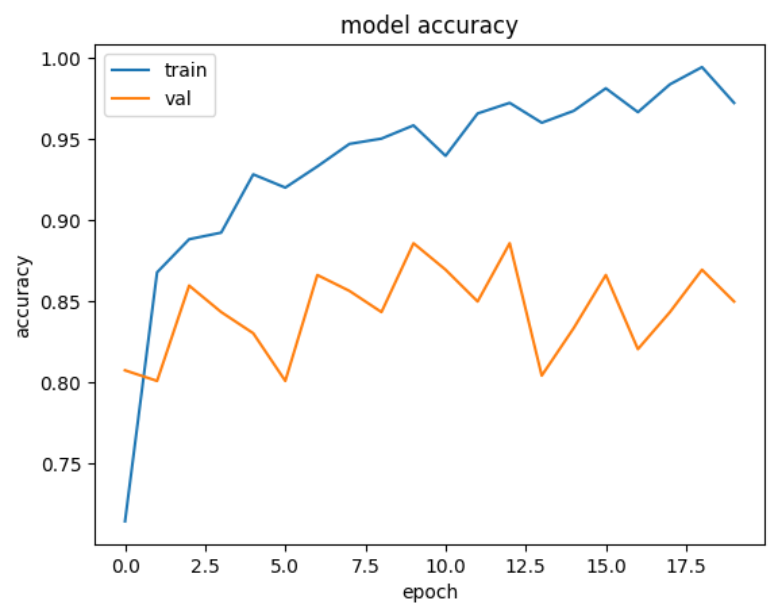


图 8 Densenet比较图

训练集的准确率在第一个epoch后迅速上升，然后在0.95左右波动。整体趋势是上升的，但存在一些波动，这可能表明模型在训练过程中存在一定的不稳定性。验证集的准确率在第一个epoch后波动较大，但总体上没有显著提升，保持在0.85左右。这种波动可能表明模型在验证集上的表现不稳定，可能受到数据集的某些特性或模型参数设置的影响。训练集和验证集之间的准确率差距较大，尤其是在训练后期，这同样可能表明模型在训练集上过拟合。

综合上述结果我们可以分析得到：

**训练集表现：**所有模型在训练集上的表现都很好，尤其是ResNet50和DenseNet，准确率接近1.0。

**验证集表现：**所有模型在验证集上的表现都不如训练集，尤其是MobileNetV2和DenseNet，验证集的准确率波动较大。

**过拟合情况：**MobileNetV2和DenseNet可能存在较明显的过拟合问题，因为训练集和验证集的准确率差距较大。ResNet50也存在一定程度的过拟合，但相对较轻。CNN的过拟合情况介于两者之间。

**稳定性：**ResNet50在验证集上的表现相对稳定，尽管有轻微的过拟合，但验证集的准确率较高。CNN的验证集准确率提升较慢，但波动较小。

### 4.2.2对混淆矩阵的图像分析：

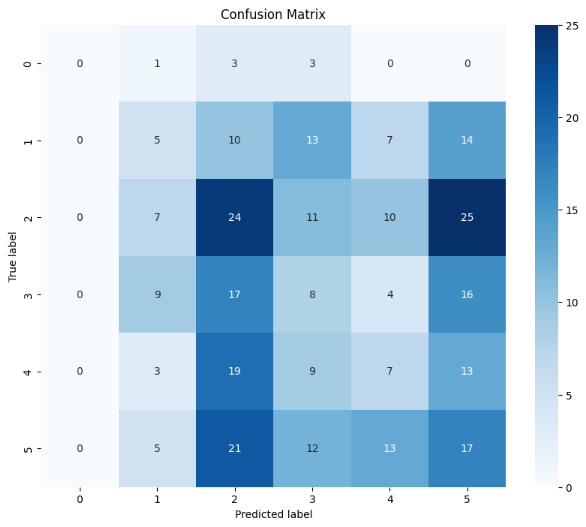


图 9 CNN混淆矩阵

根据图9得到CNN对类别2和5的预测较为准确，分别有24和21次正确预测。但是对类别1和3的预测准确率较低，尤其是类别3，有17次被错误预测为类别2。因此，总体认为该模型在某些类别上表现良好，但在其他类别上存在明显的混淆。

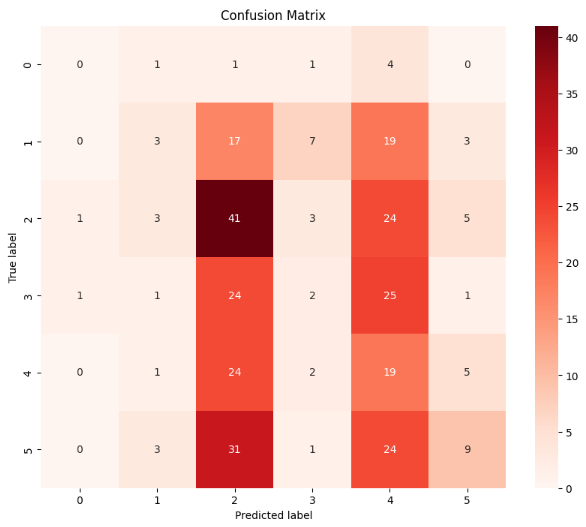


图 10 MobileNetV2混淆矩阵

根据图10得到MobileNetV2对类别2和5的预测非常准确，分别有41和31次正确预测。但是对类别1和3的预测存在较多错误，尤其是类别1，有19次被错误预测为类别3。因此，总体认为该模型在大多数类别上表现良好，但在类别1和3上存在显著的混淆。

