

队伍编号	MCB2303269
赛道	(A)

基于深度学习的道路坑洼检测和识别

摘 要

本文针对道路的坑洼检测和识别问题，运用了迁移学习、特征提取和分类两种方法，构建了卷积神经网络分类模型和支持向量机分类模型，运用了 MATLAB 软件编写程序训练模型，对测试数据集进行了分类，并将结果放在“test_result.csv”文件中。

针对问题一，要求建立一个分类精度高、速度快的模型，我们建立了两种模型进行对比择优。首先，我们选择了卷积神经网络，它是深度学习中的应用最广泛的模型之一，因为训练数据集较少，我们选择使用预训练的神经网络，我们分别使用基于迁移学习的方法和特征提取并分类的方法来建立模型。然后，我们先运用了深度学习应用中常用到的迁移学习，用训练数据集中的图像数据微调预训练的网络，运用 MATLAB 编程实现，在将训练数据集图像导入时我们发现 9 张图片无法正常导入 MATLAB，这 9 张图片的文件名我们放在了附录 1 中，我们将这些图片截图后按原来的名称储存，覆盖原来未能读取的图片，之后正常建立并训练了模型。经过微调的网络导出并命名为“net301.mat”保存在支撑材料中。其次，我们还运用了预训练的网络对训练数据集进行特征提取，将得到的数据用于训练支持向量机，运用 MATLAB 编程实现，经过训练的支持向量机模型导出并命名为“SVM.mat”。

针对问题二，对模型进行训练评估，模型训练在第一问已经完成，另外我们在训练时从训练数据集中分出了 30% 的数据用于测试验证，对模型的精度和速度进行了评估。对于特征提取并分类的方法我们还进行了分类查准率、查全率和 F1 值三个方面的评估，运用了 MATLAB 软件编程实现，具体评估结果见正文。

针对问题三，用前面训练的模型识别测试数据集中的图像并将图像分为正常和坑洼两类，首先，我们读取了测试数据集中图像的文件名，并将文件名按名称顺序储存，然后，分别使用基于迁移学习的方法构建的模型和特征提取并分类的方法构建的模型按图像名称循环识别分类图像，运用 MATLAB 编程实现，在将测试数据集图像导入时我们发现 29 张图片无法正常导入 MATLAB，这 29 张图片的文件名我们放在了附录 2 中，我们将这些图片截图后按原来的名称储存，覆盖原来未能读取的图片，之后正常识别了测试数据集。将结果分别储存并命名为“迁移学习.xlsx”和“特征提取.xlsx”，结果文件保存在支撑材料中。最后，我们抽样比较两个结果，“迁移学习.xlsx”的准确度更高，达到了 90% 以上，所以将其作为“test_result.csv”文件上传。

本文最后还对模型进行了误差分析，对模型的优点和缺点进行了客观评价。

关键词：深度学习；卷积神经网络；特征提取；迁移学习；支持向量机

目录

一、问题的重述.....	2
1.1 背景知识	2
1.1.1 引言部分	2
1.1.2 切入标题	2
1.1.3 研究意义	2
1.2 相关数据	2
1.3 具体问题	2
二、问题的分析.....	3
2.1 研究现状综述	3
2.2 对问题的总体分析和解题思路	3
2.3 对具体问题的分析和对策	4
2.3.1 问题一的分析 and 对策	4
2.3.2 问题二的分析和对策	6
2.3.3 问题三的分析 and 对策	6
三、模型的假设.....	6
四、名词解释与符号说明.....	6
4.1 名词解释	6
4.2 符号说明	7
五、模型的建立与求解	7
5.1 问题一的分析与求解.....	7
5.1.1 基于迁移学习的方法	7
5.1.2 特征提取并分类	10
5.2 问题二的分析与求解.....	10
5.2.1 基于迁移学习的方法	10
5.2.2 特征提取并分类	11
5.3 问题三的分析与求解.....	12
5.3.1 基于迁移学习的方法	12
5.3.2 特征提取并分类	12
六、误差分析	13
七、模型的评价.....	13
7.1 模型的评价.....	13
7.1.1 优点	13
7.1.2 缺点	13
八、模型的改进.....	14
参考文献.....	16
附录	17

一、问题的重述

1.1 背景知识

1.1.1 引言部分

在科学技术高速发展的今天，机动车已经成为了人们生活中不可或缺的重要一环。这虽然给人们的生活带来了众多的便利，可也带来了一系列的道路交通安全问题。这其中首屈一指的便是由于过多车辆碾压以及恶劣天气等问题所造成的坑洼道路问题，这些问题不仅极大程度地缩短了道路的使用寿命，更是会影响人们的日常使用。

为了解决这些问题，传统的方法是人为设计坑洼的特征，之后通过机器学习的方法来检测坑洼道路。可这不仅在检测坑洼道路上有一定的局限性——算法繁琐、复杂，同时算法的鲁棒性不高，更是会造成人力、物力、财力的浪费。因此，我们急切需要找到一种快捷高效、准确率高的模型来识别道路。

1.1.2 切入标题

近年来，随着计算机视觉和深度学习技术的快速发展，这为坑洼道路识别问题提供了一种很好的思路。深度学习技术有很强大的特征提取和表示能力。面对复杂多变的坑洼道路，我们通过深度学习技术可以直接从图片中自动地提取重要的图像特征，如利用深度学习技术来提取坑洼的轮廓、纹理、形态等特征，并将其转化为更容易分类的表示形式。

1.1.3 研究意义

当今时代，随着深度学习技术的快速发展，计算机视觉中的一些识别、检测、分割等问题都得到了很好的解决。相对于机器学习，深度学习不需要人为的寻找、设计特征，仅通过深度神经网络便可以自动地去提取检测所需要的特征。同时检测的效果也优于传统的图像识别方法。

计算机视觉和深度学习技术在坑洼道路检测与识别的应用虽然越来越广泛，可依旧没有找到一种快捷高效、准确率高的模型。与此同时，这种坑洼道路检测与识别的模型也对地质勘探、航空航天、自然灾害等领域的研究和应用具有重要的意义。

1.2 相关数据

- 1.训练数据集 `data.zip`（见附件 1）
- 2.测试结果提交模板 `test_result.csv`（见附件 2）
- 3.测试数据集 `test_data.zip`（见附件 3）

1.3 具体问题

问题一：根据附件 1 给出的图片进行分析，并从中提取图像的特征，从而建立一个

识别率较高、速度较快、分类准确的模型用于识别图片中的道路是否正常。

问题二：针对问题一所建立的模型进行训练，并从不同的角度对该模型进行相关特征评估。

问题三：利用前两问已经建立的模型，识别题中附件 3 中所包含图片的道路是否正常，并将结果按照题中要求填入附件 2 中。

二、问题的分析

2.1 研究现状综述

计算机视觉虽然作为人工智能的一个分支，但同样也是拥有着极大的功能实现，起先是建立在 Hubel 和 Wiesel 对猫的视觉研究上的。真正意义上开始是在二十世纪六十年代出现了第一位计算机视觉博士即 Larry Roberts，他写的论文《Machine perception of three-dimensional solids》中论述了通过将要识别的物体转化为容易识别的几何体^[1]。方家骐是我国比较早发表有关计算机视觉文章的人，他在《计算机视觉：一个兴起中的研究领域》中表明了计算机视觉是如何发展历程以及它的适用领域和它是如何进行图像处理的^[2]。汪云九翻译的《计算机视觉》点明了计算机视觉也是拥有强大的信息处理能力的^[3]。蔡愉祖在《计算机视觉概述》中着重说明了计算机视觉的应用，以及要如何处理图片建造模型^[4]。李锦成等人在《具有训练、学习和零件识别能力的计算机视觉系统》中则是着重于计算机视觉的模型的训练与识别的能力^[5]。

