

四川大學

本科生毕业论文（设计）



题 目 电视剧信息生成学习模型

学 院 计算机学院

专 业 计算机科学与技术

学生姓名 陈怡凡

学 号 2015141462014 年级 2015

指导教师 张建州

教务处制表

二〇一九年五月十日

电视剧信息生成学习模型

专业名 计算机科学与技术

学生 陈怡凡

指导老师 张建州

[摘要] 更具有实际意义的多标签分类一直是人工智能领域的一个热门研究问题。论文较为全面的探讨该课题，并研究了在文本和图像数据集上训练样本和训练模型的选择，设计实验使用电视剧情节简介和海报对电视剧体裁进行多标签分类。研究问题的难点主要来源于数据集的缺乏、标签的抽象性和算法模型的优化。因此，论文从数学角度出发分析选取“代表性”数据集的重要性，并且使用迁移学习弥补深度学习模型对资源的依赖，不仅极大优化了算法，而且简化了代码设计。此外，相关机器学习算法，包括深度学习在内都进行了系统的描述：依据分类问题解决思想的不同，将机器学习算法分为两类，判别学习算法和生成学习算法，通过理论分析和实际研究测试进行了对比；使用了 Word2Vec 和 VGG-Net 两个神经网络对深度学习展开研究。最终，深度学习模型在研究测试中取得了优异的精确率和召回率评估指标结果，参考类似算法在电影数据集中研究的相关文献，也可以看出所使用的机器学习模型的有效性。

[关键词] 电视剧体裁多标签分类；机器学习；生成学习；判别学习；神经网络；迁移学习

Generating and learning model of TV show information

Major Computer Science and Technology

Student Chen Yifan

Adviser Zhang Jianzhou

[Abstract] One of the trending topics in artificial intelligence is multi-label classification, which is really pragmatic in the real world. This dissertation comprehensively discusses the research topic and researches on the selection of training data and training models in textual and visual datasets, and utilizes TV show synopses and posters to classify their genres. The difficulties of the research problem are due to deficiency of datasets, obscure meaning of labels and optimization of algorithm modeling. Hence, the paper mathematically analyzes the importance of representative data and introduces transfer learning by using pre-trained deep learning models to tackle with the dependence of resources, which not only optimizes the algorithms extremely but also simplifies the codes. In addition, related machine learning algorithms including deep learning ones are illustrated systematically: divided on discrete ideas of solving classification problems, machine learning algorithms are classified into discriminative learning and generative learning algorithms and they are made comparisons with each other through theoretical analysis and practical testing; two neural network model—Word2Vec and VGG-Net are used in deep learning. Finally, the deep learning models have the greatest evaluation results of precision rate and recall rate in tests and it is noteworthy that designed machine learning models have worked effectively as well given some related references based on researches of movie datasets.

[Keywords] TV shows' genres multi-label classification; machine learning; generative learning; discriminative learning; neural network; transfer learning

目录

第一章 引言	5
1.1 研究背景	5
1.2 国内外研究现状	5
1.3 论文的主要工作和结构安排	6
1.3.1 完成目标	6
1.3.2 论文结构安排	6
第二章 主要研究问题及算法理论	7
2.1 多标签分类问题描述	7
2.2 数据集的创建原则	8
2.3 机器学习算法综述	9
2.3.1 判别学习算法	10
2.3.2 生成学习算法	11
2.3.3 深度学习分类算法	12
2.4 手工获取电视剧海报和简介数据集的方法描述	12
2.5 多类别支持向量机模型描述	13
2.6 多项式朴素贝叶斯模型描述	14
2.7 神经网络模型描述	15
2.7.1 运用于海报数据集的 VGG-Net 模型	15
2.7.2 运用于简介数据集的 Word2Vec 模型	16
2.8 算法分类效果的评估指标	17
2.9 项目部署介绍	19
2.10 本章小结	19
第三章 文本多标签分类模型算法设计	20
3.1 生成数据集的前期准备	20
3.1.1 获取数据的开源 python 包描述	20
3.1.2 IMDB 与 TMDB 数据的对比抉择	21
3.1.3 构建数据集的方法定义及所需工具	22
3.2 构建文本数据集的详细设计	23
3.2.1 二进制向量表示电视剧体裁标签得到输出矩阵 Y	23
3.2.2 预处理电视剧简介输入矩阵 X	23
3.3 判别模型和生成模型训练过程	24
3.3.1 使用 TF-IDF 模型对词袋中的词设置权重	24