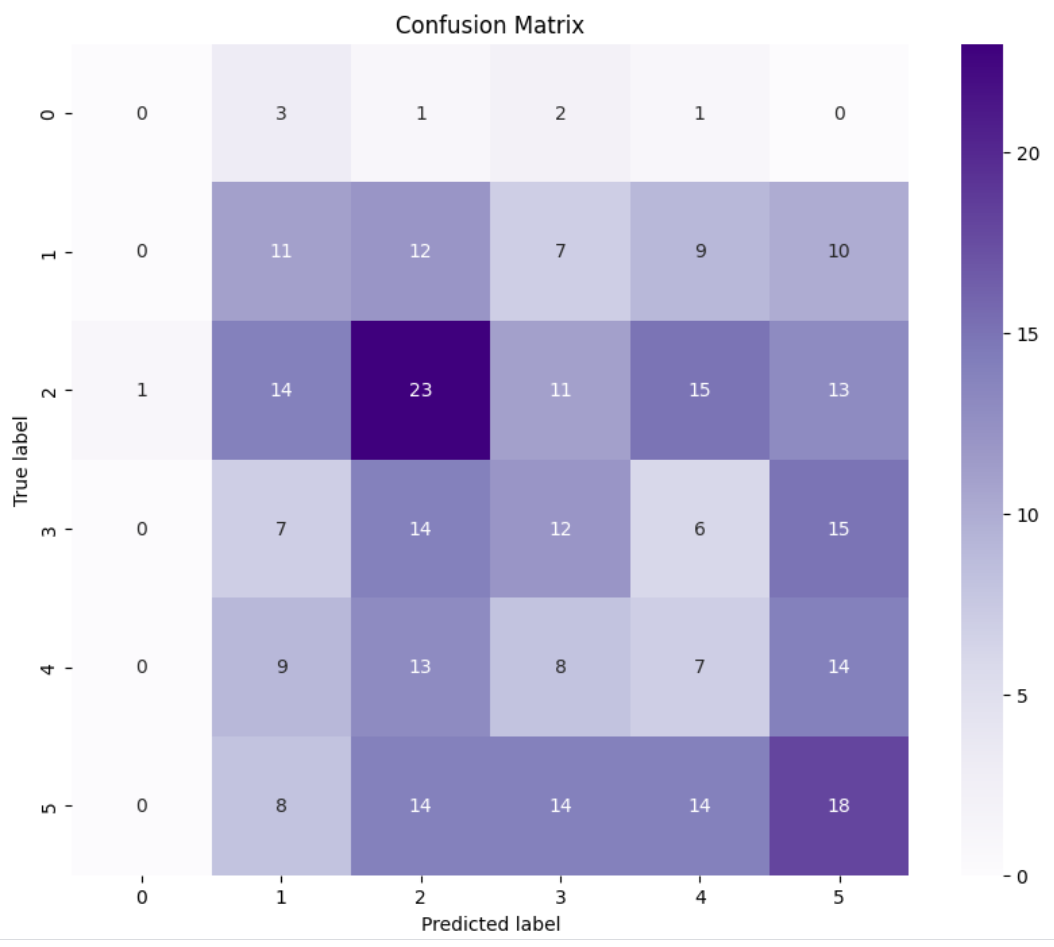


图 11 Resnet50混淆矩阵

根据图11可以得到Resnet50对所有类别的预测都相对均衡，没有明显的预测偏差。虽然每个类别的预测准确率较为均衡，但准确率普遍低于MobileNetV2，最高为类别2的23次正确预测。该模型在所有类别上的表现较为均衡，但整体准确率略低。