2.2 对问题的总体分析和解题思路

本题是基于计算机视觉的坑洼道路检测和识别，我们计划采用深度学习的方法进行建模。

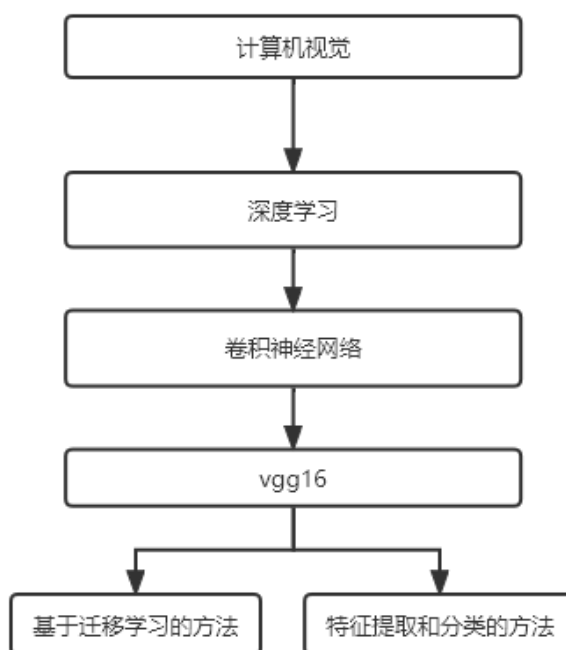


图 1 总体思路

我们之所以选择卷积神经网络是因为卷积神经网络是一项经典的机器学习算法，是深度学习中应用最广泛的网络模型之一，vgg16 是一种分类精度和速度都较好的预训练的卷积神经网络，使用预训练的神经网络 vgg16 进行图像分类我们用了两种方法，一种是使用迁移学习来重新训练卷积神经网络 vgg16 以对测试数据集进行分类，一种是从预训练的卷积神经网络中 vgg16 提取已学习的图像特征，并根据这些特征来训练支持向量机。使用这两种方法分别进行模型建立、模型训练、模型评估和识别分类测试数据集，最后抽样比较两种方法建立的模型对测试数据集的分类准确率，择优上传结果。

2.3 对具体问题的分析和对策

2.3.1 问题一的分析 and 对策

问题一要求根据训练数据集建立一个识别道路图像是正常还是坑洼的模型。首先，我们采用在视觉识别、语音识别和自然语言处理等很多领域都表现出色的深度学习来进行坑洼道路检测和识别。然后，我们使用深度学习中应用最广泛的网络模型之一卷积神经网络（Convolutional Neural Network, CNN）。

卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经网络，是深度学习的代表算法之一。它具有强大的表征学习能力，能够按其阶层结构对输入信息进行平移不变的分。它的基本架构包括特征抽取器和分类器。特征抽取器通常是由若干个卷积层和池化层叠加而组成的，卷积层主要由局部感知、参数共享、多核卷积三部分组成，主要功能是通过卷积运算将卷积核与图像的对应区域进行卷积得到一个值，通过在图像上不断移动卷积核来计算卷积值，从而完成对整幅图像的卷积运算。由于通过卷积运算得到的特征数据过多，导致出现运算量大和过度拟合的情况，池化层的作用便是将这些特征进行聚合处理达到提高效率和降低维度的效果。在特征抽取器后面一般会连接分类器，它通常是由一个多层感知机构成的。在最后一个特征抽取器后面会将所有的特征图展开并排列成一个向量，从而得到特征向量，并将其作为后层分类器的输入。