3.3.2 支持向量机判别模型代码设计.....	25
3.3.3 多项式朴素贝叶斯生成模型代码设计	25
3.3.4 Word2Vec 迁移学习模型的训练过程设计.....	25
第四章 图片多标签分类模型算法设计	27
4.1 构建海报图片数据集的详细设计	27
4.2 VGG-Net 迁移学习模型训练过程	27
4.2.1 特征抽取方式.....	27
4.2.2 VGG-Net 的迁移学习代码实现	27
4.3 所有算法的统一评估设计	28
第五章 算法测试以及结果分析	30
5.1 数据集介绍	30
5.1.1 数据集的统计数据	30
5.1.2 利用成对比较分析数据集	30
5.2 多标签分类模型的评估指标统计	32
5.2.1 多类别支持向量机使用参数及测试结果	33
5.2.2 多项式朴素贝叶斯使用参数及测试结果	36
5.2.3 Word2Vec 模型测试示例及结果	38
5.2.4 VGG-Net 模型测试示例及结果	40
5.2.5 评估总结	41
5.3 结果分析	43
5.3.1 传统机器学习模型对比分析.....	43
5.3.2 神经网络模型结果分析	43
5.4 创新点总结	43
总结	45
参考文献	46
致谢	48
附录 1	49
附录 2	67

第一章 引言

1.1 研究背景

分类是人类社会中应用最为广泛的研究和解决问题的手段之一，也是在人工智能领域也是最先开始研究的内容之一，例如对文本、图像以及视频进行分类。直至今日，相比其他问题，分类这一领域已经取得了较多的研究成果。但传统的机器学习的分类问题通常局限于二分类问题，例如判断图片上是狗还是猫，或者文本中的情感是正面的还是消极的；在现实中，这样的分类方式过于理想化，也不够具有代表性。现实中的文本、图片等数据绝大多数包含两种以上的语义信息，例如风景图中会有人、建筑、天空以及树木，生物基因序列对中信息的多样性。因此，本论文将从该角度出发，对文本以及图片的多标签分类这一问题展开研究。

机器学习包括深度学习在内，在这一世纪已成为计算机科学的主导研究算法。这是由于网络的发展所带来的指数级增长的数据已无法再用传统算法进行高效地处理，因此利用人工智能解决实际问题已成为一个普遍的意识，特别是在处理图像视频数据方面，更能彰显机器学习算法的优势。为了更加有效地分类多标签的文本和图片，本文也将使用、学习、比较和改进一系列的机器学习算法，来尝试得到最优的结果。具体来说，考虑到深度学习和非深度学习算法的差异与特性，将会用经典的机器学习算法对文本进行多标签分类，也就是多语义分类；而对图片进行分类时，考虑到数据集以及数据量剧烈增长，因此使用神经网络来训练、预测、研究等更为合适，本文也将讨论如何应用迁移学习来在优化的模型上获得解决本论文所研究问题的分类模型。

通过大量的前期学习和阅读文献，本论文的数据集将从两个大型电影电视剧网站IMDB (<http://www.imdb.com/>) 与 TMDB (<https://www.themoviedb.org/>) 抓取，原始数据集信息由于一方面包括了电视剧以及电影的所有相关信息，如发行日期，导演，演员等，与研究内容无关，另一方面信息并没有经过归一化，因此清理和转换所需的数据集也是本论文的工作之一。

1.2 国内外研究现状

对多分类问题的研究，通常根据数据集性质的不同，采用传统机器学习或深度学习两种研究思路。对于文本多类别分类，大多采用的是改进的经典算法，如多类别支持向量机，决策树和随机森林等。图片的多标签分类更为复杂，经常需要结合实际问题来应用不同算法，本论文主要参考 ImageNet 图像识别大赛中性能较好的模型来进行模型构建。