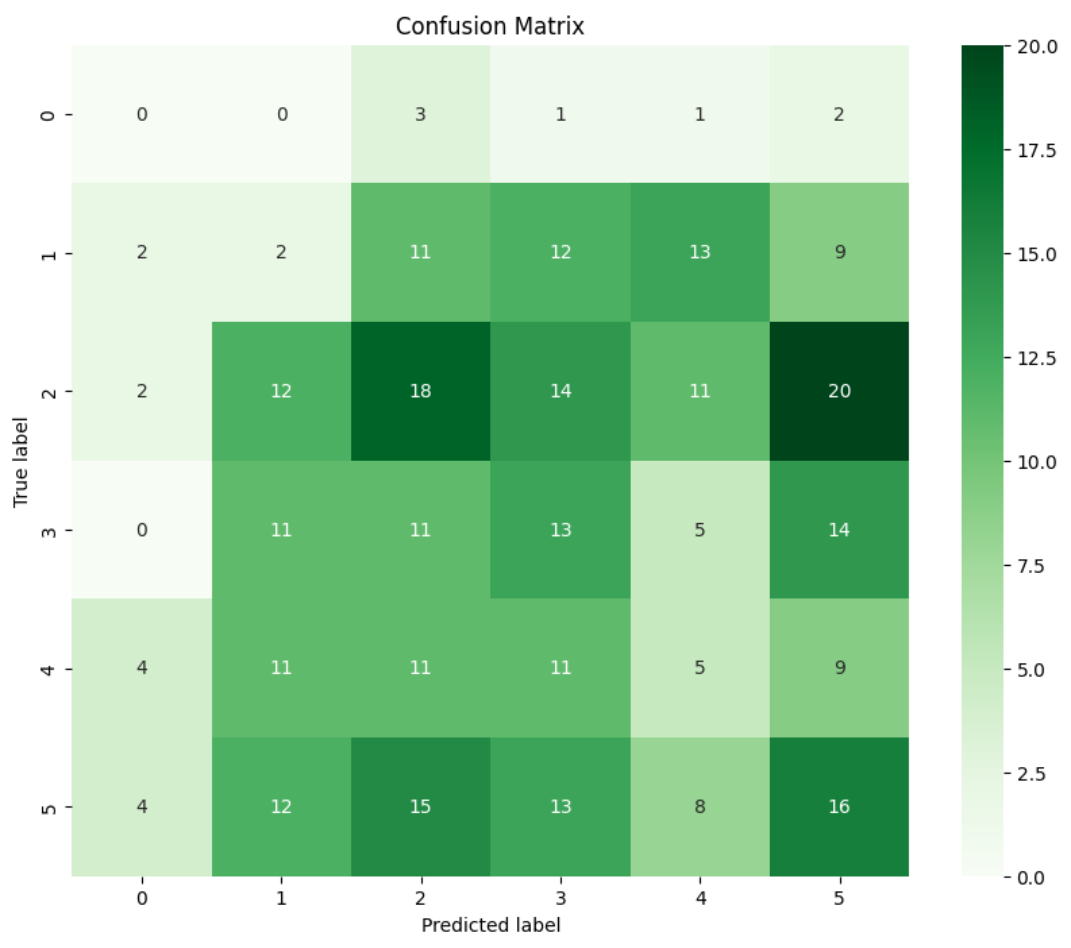


图 12 Densenet混淆矩阵

根据图12可以得到Densenet对类别2和5的预测较为准确，分别有18和16次正确预测。但是对类别1和4的预测准确率较低，尤其是类别1，有11次被错误预测为类别2。总体认为该模型在某些类别上表现良好，但在其他类别上存在混淆，整体表现与ResNet50相似。

根据以上情况得到如下对比分析：

准确率：MobileNetV2在大多数类别上的表现最好，尤其是在类别2和5上，显示出最高的准确率。

均衡性：ResNet50在所有类别上的表现最为均衡，没有明显的预测偏差，但整体准确率略低。

混淆情况：CNN和DenseNet在某些类别上表现良好，但在类别1和3（或类别1和4）上存在显著的混淆。

因此，对于CNN和DenseNet，可能需要进一步优化模型或调整数据集，以减少在特定类别上的混淆。对于MobileNetV2，尽管整体表现最好，但仍然可以探索方法来减少类别1和3之间的混淆。

### 4.2.3Classificaton Report对比分析：

表 1 CNN指标结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 精确率(Precision) | 召回率(Recall) | F1-score | 支持(Support) |
| 0 | 0.00 | 0.00 | 0.00 | 7 |
| 1 | 0.17 | 0.10 | 0.13 | 49 |
| 2 | 0.26 | 0.31 | 0.28 | 77 |
| 3 | 0.14 | 0.15 | 0.15 | 54 |
| 4 | 0.17 | 0.14 | 0.15 | 51 |
| 5 | 0.20 | 0.25 | 0.22 | 68 |
| accuracy:0.7168 loss:1.0029 | | | | |

针对CNN模型分类别进行分析，其中：

类别0：模型完全未能识别类别0，所有指标都是0。类别1：精确率和召回率都很低，F1分数为0.13，表明模型在这一类别上的表现不佳。

类别2：表现相对较好，F1分数为0.28，是所有类别中最高的，但仍然有改进空间。

类别3：精确率和召回率都很低，F1分数为0.15，表明模型在这一类别上的表现不佳。

类别4：与类别1和3类似，F1分数为0.15，表现不佳。

类别5：F1分数为0.22，表现略好于类别1、3和4，但仍然不理想。

综上得到CNN模型的准确率为0.7168，损失为1.0029，表明模型在整体上有一定的识别能力，但在某些类别上的表现非常差。

表 2 MobileNetV2指标结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 精确率(Precision) | 召回率(Recall) | F1-score | 支持(Support) |
| 0 | 0.00 | 0.00 | 0.00 | 7 |
| 1 | 0.25 | 0.06 | 0.10 | 49 |
| 2 | 0.30 | 0.53 | 0.38 | 77 |
| 3 | 0.12 | 0.04 | 0.06 | 54 |
| 4 | 0.17 | 0.37 | 0.23 | 51 |
| 5 | 0.39 | 0.13 | 0.20 | 68 |
| accuracy:0.7523 loss:0.9010 | | | | |

针对MobileNetV2模型分类别进行分析，其中：

类别0：与CNN类似，模型完全未能识别类别0。

类别1：精确率有所提高，但召回率仍然很低，F1分数为0.10。

类别2：表现显著提升，F1分数为0.38，是所有类别中最高的。

类别3：精确率和召回率都很低，F1分数为0.06，表现不佳。

类别4：精确率和召回率有所提高，F1分数为0.23，表现一般。

类别5：精确率提高，但召回率下降，F1分数为0.20。

总的来说，MobileNetV2的准确率为0.7523，损失为0.9010，表明模型在整体上比CNN有轻微的提升，尤其是在类别2上。

表 3 Resnet50指标结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 精确率(Precision) | 召回率(Recall) | F1-score | 支持(Support) |
| 0 | 0.00 | 0.00 | 0.00 | 7 |
| 1 | 0.21 | 0.22 | 0.22 | 49 |
| 2 | 0.30 | 0.30 | 0.30 | 77 |
| 3 | 0.22 | 0.22 | 0.22 | 54 |
| 4 | 0.13 | 0.14 | 0.14 | 51 |
| 5 | 0.26 | 0.26 | 0.26 | 68 |
| accuracy:0.7650 loss:0.9185 | | | | |