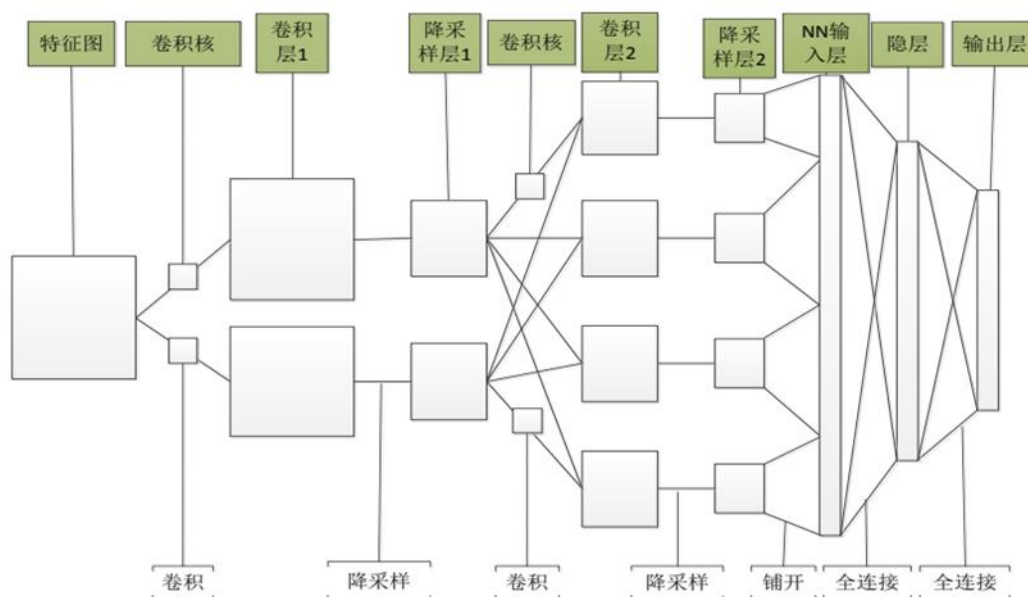


图 2 卷积神经网络结构图

题目要求建立一个识别率高、速度快、分类准确的模型，通过权衡网络的准确度、速度和规模三个重要特征，我们选择了这些特征都较好的 VGG-16。

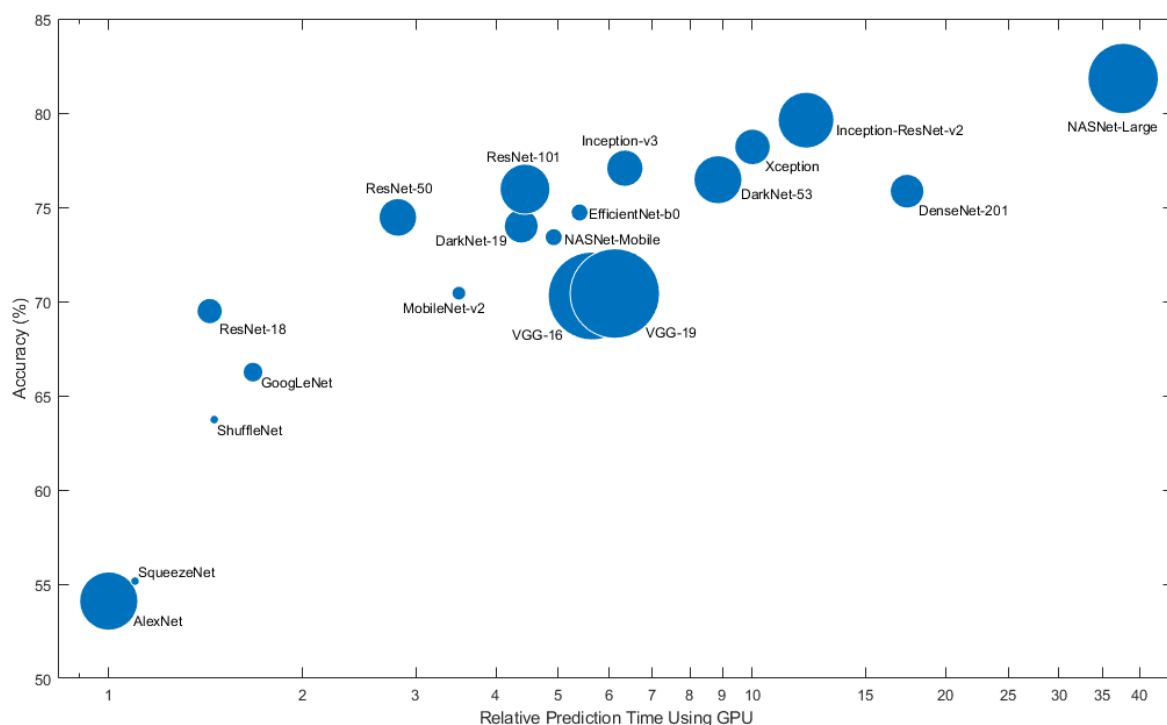


图 3 不同网络 ImageNet 验证准确度和使用网络进行预测所需的时间

问题一要求我们对附 2 中所给训练集图片进行分析和数值化处理，提取相关的数据特征，并建立识别模型。我们采用两种方法来建立数据模型，第一种是我们对预训练的神经网络模型——VGG16 进行迁移学习来微调网络，迁移学习可以使用较少数量的训练图像快速地将已学习的特征迁移到预训练的网络去从而提高模型的精度。第二种方法是将需要处理的图像数据进行存储，通过构建预训练网络来处理导入的图像数据之后得到图像特征，基于提取的图像特征来训练支持向量机，最后用训练好的支持向量机模型进行分类。

支持向量机最初是由 V. Vapnik 等人所提出来的，是通过借助最优化方法来解决机器学习问题的一种工具，也是克服“维数灾难”和“过学习”等问题的一种强力手段。其的主要思想是找到一个超平面，使得这个平面尽可能多的将两类数据点区分开，同时使这两类数据点距离分类面尽可能得远。基本原理如下：

首先，我们先将题中给定的训练集用数据表示出来

$$T = \{[a_1, y_1], [a_2, y_2], [a_3, y_3], \dots, [a_n, y_n]\} \in (Z \times Y^n)$$

其中， $a_i \in Z \subset R^n$ ； Z 为输入空间，输入空间中的每个点 a_i 均由 n 个属性特征组成；

$$y_i \in Y = \{-1, 1\}, i = 1, 2, \dots, n$$

我们通过寻找 R^n 上的一个实值函数 $g(x)$ ，以便将其作为分类函数 $f(x)$ 的参数来对给定数据集进行分类。其中分类函数 $f(x)$ 如下：

$$f(x) = \text{sign}(g(x))$$

2.3.2 问题二的分析和对策

问题二要求我们对问题一中建立的模型进行训练,并从不同的维度进行评估,首先,我们先进行模型的训练,基于迁移学习的方法建立的模型根据训练数据集对预训练的网络 vgg16 进行微调,特征提取和分类的方法建立的模型使用预训练的网络 vgg16 提取训练数据集中的特征来训练支持向量机。然后,进行模型的评估,将训练数据集中拆出 30% 作为测试,对模型的分类精度和速度进行评估。此外,对于特征提取和分类的方法建立的模型我们还进行了分类准确率、分类查准率、分类查全率、分类 F1 值。

2.3.3 问题三的分析 and 对策

问题三中要求我们使用训练好的模型识别测试集中的图像,首先,第一种方法我们将题中所给测试集中的图片加载到我们建立的迁移学习模型中,经过微调的 vgg16 网络进行分类,然后第二种方法我们直接使用 vgg16 提取测试集中图像的特征,并经过训练好的 SVM 模型分类,将文件名和分类的结果分别储存到一个表中。最后,抽样估算一下各自的分类准确度,选择分类准确度高的结果。

三、模型的假设

- 1.训练数据集中包含的特征与测试数据集中包含的特征基本对应
- 2.附件中的图像清晰、可读

四、名词解释与符号说明

4.1 名词解释

计算机视觉技术:通过使用摄影机、电脑、传感器等设备来代替人的眼睛或模拟人的判别准则,对目标进行识别、跟踪和检测,并做进一步做图形处理。从而可以理解和识别图像,用图像分析做出相应结论的实用技术,其中图像处理和图像分析是计算机视觉技术的核心^[6]。

数字图像:以二维数字组形式所表示的图像,其数字基本单元为像元。数字图像的恰当应用通常需要有数字图像与看到的现象之间关系的知识^[7]。

深度学习:通过学习样本数据的内在规律和表示层次,从而让机器能够像人一样具有分析学习能力,如能够识别文字、图像和声音等数据。

迁移深度学习:是针对在训练样本量有限的情况下的一种有效学习方法,可以缓解由于训练样本量不足所导致的卷积神经网络出现过拟合现象。迁移学习从本质上讲就是知识和经验的迁移重用,其核心是利用了源域和目的域之间存在相似性原理^[8]。

卷积神经网络:通过利用局部连接、强大的特征提取能力和权值共享的特性充分提取图像特征和显著降低模型的训练参数^[9]。

VGG16:一种深度卷积神经网络模型,该模型的设计思想是通过堆叠多个较小的卷积层和池化层来构建深层网络,以增强模型的表达能力。通俗来讲,该模型是由 16 个卷积层和 3 个全连接层组成。其中卷积层主要用于提取输入图像的特征,而全连接层则用

于将提取到的特征映射到类别概率上^[10]。

特征提取：通过对原始数据进行某种变换，得到最能反映分类本质的特征，形成模式的特征空间。以后的模式分类或模型匹配就在特征空间的基础上进行^[11]。

支持向量机：即一般化的线性分类器，它的学习策略是间隔最大化，最终可转化为一个凸二次规划问题的求解，故也被称为最大边缘区分类器^[12]。

预训练的神经网络模型：该模型是迁移学习的一种应用，通过利用几乎无限的文本，学习输入句子的每一个成员的上下文相关的表示。同时，它还可以将从开放领域学习到的知识迁移到下游任务中去，以改善低资源任务。

4.2 符号说明

序号	符号	符号说明
1	$fnames$	测试图片的文件名
2	$label$	分类标识：1 表示正常，0 表示坑洼
3	T	训练数据集
4	Z	输入空间
5	a_i	输入空间中的点
6	n	属性特征序号
7	Y	输出空间

五、模型的建立与求解

5.1 问题一的分析与求解

5.1.1 基于迁移学习的方法

首先，我们先对题中所给的 301 张图片进行数据预处理。根据题中所给图片的文件名称，我们可以清楚的知道其中包含了 35 张坑洼道路的图片 and 266 张正常道路的图片。



图 4 训练书籍集部分读入错误

在 Matlab 读取图片时，我们又发现了这其中一共有 9 张图片程序无法读入，为了解决这一问题，我们采用了打开图片后截屏保存的方式使之能够被程序所读入。为了使结果更加合理，我们将数据（正常和坑洼道路按比例）划分为训练数据集和验证数据集两部分——70%的图像用于训练，30%的图像用于验证。在进行 Matlab 提取图片特征时，vgg16 网络要求输入图像的大小为 $224 \times 224 \times 3$ ，其中 3 代表我们对图片的颜色处理采取的是 RGB 三色模型。但训练数据集中的图像大小不一，我们采用数据增强的方式自动调整输入图像的大小，数据增强包括沿垂直轴随机翻转训练图像，以及在水平和垂直方向上随机平移训练图像最多 30 个像素并将训练图像缩放最多 10%，并且数据增强还能够帮助记忆图像的具体细节以及预防过拟合。