多标签分类的难点相比于传统的单一标签分类问题有如下两点：一是测试样本的标签数量不确定，有些样本标签数在一个至上百个之间均有可能；二是标签之间的相互依赖

性，即标签之间可能存在包含关系。因此解决以上两个问题使提高多分类算法性能的关键。

本论文将聚焦用电视剧的简介和海报对电视剧进行体裁预测这一问题，对多标签分类展开研究。该问题跟通常的多标签分类问题不同之处在于：一，用作预测的样本的不成熟性，也就是指示的特征并不明确。虽然电视剧简介通常包括故事内容，但事实上要单从文字中获得准确的体裁信息即便使对于人工分类来说，也是较为困难的；海报作为输入样本同理。二，对于大型视频的分类所用的数据集大多是利用的电影数据集，鲜有对于电视剧体裁分类的研究。这也可能是由于电视剧体裁分类实际上要考虑的因素众多，通常还是借助人工、主观判断，没有统一标准，因此给算法评估造成了一定的负面影响。但即使是使用电影信息数据集研究体裁预测，各种指标的最高正确率也远远低于标准的文本和图片多标签分类的正确率，如电影评论情感分类，图片中多类目标标记。

1.3 论文的主要工作和结构安排

1.3.1 完成目标

本论文将完成的主要目标包括三部分：

- 1) 对现在主流生成模型及判别模型进行研究、总结和对比，了解机器学习，包括深度学习在多标签分类中的最新技术及常见算法。
- 2) 手工创建一个电视剧的简介和海报的文字与图片数据集，包括每个数据的标签，也就是每个电视剧所属的体裁。
- 3) 对 2) 中的数据集进行多标签分类，优化、对比并评估使用的各类算法。选用的算法为支持向量机，朴素贝叶斯和神经网络算法。

1.3.2 论文结构安排

本论文的结构安排将从理论研究出发，探讨研究问题的提出、解决以及最后研究结果的展示。首先将在第二章介绍解决多标签分类问题的三大思路以及主流的机器学习算法（包括深度学习），还有本论文所采用的算法的实现思路；第三、四章会详细的介绍程序详细设计和代码实现；第五章将给出结果评估以及分析。最后是总结毕业论文设计全过程。

第二章 主要研究问题及算法理论

2.1 多标签分类问题描述

首先需要区分的一组概念是多标签分类（Multi-label Classification）与多类别分类（Multi-class Classification）。多类别分类可以看作是二分类的拓展，强调输入样本被分类时类别的多样性；而多标签分类是指在该输入样本本身的多维性，即标记集中每个标签在同一样本中进行分类判断时是两两独立的。因此多标签分类问题是一个更加广泛且普遍的问题。

传统的单一标签分类是指，在标签域 L , $|L| > 1$ 中需要找到一个标签 l , 使 l 满足对所给输入 χ_i , $i \in L$ 来说，特征距离最小；当标签集满足 $|L| = 2$ 时，问题便转换为单一分类下的二分类特殊情形，使用场景多为判断文本情感或网络数据的过滤。当 $|L| = 2$, 并且从寻找单一标签 l 变为寻找标签集合 $Y \subseteq L$, $|Y| \geq 1$ 时，问题则称为多标签分类问题，此时分类器 $\theta(\chi)$ 返回的是 $\theta(\chi) = (\theta_1(\chi), \theta_2(\chi), \theta_3(\chi), \theta_4(\chi), \theta_5(\chi) \dots)$, 即一个向量。1.2 小节将会介绍国内外研究该问题时三大主流思想，分别是：集成学习，问题转换以及算法改进^[2]。

事实上，国外学者对多标签学习的研究开始较早。在 2000 年的一篇论文^[1]就提出了使用改进的提升算法（boosting algorithm）对多标签文本进行分类，也就是 AdaBoost.MH 算法和 AdaBoost.MR 算法，实现方式是通过在每一轮训练时给弱分类器分配一组权重，根据弱分类器的预测值与实际值来优化权重，最终得到强分类器。这类弱学习算法是用集成学习（Ensemble Learning）解决多标签分类问题的代表之一。另一种常见的使用集成学习方法是 bagging 算法。该算法与提升算法虽然都是通过综合弱分类器的预测值来提升分类决策，但 bagging 算法采用的思路是首先抽取多个采样集，然后在每个采样机上进行多个弱分类器的训练，接着组合这些无依赖关系的弱分类器的分类策略，得到最终的强分类器。在此基础上，2001 年 Breiman 在^[3]中提出随机森林（Random Forest）算法这一集成学习方法，关注了分类决策树的应用，将 bagging 思想和随机特征选取相结合，成为一种广泛使用的非线性机器学习算法。