针对Resnet50模型分类别进行分析，其中：

类别0：与前两个模型一样，未能识别类别0。

类别1：精确率和召回率都有所提高，F1分数为0.22。

类别2：表现良好，F1分数为0.30，是所有类别中最高的。

类别3：精确率和召回率都有所提高，F1分数为0.22。

类别4：精确率和召回率都有所提高，F1分数为0.14。

类别5：精确率和召回率都有所提高，F1分数为0.26。

针对Resnet50模型来看，准确率为0.7650，损失为0.9185，表明模型在整体上比CNN和MobileNetV2有提升，尤其是在类别2和类别5上。

表 4 Densenet指标结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | 精确率(Precision) | 召回率(Recall) | F1-score | 支持(Support) |
| 0 | 0.00 | 0.00 | 0.00 | 7 |
| 1 | 0.04 | 0.04 | 0.04 | 49 |
| 2 | 0.26 | 0.23 | 0.25 | 77 |
| 3 | 0.20 | 0.24 | 0.22 | 54 |
| 4 | 0.12 | 0.10 | 0.11 | 51 |
| 5 | 0.23 | 0.24 | 0.23 | 68 |
| accuracy:0.7391 loss:0.8502 | | | | |

针对Densenet模型来看，其中：

类别0：未能识别类别0，所有指标都是0。

类别1：精确率和召回率都很低，F1分数为0.04。

类别2：表现相对较好，F1分数为0.25。

类别3：精确率和召回率都有所提高，F1分数为0.22。

类别4：精确率和召回率都很低，F1分数为0.11。

类别5：精确率和召回率都有所提高，F1分数为0.23。

总的来说，Densenet的准确率为0.7391，损失为0.8502，表明模型在整体上比CNN稍好，但不如MobileNetV2和ResNet50。

综合上述4种方法的指标分析，可以发现所有模型在类别0上的表现都很差，这可能是由于样本数量少或者模型对该类别的特征学习不足。所有模型在类别2上的表现都相对较好，尤其是MobileNetV2和ResNet50。ResNet50在大多数类别上都显示出较好的精确率和召回率，这表明它在识别正例和减少假阳性方面做得较好。F1分数是精确率和召回率的调和平均，ResNet50在大多数类别上都显示出较好的F1分数，这表明它在平衡精确率和召回率方面做得较好。DenseNet的损失最低，这表明它在训练过程中的误差最小，但准确率并不是最高，这可能意味着它在某些类别上过于保守。通过分析发现：ResNet50在整体性能上表现最佳，尤其是在类别2上，同时在其他类别上也保持了较好的性能。MobileNetV2在类别2上的表现也非常出色，但整体准确率略低于ResNet50。CNN和DenseNet在某些类别上的表现不佳，需要进一步优化。所有模型在类别0上的表现都很差，这需要后续进行进一步优化。

### 4.2.4 综合分析

综合上述三种分析（准确率图像分析、混淆矩阵分析、指标分析），最终发现在4种方法中Resnet50的表现较为稳定，综合性能较为优异。虽然仍有一定的不足，后续需要进行进一步优化调整。

# 五、运行结果

综合上述评估，最终选用Resnet50作为训练模型。将输入图像调整为224\*224像素大小后，转换为numpy数组，接着调整图像形状为（1，224，224，3），使用model\_resnet进行预测，找到预测结果中最大概率的索引，表示预测的类别，最后打印置信度和对应类别名称。创建picture文件夹，从网页下载两张有关天气的图像，之后输入程序中即可得到对应的预测结果。两张运行结果如下所示：