我们进行迁移学习是对预训练的 vgg16 网络模型进行微调，生成新的网络。

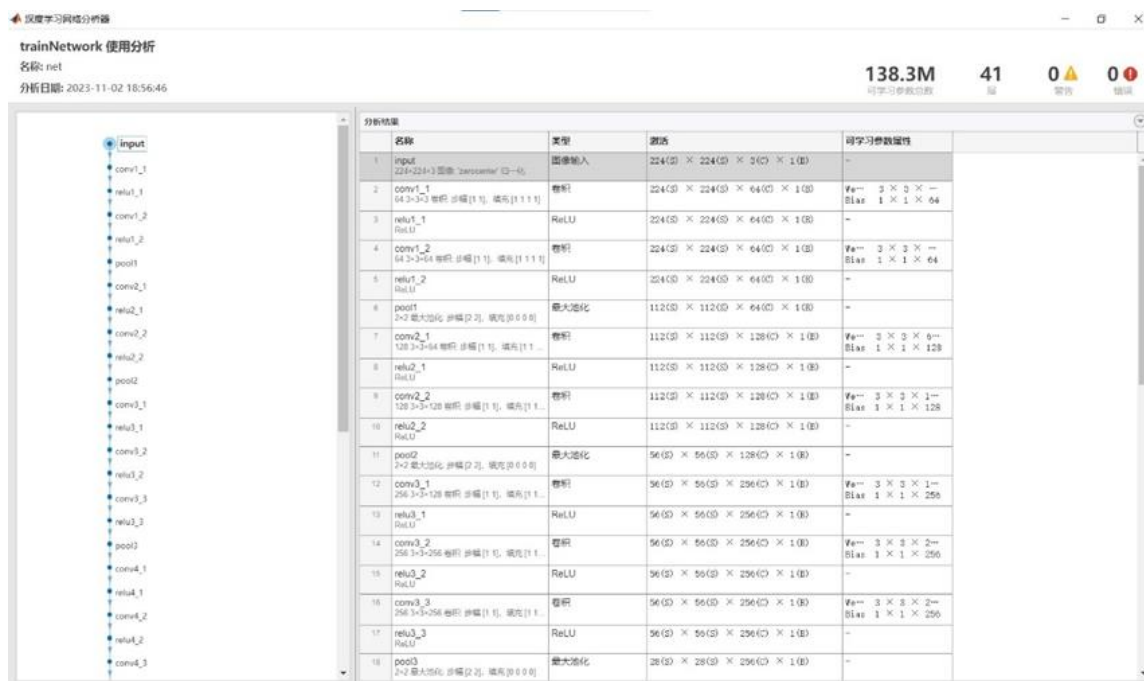


图 5 vgg16 网络架构以及有关网络层的详细信息

vgg16 神经网络模型的设计思想是通过堆叠多个较小的卷积层和池化层来构建深层网络，以增强模型的表达能力。准确来讲，该模型是由 16 个卷积层和 3 个全连接层所组成的。卷积层是由多个卷积核所组成的，每个卷积核都是一个小矩阵，通过与输入的数据进行卷积运算，从而提取出不同位置的特征。全连接层则用于将提取到的特征映射到类别概率上。其中“loss3-classifier”和“output”这两个层包含有关如何将网络

提取的特征合并为类概率、损失值和预测标签的信息，进行微调时需要将这两个层替换成新层。

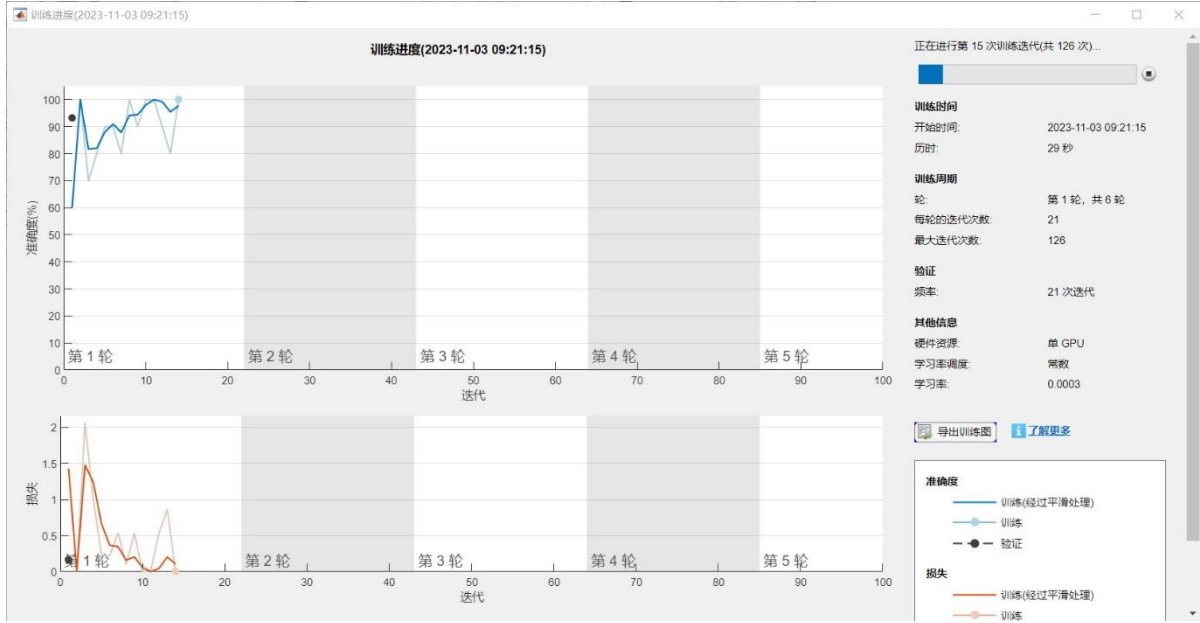


图 6 网络训练过程中

迁移学习是一种利用已经训练好的模型，在该模型中通过加入一些新的元素进行训练从而微调该模型的方法。它主要分为两个步骤：第一步是特征提取，我们使用 freezeWeights 将 vgg16 神经网络模型的初始层冻结住可以显著加快网络的训练速度，因为训练数据集较小，还可以防止训练数据集过拟合。冻结的做法是将初始层的学习率设置为 0，被冻结的层不会更新参数，也不需要计算梯度。之后使用 createLgraphUsingConnections 函数按原始顺序连接所有层^[13]。

只训练我们新加入的 301 张图片——新的全连接层。网络的卷积层会提取最后一个可学习层和最终分类层用来对输入图像进行分类的图像特征，从而提取出输入数据的特征。第二步是微调，我们将之前冻结的前几层解冻后，对整个网络模型进行微调，以便更好的进行道路图片检测。