问题转换的方法实现方式大致可分为三类：二分类关联（Binary Relevance），分类器链（Classifier Chains）以及标签集转换（Label Powerset）^[4]。二分类关联的思想容易理解，即把每一个标签单独看作一个单一分类问题，最具代表性的是高斯朴素贝叶斯算法。而分类器链方式是依次训练一连串的标签分类器，每次在进行训练的分类器只针对一个标签训练，但不仅要训练输入数据，还有前面的分类器的训练结果，但非同类的输入数据之间彼此独立，也相当于转化为多个单一标签问题。这与二分类关联法有些相似，但区别在于按序训练的不同分类器之间保留了前后标签的关联性。至于标签集转换，是用规则去再次定义每个输入样本的标签，使问题被转换为一个或多个单一标签分

类问题。Label-Powerset^[5]是该类型算法中应用最广的一种算法，主要算法实现是将每个输入变量的所有标签统一为一个新标记集，把得到的每个标记集当作一个新的唯一标签来训练，已达到将问题转换为单一标签问题。改进的 LP 算法被称为 Random K-labelsets^[5]，采用集成思想构建多个 LP 分类器，在不同分类器中训练标记集中的随机标签子集，考虑标签之间的相关性，弥补 LP 算法的缺陷。

第三种思路是算法改进。这种方法顾名思义，是直接基于多标签进行研究，这也是与问题转换方法最大的不同之处。Multi-Label k-Nearest Neighbor (ML-kNN)^[6]，Rank Supporting Vector Machine (Rank-SVM)^[7]和各种基于神经网络的算法是其中的代表。ML-KNN 算法的核心思想是基于最大后验概率原则 (MAP) 决定输入变量的标签，最大后验概率是指由 KNN 算法的前验、后验概率所得到的样本的标记集概率^[6]。而 Rank-SVM 算法是将问题转化成对 (Pairwise) 的分类问题，之后再利用 SVM 算法解决。

总的来说，由于深度学习在目前各类数据集中的优良表现，已成为了多标签分类问题的首要考虑算法。集成的学习方法在实际应用中对于复杂的数据来说，效果要比一些利用单一分类思想的问题转换方法要好，这是由于问题转换方法要考虑类别标记直接的联系，因此对于标签数量庞大的数据集来说计算复杂度较低；但集成方法在这类复杂问题中存在极其耗时的问题，因此实际使用时不如问题转换和算法改进这两个方法使用广泛。

2.2 数据集的创建原则

对于机器学习算法的研究，获取一个具备代表性的数据集是一个必要条件。这一项步骤事实上是一个看似简单其实较为关键的问题，这是因为一个良好的数据集往往能对最后的算法设计以及结果评估产生很大影响。

首先从数学原则来考虑机器学习。有监督的机器学习算法解决问题过程可以抽象的总结为：给出一个输入变量 X 以及输出变量 y ，如果函数方程 $g(X)=y$ 是未知的，那么我们就要用机器学习算法来“学习”一个函数方程 F ，使 F 与 g 尽可能相似。

举例来说，如果现在问题为预测某人的电影喜好，那么 X 可以为某人的性别，年龄和出生地等， y 应最可能为电影类别。之后，准备收集如上所说的数据集，如果收集的数据集是随机走进一间大学教室向教室中的人征集的信息，那么可想而知，用这样的数据集训练得到的模型一般来说效果较差。这是因为这个模型不具有代表性：如果用它预测一名儿童的电影喜好，那么应该会得到完全与实际不符的输出。

图 1 说明了从数学的角度展示了为什么数据集的代表性尤为重要。假设该图垂直方向上的坐标轴为上述的输出变量 y ，水平方向的两条坐标轴为输入变量 X_1 与 X_2 ，则机器学习算法则是要找到一个函数方程 F 使其尽可能接近出如图绘制的函数。如果收集到的数据集 X_1 与 X_2 均落在 (50, 60) 的区间中，那么算法所学习到的函数 F 只能表示中间角落