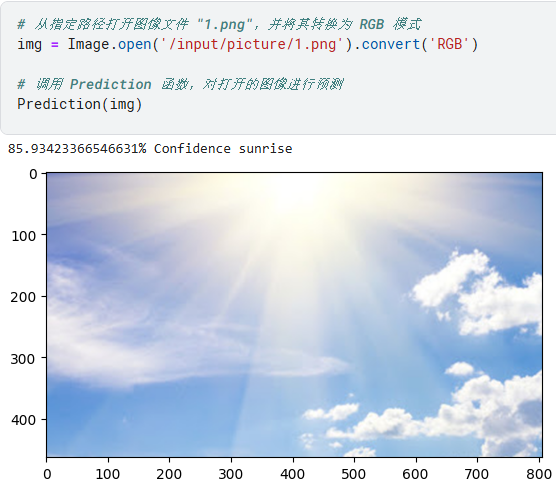


图 13运行结果1

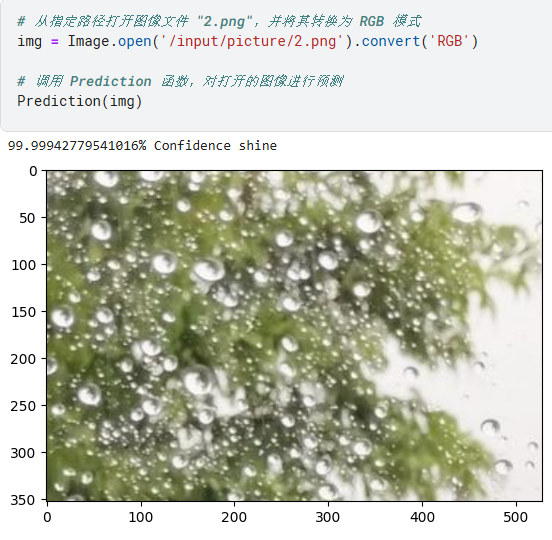
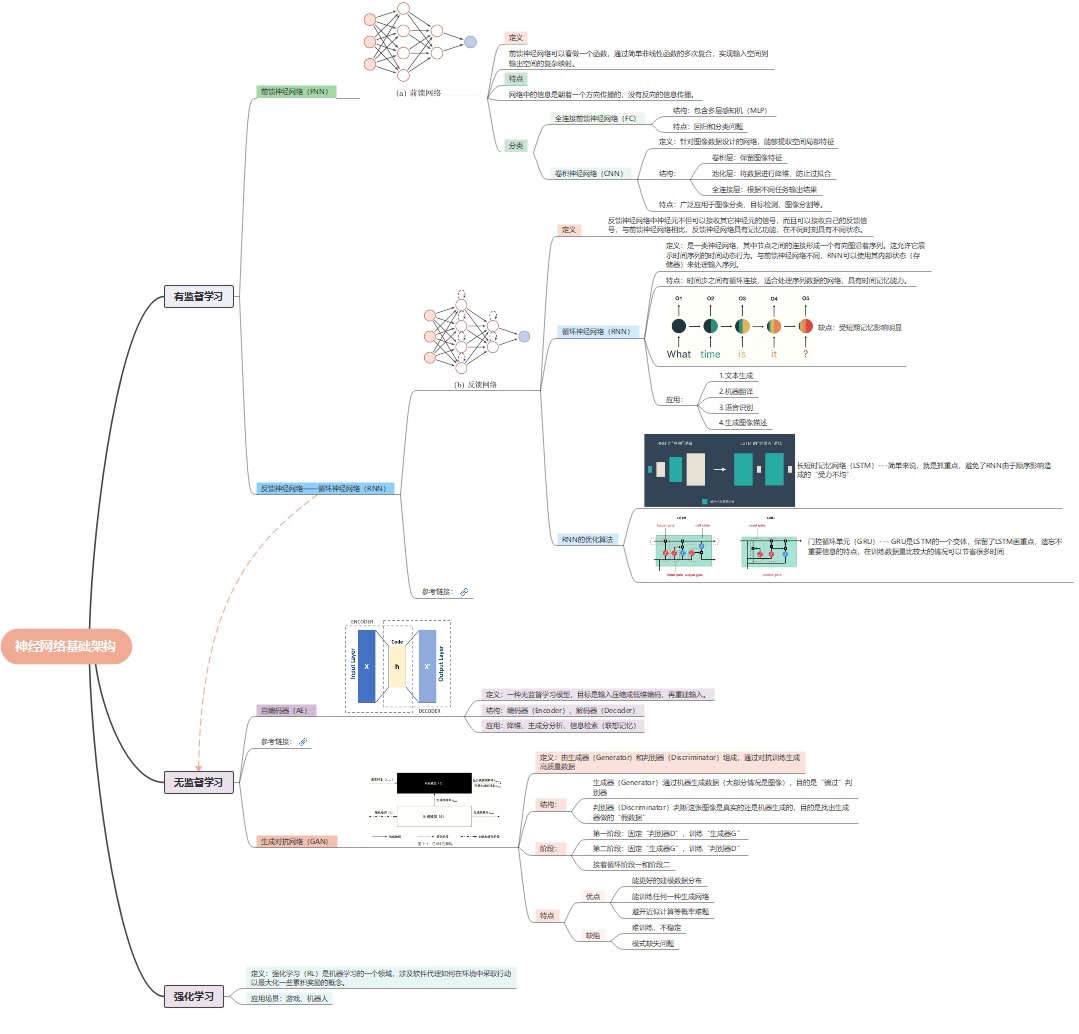


图 14 运行结果2

# 课程回顾与总结

以下自制的思维导图中从有监督学习、无监督学习、强化学习的分类体系出发，分别介绍了监督学习中的：前馈神经网络（全连接神经网络、卷积神经网络）、反馈神经网络（循环神经网络）、无监督学习中的：自编码器、生成式对抗网络以及强化学习，主要分为定义、特点、分类、结构、应用极大板块进行介绍。其中参考的网页链接附在—参考链接中。由于页面限制，可放大观看细节。



**图 15神经网络整体框架图**

在学习神经网络的过程中，前馈神经网络（FNN）、卷积神经网络（CNN）、循环神经网络（RNN）、生成对抗网络（GAN）、自编码器（AE）和深度强化学习（DRL）等构成了核心内容。这些网络类型各自有其特点和应用场景，但在学习和实践中也伴随着许多重点与难点问题。以下是一些相关的重难点总结：