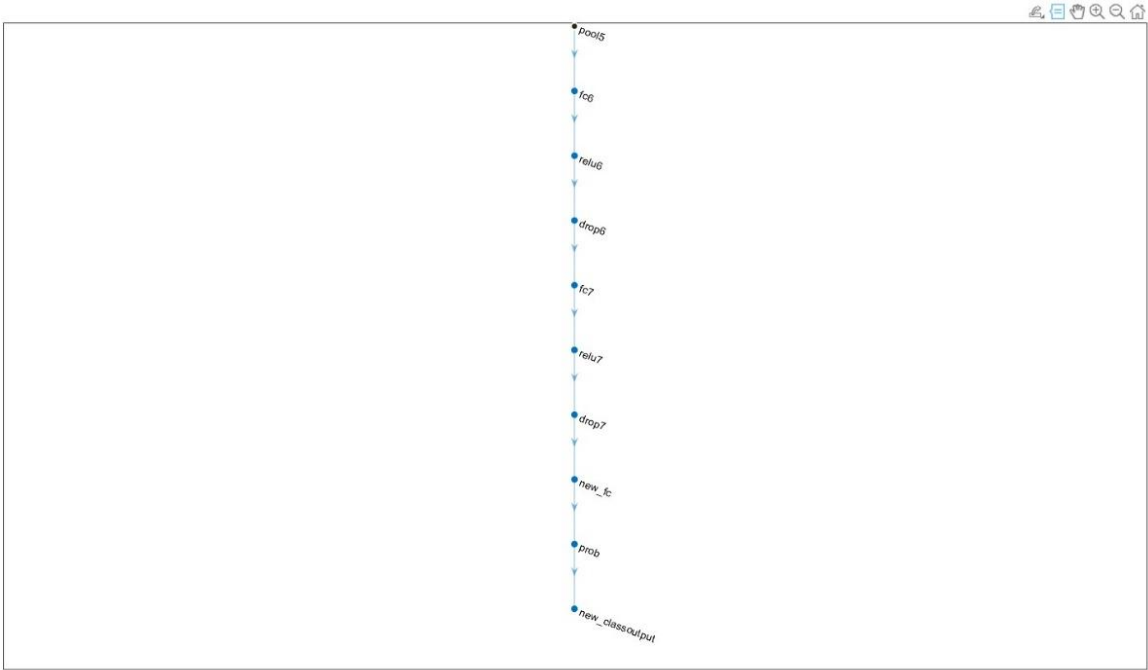


图 7 网络的最后几层放大图

5.1.2 特征提取并分类

通过预训练过的卷积神经网络来提取训练图像的图像特征，基于提取的特征来训练支持向量机。

首先，进行数据的导入，将需要使用到的图像作为图像数据储存在 Matlab 工作区，并且用 ImageDatastore 标注图像和文件夹名称一样，存储为 ImageDatastore 对象，之后将数据拆分其中的 70%为训练时使用的数据，其余的 30%为测试所用的数据。其次则是要加载预训练时用到的 vgg16 网络，其中 vgg16 网络的图像输入层设置为 $224 \times 224 \times 3$ （3 是颜色通道数，224 为像素）。最后一步是提取图像特征，将图像数据存储到网络之前需要先使用图像增强调整图像的大小为 $224 \times 224 \times 3$ 。然后 activations 的输入参数为调整过大小的图像数据，网络构造更深层包含更高级别的特征，这些特征使用较浅层的较低级别特征构建，所以要想要获得训练图像集和测试图像集的特征，我们还要通过使用 activations 对 vgg-16 网络末尾的全局池化层（简称“pool5”）进行处理。通过处理后的全局池化层汇集了所有输入特征，我们得到了将近 4096 个图像特征^[14]。

5.2 问题二的分析与求解

5.2.1 基于迁移学习的方法

该问要求我们对问题一建立的模型进行训练，并对训练的结果从不同的角度进行评价。

我们对题给的图片进行分类标注，将正常的道路标注为 1，反之则为 0。之后通过将图片随机翻转、在水平和垂直方向上随机平移训练图像，从而对数据集进行扩充，并将所有图片的格式设置为 $224 \times 224 \times 3$ 。

之后，我们就开始训练我们刚才输入的图片了。首先，我们将 InitialLearnRate 的参数设置为一个较小的值以便可以降低尚未冻结的迁移层中的学习速度。同时，我们增大最后一个可学习层的学习率因子，从而加快新的最终层中的学习速度。这种学习率设置组合方式会加快新层中的学习速度，减慢中间层中的学习速度，以便停止较浅的冻结层中的学习。

之后，我们便对该模型进行评估。如下图所示，我们可以清楚的看出该模型的验证准确率为 100%，并且所需要的时间也仅仅只有不到三分钟。

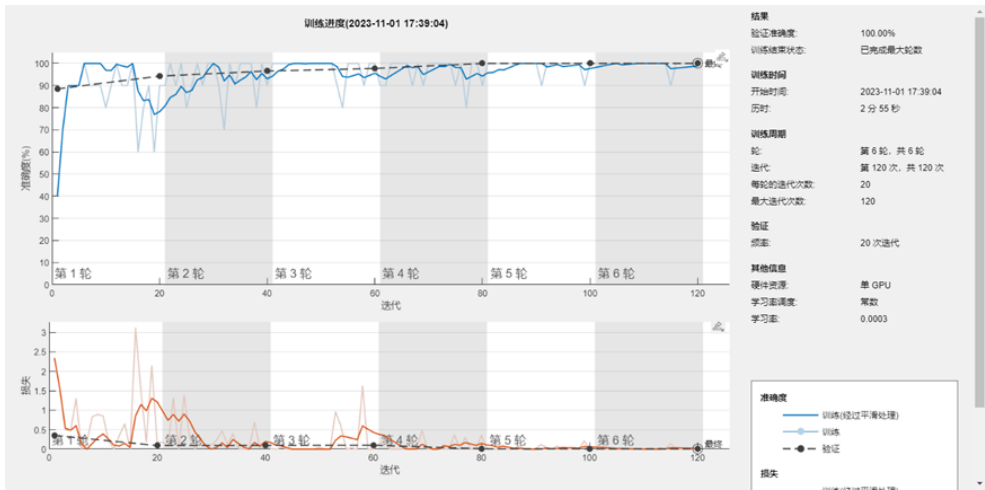


图 8 网络训练完成

5.2.2 特征提取并分类

首先，训练支持向量机，从训练图像中得到的图像特征作为预测变量，然后运用函数 `fitcecoc` 拟合多类支持向量机。然后，遍历数据得到图像特征，测试图像集分类则是通过使用训练过的 SVM 模型和测试图像集中获得的图像特征得到的，然后计算出的测试集的分类准确（accuracy）率、分类查准（Precision）率、分类查全（Recall）率、分类 F1（F1 score）率。



图 9 评价指标输出

准确率 *ACC*：正例和负例中预测正确数量占总数量的比例

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

查准率 *precision*：以预测结果为判断依据，预测为正例的样本中预测正确的比例。

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

查全率 *recall*：以实际样本为判断依据，实际为正例的样本中，被预测正确的正例占总实际正例样本的比例。

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

F1 值：中和了精确率和召回率的指标

$$F1 = \frac{2PR}{P + R}$$

同时训练过的多类支持向量机是具有很高精准度，我们得到的训练集和测试集的分类准确率、分类查准率、分类查全率、分类 F1 率均为 100%表明测试集和训练集中坑洼道路图像和正常道路图像是完全分开的，能够准确的识别出基本上所有坑洼或正常的路面，根据分类 F1 率表示分类查准率和分类查全率是存在平衡的^[15]。

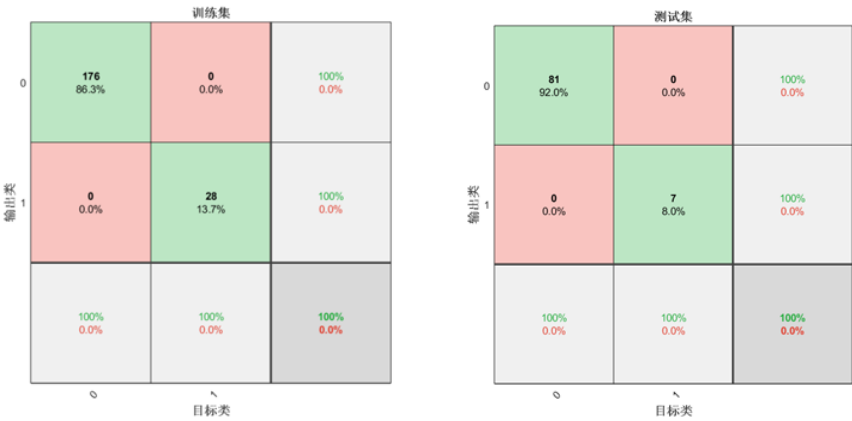


图 10 训练集和测试集的混淆矩阵

混淆矩阵基本解释：TP 表示真正例，预测为正例而且实际上也是正例；FP 表示假正例，预测为正例而且实际上却是负例；FN 表示假负例，预测为负例而且实际上也是负例；TN 表示真负例，预测为负例而且实际上也是负例。

5.3 问题三的分析与求解

5.3.1 基于迁移学习的方法

我们使用 `dir` 函数列出测试集文件夹中的内容，然后使用 `imread` 函数将图片读入时发现错误，我们将出现错误的图片用画图打开后截图保存覆盖了原先出现错误的图片，使得可以正常读入图片，出现错误的图片共有 29 张，具体的图片名称在支撑材料中有展示。

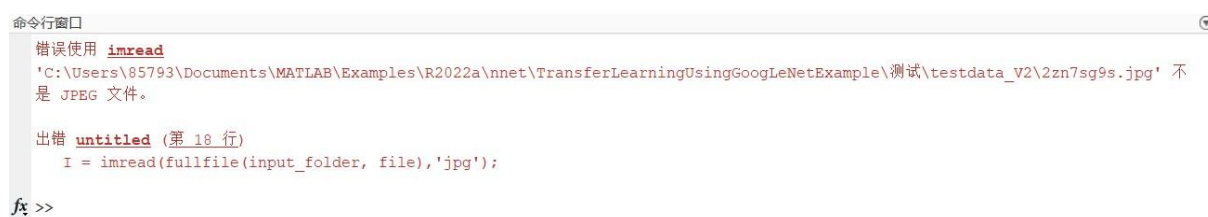


图 11 测试数据集部分读取错误

每一张读入的图片都经过数据预处理后，用微调后的网络分类，经过循环操作，将每张图片的分类结果按顺序保存在一个变量中，最后将文件名称和分类结果保存在“test_result.csv”中。

经过抽样比较此模型的分类准确率大约为 83%，进行 4945 张图像的分类用时大约 150 秒。



图 12 基于迁移学习的方法耗时

5.3.2 特征提取并分类

直接使用“基于迁移学习的方法”中处理过后的图片，循环提取每张图片的特征，并使用训练好的支持向量机进行分类，将分类结果和图像名储存在一个表中输出。

经过抽样比较此模型的分类准确率大约为 33%，进行 4945 张图像的分类用时大约 332 秒。



图 13 特征提取并分类耗时

六、误差分析

误差出现的原因有很多种，比如由于图像本身存在的问题例如由于图像的清晰度、格式等引发的问题。除了图像引发的外，还有些误差是因为采集的样本不够导致无法识别出一些特殊的情况。除此之外，在模型的建立中的一些小参数等也是会引发误差的，同时也包括程序无法正常识别不同大小的图像等等。

1. 没有考虑到因图像清晰度造成的识别错误，比如像有些图片比较模糊导致的将正常道路识别成坑洼或坑洼识别成正常。

2. 训练数据集较少对模型分类准确度有影响，计算分类准确率的过程中需要用到训练时图像特征。

3. 图像的大小不一也是会引发一些问题的产生，有可能会使有些较大图像无法整个的识别出来，虽然经过预处理但还是会对结果造成影响。

4. 在计算数据时也会发生一些误差，比如少输入或多输入数据都是会引发误差。

然而这些误差也是可以避免的，比如认真的处理每一个数据，训练数据图像集时多导入些图像样本，在编程代码时减少拼写错误，不会的代码上网寻找参考，减少程序的逻辑错误等。

七、模型的评价

7.1 模型的评价

7.1.1 优点

在研究本题时，我们首先通过权威报刊、杂志等渠道查阅相关资料，以期获得灵感。在解决问题时，为了使得我们最终选取的模型效果更好，我们采用了两种方法进行建模，通过对测试集识别效果、时间等方面的对比，选出一种更适合本题的模型。同时，为了使我们建立的模型更有说服力，我们在文中放了许多易于理解的图片。

1. 运用 MATLAB 将所用网络可视化展示，更加便于理解网络的结构；
2. 运用 MATLAB 软件做出结果的相关图表，更加直观、清晰、简捷；
3. 运用了两种不同的方法进行分析，使结果更加可信
4. 比较不同神经网络的精确度和速度，选用了精确度和速度都较好的 vgg16

7.1.2 缺点

在刚开始解决问题时，我们组的研究方向与出题人的方向有所不同，我们组虽然考虑的是运用神经网络中卷积神经网络进行深度学习，不过在模型训练集的选择上，我们考虑的是在题中所给训练集的基础上，扩大范围——从互联网上寻找大量的图片，以提高我们模型的精度。这时精度虽然相较于之前有所增强，不过训练模型所耗用的时间扩大了好几倍。这时，我们才发现我们之前的思路有所偏差。另外，由于卷积神经网络无法避免的误差问题，这会导致我们建立的模型在识别道路上的精度有所下降。

1. 特征提取并分类模型训练的时间相比于基于迁移学习的模型的训练时间较长
2. 没有考虑图像清晰度对识别的影响

3. 特征提取并分类模型分类的精度相比于基于迁移学习的模型分类精度较低

八、模型的改进

对于模型的后续改进，可以从头训练一个新的卷积神经网络，并且增加一个能够在图像中框出坑洼位置的功能。具体的步骤如下：

因为训练数据集包含的图片较少，所以新建立的卷积神经网络的鲁棒性不佳，首先加载数据，手动添加每个图像上坑洼的位置，用一个矩形标识框将坑洼的位置框起来。

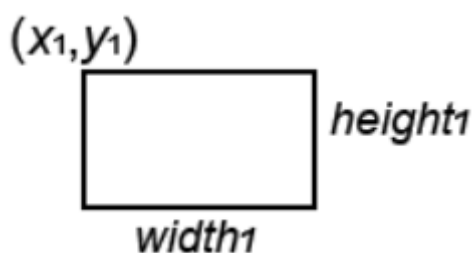


图 14 矩形标识框的数据

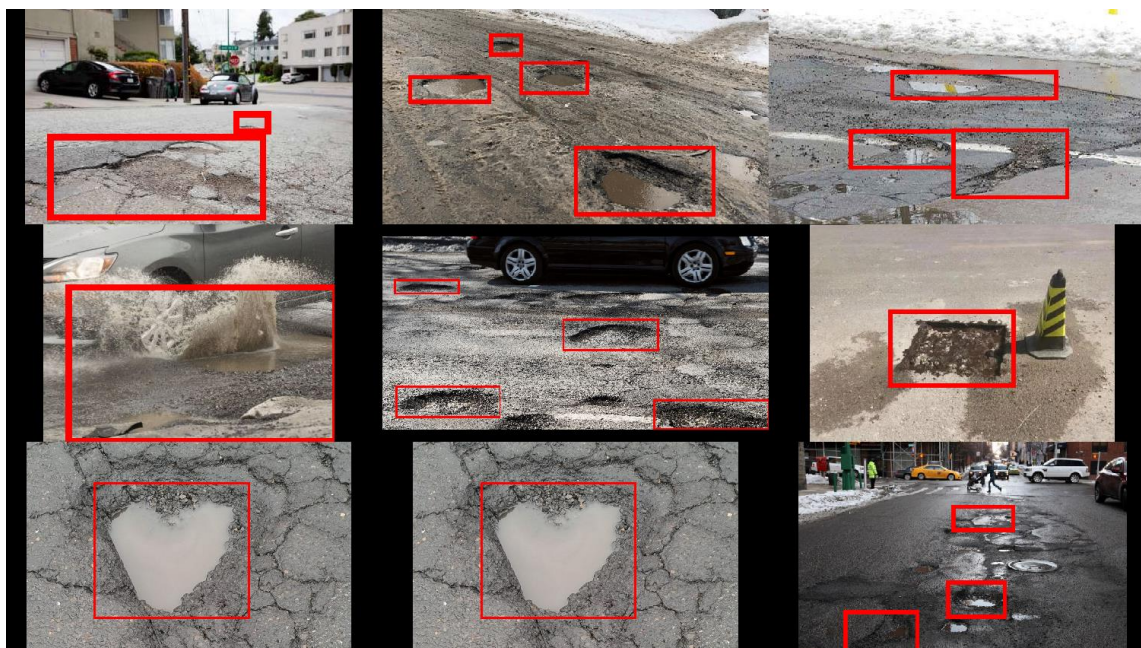


图 15 坑洼的位置添加标识框

然后，构建卷积神经网络的输入层、中间层和输出层，为了将数据利用完全，在训练之前将 40%的数据用于测试使用。训练完成后就得到了卷积神经网络模型，最后对训练好的卷积神经网络进行测试和评估。（测试时标识框上方的数字为得分）