## 6.1前馈神经网络 (FNN)

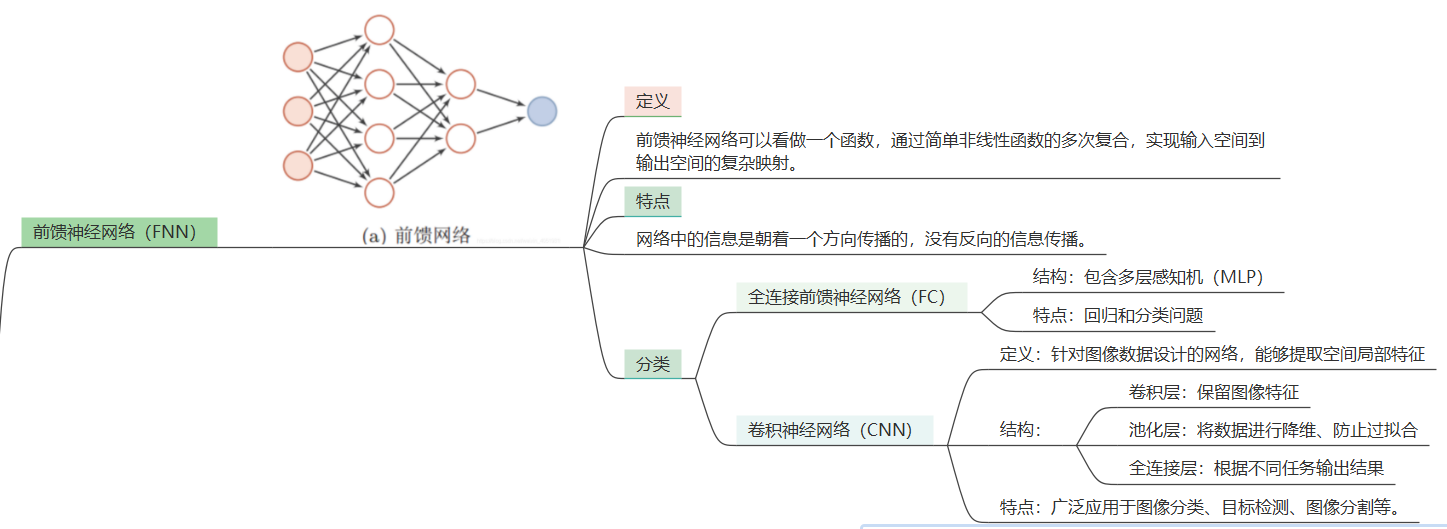


图 16 FNN框架

前馈神经网络是神经网络的基础，主要用于有监督学习任务。其重点在于理解网络的基本结构（如输入层、隐藏层和输出层）以及激活函数（如 ReLU、Sigmoid、Tanh）的作用。此外，反向传播算法是 FNN 的核心，通过优化损失函数（如均方误差或交叉熵）来调整权重。学习中需要关注超参数调节，如学习率、隐藏层数量和神经元数量。然而，FNN 存在梯度消失与梯度爆炸问题，尤其是在深层网络中。此外，过拟合问题也是一个常见难点，需要通过正则化、Dropout 或数据增强等方法解决。

## 6.2卷积神经网络 (CNN)

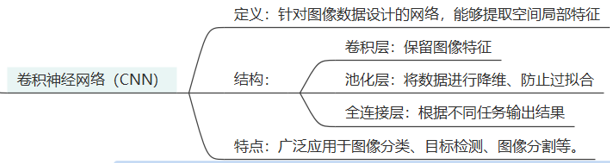


图 17 CNN框架

CNN 是处理图像数据的核心模型，其重点在于卷积操作和池化操作的理解。卷积核通过提取局部特征生成特征图，而池化操作降低特征图维度以保留关键信息。学习者还需熟悉经典网络架构（如 LeNet、AlexNet、VGG 和 ResNet），以及它们在图像分类、目标检测和图像分割中的应用。然而，CNN 的学习难点在于超参数选择（如卷积核大小、步幅和填充）以及深层网络的训练问题。深层网络可能出现梯度消失问题，需要借助残差网络（ResNet）等技术解决。此外，CNN 通常需要大量标注数据，数据不足时需要通过迁移学习或数据增强方法进行补充。