图 16 测试结果图

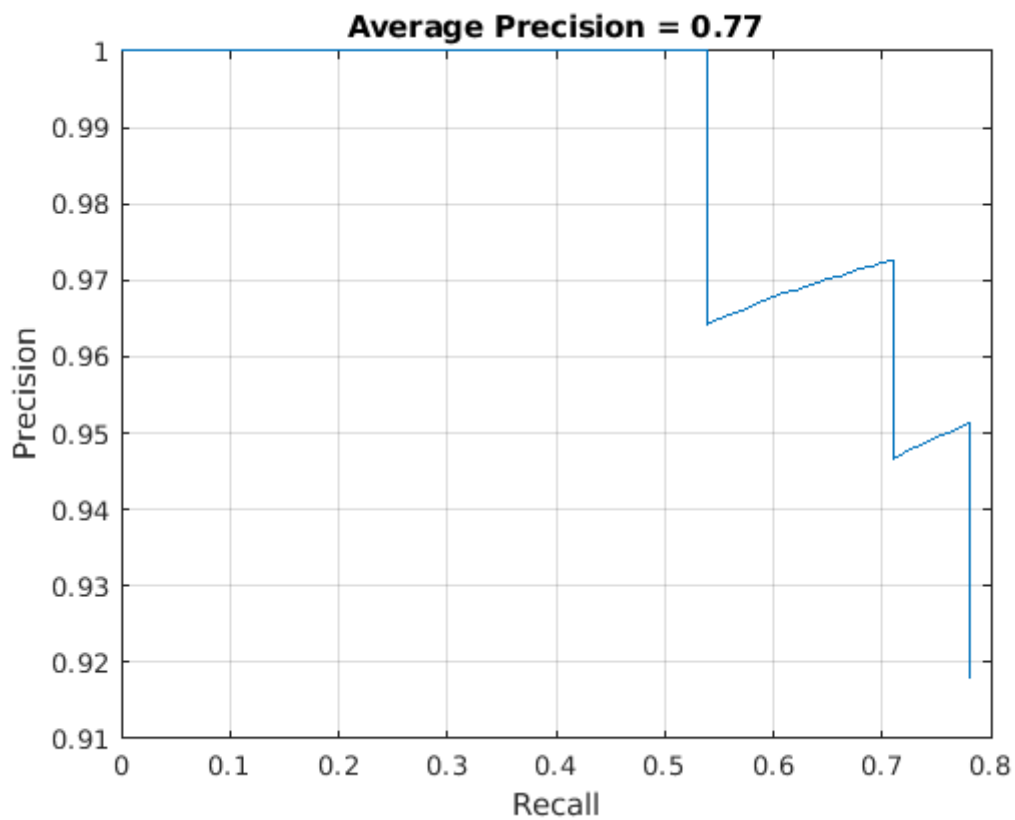


图 17 召回率-精确率训练评估曲线

具体的网络构建和训练步骤放在支撑材料中命名为“CNN.m”，因为训练数据较少，所以精确率不太高，可以考虑增加 CNN 的层数来改善，但同时会增加训练和检测的时间和空间。

参考文献

- [1] L. G. Robert. Machine perception of three-dimensional solids. In: Optical and Electro-Optical Information Processing, Tippet J, et al. eds. 1965: 159-197.
- [2] 方家骥. 计算机视觉: 一个兴起中的研究领域[J]. 计算机应用与软件, 1984(03): 6-13.
- [3] M. M. Waldrop, 汪云九. 计算机视觉[J]. 世界科学, 1985(07): 39-41.
- [4] 蔡愉祖. 计算机视觉概述[J]. 系统工程与电子技术, 1986(01): 60-68.
- [5] 李锦成, 丁鼎一, 邹瑞雨. 具有训练、学习和零件识别能力的计算机视觉系统[J]. 信息与控制, 1986(04): 23-27+64.
- [6] 王素方主编. 进出口肉及肉制品分析检验技术. 郑州: 河南科学技术出版社. 2012. 第 64 页
- [7] 王晓峰, 高俊波, 孔繁荣主编. 英汉人工智能辞典. 上海: 上海交通大学出版社. 2019. 第 108 页
- [8] 王紫腾. 基于深度迁移学习与多特征网络融合的高分辨率遥感图像分类[D]. 南京邮电大学, 2023. DOI:10.27251/d.cnki.gnjdc.2022.000148.
- [9] 刘东旭. 基于卷积神经网络的高光谱遥感图像分类方法研究[D]. 中国科学院大学(中国科学院长春光学精密机械与物理研究所), 2023. DOI:10.27522/d.cnki.gkcg.2023.000005.
- [10] 陈敏. 基于 VGG16 和迁移学习的太阳射电频谱分类[D]. 云南大学, 2022. DOI: 10.27456/d.cnki.gyndu.2022.002158.
- [11] 孙正等编著. 数字图像处理与识别. 北京: 机械工业出版社. 2016
- [12] 李明禄主编. 英汉云计算·物联网·大数据辞典. 上海: 上海交通大学出版社. 2018. 第 371-372 页
- [13] 训练深度学习网络以对新图像进行分类[OL/EB].
<https://ww2.mathworks.cn/help/releases/R2022a/deeplearning/ug/train-deep-learning-network-to-classify-new-images.html#TransferLearningUsingGoogLeNetExample-4>, 2023. 11. 01
- [14] 使用预训练网络提取图像特征
<https://ww2.mathworks.cn/help/releases/R2022a/deeplearning/ug/extract-image-features-using-pretrained-network.html#ExtractImageFeaturesUsingDeepLearningExample-4>, 2023. 11. 01
- [15] Demir, F., Sengur, A. & Bajaj, V. Convolutional neural networks based efficient approach for classification of lung diseases. Health Inf Sci Syst 8, 4 (2020).
<https://doi.org/10.1007/s13755-019-0091-3>

附录

附录 1: 训练数据集中无法正常读取的 9 张图像文件名

normal62.jpg
normal65.jpg
normal86.jpg
normal109.jpg
normal115.jpg
normal209.jpg
normal238.jpg
normal298.jpg
normal303.jpg

附录 2: 测试数据集中无法正常读取的 29 张图像文件名

1s5wwgyj.jpg
2opx9qe9.jpg
2zn7sg9s.jpg
3inaj8fa.jpg
5v72lm8x.jpg
6clf5qkv.jpg
95xn0wdo.jpg
9seh7a2k.jpg
b4dhh72i.jpg
gcnt738f.jpg
gok9u1ui.jpg
gr4rtntp.jpg
i7r79svl.jpg
jdbnxrvc.jpg
keh6yya4.jpg
lg35k89d.jpg
mqk8hand.jpg
nm8588vt.jpg
no6pc45v.jpg
of8zc3cs.jpg
ozyorkia.jpg
p1bsn9ra.jpg
q07xyecg.jpg
qnprg86m.jpg
qrguntxk.jpg
ves8f2lu.jpg
vq60mgrc.jpg
x8o38g7e.jpg
xv6dx5c5.jpg