## 6.3循环神经网络 (RNN)

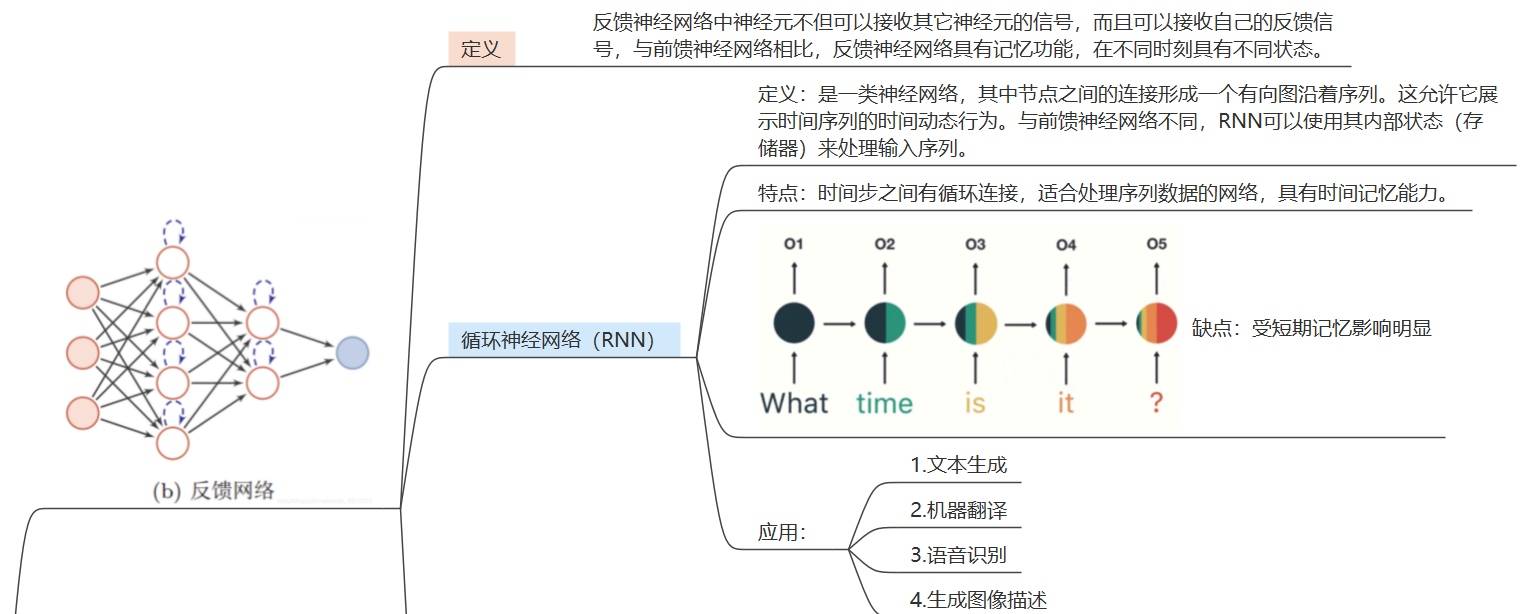


图 18 RNN框架

RNN 擅长处理时间序列数据，其重点在于理解隐藏状态的概念，以及如何通过时间步捕获序列数据的依赖信息。学习中需要掌握 LSTM 和 GRU 等变种网络，它们通过门控机制解决了 RNN 的长时依赖问题。RNN 的难点主要集中在梯度消失和梯度爆炸问题上，尤其是在处理长序列数据时。此外，RNN 的训练效率较低，时间复杂度较高，在实验过程中也感受到了代码的执行效率较低。

## 6.4自编码器 (AE)

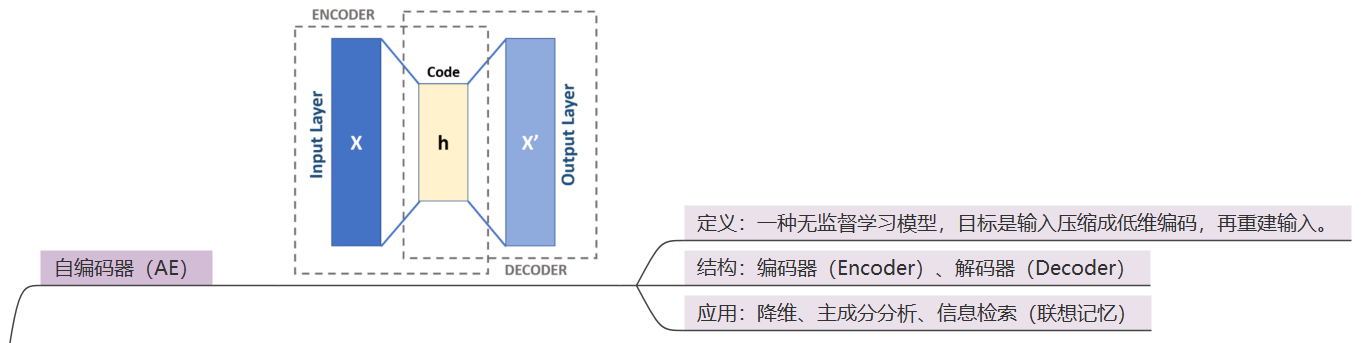


图 19 AE框架

自编码器是一种无监督学习模型，重点在于编码器和解码器的结构设计，以及其在特征提取、降维和去噪中的应用。变分自编码器（VAE）则进一步具备数据生成能力。然而，自编码器的难点在于如何优化重建误差以提升模型的表现，同时避免模型只记住输入数据而未能学到通用特征。

## 6.5生成对抗网络 (GAN)

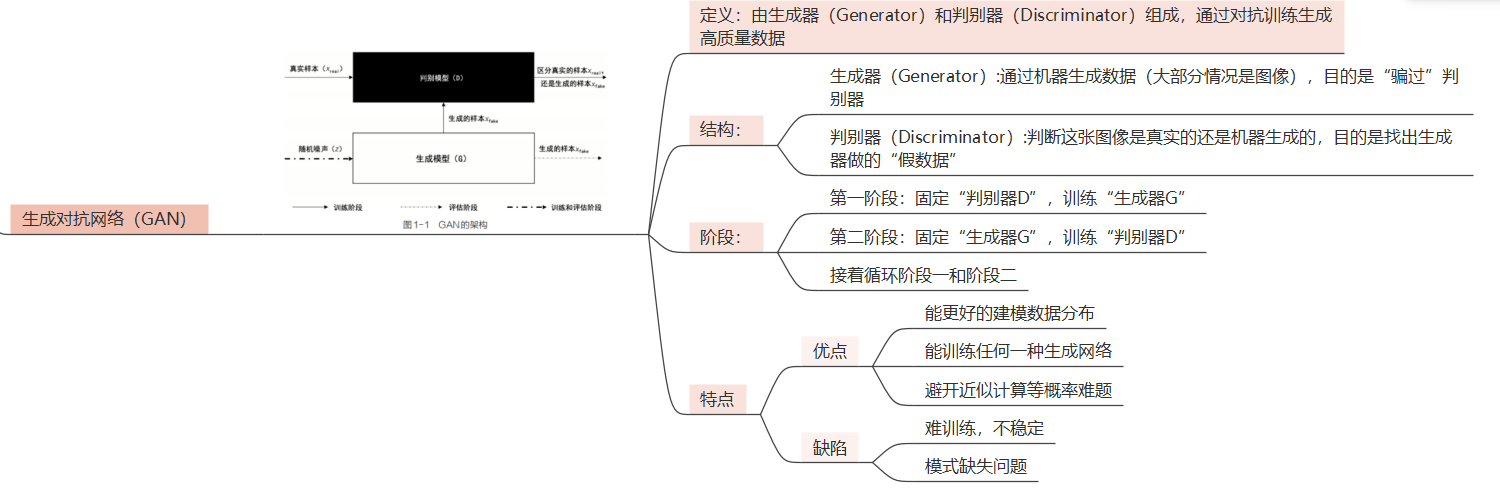


图 20 GAN框架

GAN 是近年来的热门研究方向，其重点在于生成器和判别器之间的对抗训练过程。学习者需要理解 GAN 的损失函数设计及其优化目标，掌握如何平衡生成器和判别器的训练速度。GAN 的难点在于训练过程的不稳定性，可能出现模式崩溃（Mode Collapse）和收敛问题。为解决这些问题，学习者需要了解 WGAN、DCGAN 等改进版本的技术细节。

## 6.6深度强化学习 (DRL)

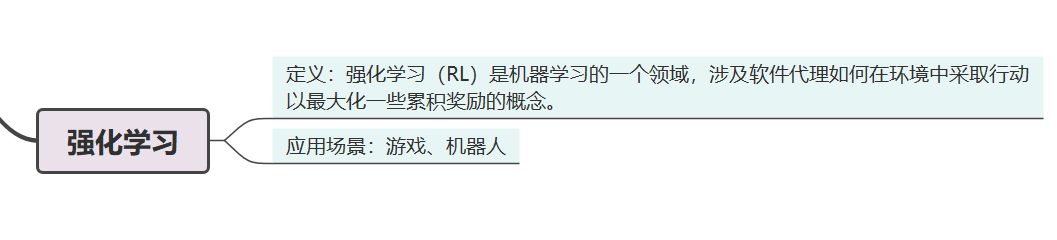


图 21 DRL框架

深度强化学习结合了强化学习和深度学习的优势，重点在于理解状态（State）、动作（Action）和奖励（Reward）之间的关系，以及值函数和策略的概念。DQN（Deep Q-Network）是 DRL 的经典算法之一。然而，DRL 的难点在于其样本效率低，训练不稳定，且在高维状态空间中难以处理复杂任务。学习者需要掌握策略梯度方法、Actor-Critic 方法等技术，探索如何提高样本效率和模型稳定性。

# 七、心得与展望

在参与基于深度学习的天气图像分类项目过程中，我深刻体会到神经网络在图像处理和模式识别中的强大潜力。以下是我在学习和实践中获得的一些心得体会。

**1. 深度学习的基础知识**

在项目初期，学习了卷积神经网络（CNN）的基本原理。通过对不同网络架构（如MobileNetV2、ResNet50和DenseNet）的理解，我逐渐意识到，每种模型都有其独特的优势和适用场景。特别是ResNet的残差连接机制，通过缓解梯度消失问题，使得深层网络的训练变得更加高效。这种深入的理解让我在选择和调优模型时更加得心应手。

**2. 数据预处理的重要性**

在数据预处理阶段，我认识到数据的质量直接影响模型的表现。调整图像大小、标准化像素值以及合理划分训练集和验证集是确保模型能够有效学习的基础。通过对数据集的分布进行可视化，我深刻理解了数据不均衡问题的潜在影响，这促使我在后续阶段更加关注数据增强技术，以提高模型的泛化能力。

**3. 实践中的挑战与解决**

在模型训练过程中，我遇到了过拟合的问题。通过引入正则化手段和 dropout 层，我成功减轻了这一问题。并且尝试在卷积层后添加 layers.BatchNormalization()，可以加速训练并提高稳定性。为了提升模型的性能尝试了很多方法，例如在每个卷积层中增加卷积核的数量，例如在第一个和第二个卷积层之间添加一个额外的卷积层；从 16 增加到 32 或 64，以提高模型的表达能力；以及使用不同的激活函数（如 LeakyReLU 或 ELU）；使用全局平均池化（layers.GlobalAveragePooling2D()）替代最大池化。每当看到模型的准确率提升时，内心的成就感无与伦比。同时，我意识到参数调优的复杂性，尤其是学习率、批量大小等超参数对模型性能的影响，有时甚至需要反复试验才能找到最佳配置。

**4. 模型评估与优化**

在使用 classification\_report 和混淆矩阵对模型效果进行评估时，我体会到数据科学不仅仅是训练模型，更重要的是如何合理地评估和解释模型的表现。通过对不同模型的比较，我了解到准确率、精确率、召回率和F1分数等指标各自的意义，以及在不平衡数据集下选择合适指标的重要性。

**5. 未来的展望**

这一项目让我对深度学习在实际应用中的广阔前景充满期待。天气变化的实时监测、无人机监测的智能化、智能家居系统的天气适应能力等，都让我意识到科技与生活的紧密结合。未来，我希望能继续深入研究深度学习，探索更多创新的应用场景，并为应对气候变化贡献自己的力量。

# 八、参考文献

[1]: Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In \*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)\* (pp. 4510-4520).

[2]: Alzubaidi, L., Zhang, J., Humaydi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, I., Santamaria, M. A., Fadhel, M. A., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. \*Journal of Big Data\*, 8(1), Article 53.

[3]: Elpeltagy, M., Sallam, H. Automatic prediction of COVID− 19 from chest images using modified ResNet50. *Multimed Tools Appl* **80**, 26451–26463 (2021).

[4]: Guo, Y., Wang, X., Yuan, J., & Ding, Q. (2020). Multiple feature reweight DenseNet for image classification. In \*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition\* (pp. 1-10).