附录 3: MATLABR2022a 源程序

clc
clear

```

imds = imageDatastore('data', ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');
[imdsTrain,imdsValidation] = splitEachLabel(imds,0.7);
load('vgg16.mat')
analyzeNetwork(net)
net.Layers(1)
inputSize = net.Layers(1).InputSize;
lgraph = layerGraph(net);
[learnableLayer,classLayer] = findLayersToReplace(lgraph);
[learnableLayer,classLayer]
numClasses = numel(categories(imdsTrain.Labels));
if isa(learnableLayer,'nnet.cnn.layer.FullyConnectedLayer')
    newLearnableLayer = fullyConnectedLayer(numClasses, ...
        'Name','new_fc', ...
        'WeightLearnRateFactor',10, ...
        'BiasLearnRateFactor',10);
elseif isa(learnableLayer,'nnet.cnn.layer.Convolution2DLayer')
    newLearnableLayer = convolution2dLayer(1,numClasses, ...
        'Name','new_conv', ...
        'WeightLearnRateFactor',10, ...
        'BiasLearnRateFactor',10);
end
lgraph = replaceLayer(lgraph,learnableLayer.Name,newLearnableLayer);
newClassLayer = classificationLayer('Name','new_classoutput');
lgraph = replaceLayer(lgraph,classLayer.Name,newClassLayer);
figure('Units','normalized','Position',[0.3 0.3 0.4 0.4]);
plot(lgraph)
ylim([0,10])
layers = lgraph.Layers;
connections = lgraph.Connections;
layers(1:10) = freezeWeights(layers(1:10));
lgraph = createLgraphUsingConnections(layers,connections);
pixelRange = [-30 30];
scaleRange = [0.9 1.1];
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange, ...
    'RandXScale',scaleRange, ...
    'RandYScale',scaleRange);
augimdsTrain = augmentedImageDatastore(inputSize(1:2),imdsTrain, ...
    'DataAugmentation',imageAugmenter);
augimdsValidation = augmentedImageDatastore(inputSize(1:2),imdsValidation);
miniBatchSize = 10;
valFrequency = floor(numel(augimdsTrain.Files)/miniBatchSize);

```

```

options = trainingOptions('sgdm', ...
    'MiniBatchSize',miniBatchSize, ...
    'MaxEpochs',6, ...
    'InitialLearnRate',3e-4, ...
    'Shuffle','every-epoch', ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',valFrequency, ...
    'Verbose',false, ...
    'Plots','training-progress');
net = trainNetwork(augimdsTrain,lgraph,options);
[YPred,probs] = classify(net,augimdsValidation);
accuracy = mean(YPred == imdsValidation.Labels)

```

```

clc;clear;close all
image_path=[pwd '\data\'] ;
namelist=dir([image_path,'*.jpg']);
num_data=size(namelist,1);
for i=1:size(namelist,1)
    name=namelist(i).name;
    if strcmp(name(1:6),'normal')
        label(i,:)=0;
    else
        label(i,:)=1;
    end
end
load('vgg16.mat')
inputSize = net.Layers(1).InputSize;
layer = 'fc6';
feature_all=[];
ind_error=0;
for i=1:num_data
    try
        I =imread([image_path namelist(i).name]);
        I =imresize(I,[224 224]);
        feature= activations(net,I,layer,'OutputAs','rows');
        feature_all=cat(1,feature_all,feature);
    catch
        ind_error=ind_error+1;
        error_file(ind_error).name=namelist(i).name;
        label(i)=[];
        continue
    end
end
feature_all=double(feature_all);
delete('01_feature_vgg16.mat')

```

```

save('01_feature_vgg16.mat','feature_all')
delete('02_LABEL.mat')
save('02_LABEL.mat','label')

```

```

clc
clear
load('net301.mat');
all_pic =
dir( 'C:\Users\85793\Documents\MATLAB\Examples\R2022a\nnet\TransferLearningUsingGo
ogLeNetExample\测试\testdata_V2\*.jpg');
pic_name = cat(1,all_pic.name);
input_folder = 'testdata_V2';
files = dir(fullfile(input_folder, '*.jpg'));
C=[];
B=[];
tic
for i = 1:length(files)
    file = files(i).name;
    I = imread(fullfile(input_folder, file),'jpg');
    I = imresize(I, [224 224]);
    [YPred,probs] = classify(net,I);
    C=[C,YPred;];
    B=[B,probs];
end
toc
TB=B';
tc=C';

```

```

clc;clear;close all
tic
load('02_LABEL.mat')
load('01_feature_vgg16.mat')
len=length(label);
idx = randperm(len);
num_train = round(0.7*len);
num_val = round(0*len);
x_train=feature_all(idx(1:num_train),:);
x_val=feature_all(idx(num_train+1:num_train+num_val),:);
x_test=feature_all(idx(num_train+num_val+1:end),:);
Label = categorical(label);
Label_Train = Label(idx(1:num_train));
Label_Val = Label(idx(num_train+1:num_train+num_val));
Label_Test = Label(idx(num_train+num_val+1:end));
SVM_3=SVM_3_Classifier(x_train, Label_Train);

```

```

delete('SVM.mat')
save('SVM.mat','SVM_3')
Yp_Train = SVM_3.predictFcn(x_train);
figure
plotconfusion(Label_Train,Yp_Train)
title('训练集')
[A,~]=confusionmat(Label_Train,Yp_Train);
A=A';
Acc=(A(1,1)+A(2,2))/sum(A,"all");
Precise=A(2,2)/(A(2,1)+A(2,2));
Recall=A(2,2)/(A(1,2)+A(2,2));
F1_score=2*Precise*Recall/(Precise+Recall);
disp(['训练集分类准确率: ',num2str(Acc*100),'%'])
disp(['训练集分类查准率: ',num2str(Precise*100),'%'])
disp(['训练集分类查全率: ',num2str(Recall*100),'%'])
disp(['训练集分类 F1 值: ',num2str(F1_score*100),'%'])
disp('-----')

```

```

Yp_Test = SVM_3.predictFcn(x_test);
figure
plotconfusion(Label_Test,Yp_Test)
title('测试集')
[A,~]=confusionmat(Label_Test,Yp_Test);
A=A';
Acc=(A(1,1)+A(2,2))/sum(A,"all");
Precise=A(2,2)/(A(2,1)+A(2,2));
Recall=A(2,2)/(A(1,2)+A(2,2));
F1_score=2*Precise*Recall/(Precise+Recall);
disp(['测试集分类准确率: ',num2str(Acc*100),'%'])
disp(['测试集分类查准率: ',num2str(Precise*100),'%'])
disp(['测试集分类查全率: ',num2str(Recall*100),'%'])
disp(['测试集分类 F1 值: ',num2str(F1_score*100),'%'])
toc

```

```

clc;clear;close all
image_path=[pwd '\testdata_V2\'];
namelist=dir([image_path,'*.jpg']);
num_data=size(namelist,1);
tic
load('vgg16.mat')
inputSize = net.Layers(1).InputSize;
layer = 'fc6';
feature_all=[];
ind_error=0;
for i=1:num_data

```

```

try
    I = imread([image_path namelist(i).name]);
    I = imresize(I,[224 224]);
    feature= activations(net,I,layer,'OutputAs','rows');
    feature_all=cat(1,feature_all,feature);
    file(i).name=namelist(i).name;
catch
    ind_error=ind_error+1;
    error_file(ind_error).name=namelist(i).name;
    continue
end
end
feature_all=double(feature_all);
load('SVM.mat')
Yp = SVM_3.predictFcn(feature_all);
toc
columns = {'fnames', 'label'};
data = table({file.name}', Yp, 'VariableNames', columns);
writetable(data, 'test_result.csv')

```