红色部分，进而，如果要预测位于蓝色部分的 y ，则是几乎不可能的。因此为了学习到一个好的函数方程，就需要收集一个具有代表性的数据集也就是包含尽可能多的不同的 X_1 与 X_2 的取值。

在本文中，所需的训练数据集是关于电视剧的文本和图像数据，基于上述考虑，将使用电视剧的简介和海报图片作为 X 来判断体裁标签 y 。这是因为简介中核心的故事情节往往隐含着电视剧的体裁，而海报中的视觉元素恰恰是用来吸引对某类体裁感兴趣的观众。生成有代表性数据集最简单的方式是从互联网中手工抓取信息构建符合问题研究的数据集，而不是通过实地调查收集。本文将使用电视剧的海报作为图像数据集，电视剧的情节简介作为文本数据集，使用这两个数据集，就可以建立预测电视剧体裁的有监督学习模型。

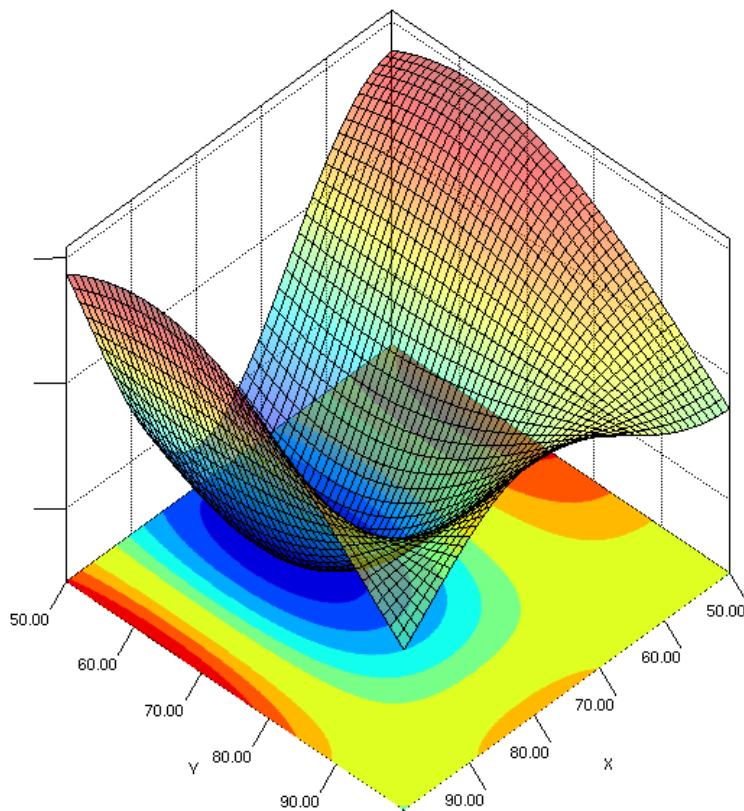


图 1 由 JZY3D 绘制的填充等值线示例图

(<http://www.jzy3d.org/js/slider/images/FilledContoursDemo.png>)

2.3 机器学习算法综述

本节内容主要为总结并介绍近年研究中对多标签分类问题使用的有监督的机器学习算法，并将分为两类介绍。

简单来说，有监督的机器学习的目标是为了学习到一个模型^[8]，并且该模型的形式一

般是一个决策函数 $y = f(x)$ 或一个条件概率 $P(y|x)$ 。算法得到生成模型则被称为生成学习算法，得到判别模型则被称为判别学习算法，表 1 举例展示了两个概念的不同。以下两小节将作具体解释。

表 1 第一个表格是输入的四个样本以及标签，第二个表格是得到的生成模型，联合概率

$$\sum P(x,y) = 1; \text{ 第三个表格是判别模型, } \sum P(x|y) = 1.$$

输入样本及其标签	样本 1	样本 2	样本 3	样本 4
χ	0	0	1	0
γ	1	0	1	0

生成模型	$\gamma=0$	$\gamma=1$
$\chi=0$	1/2	1/4
$\chi=1$	0	1/4

判别模型	$\gamma=0$	$\gamma=1$
$\chi=0$	2/3	1/3
$\chi=1$	0	1

2.3.1 判别学习算法

判别学习方法是指算法直接从输入数据中学习从输入 χ 到输出标签的映射 $y = f(x)$ 和条件概率 $P(y|x)$ 作为预测时使用的模型，使得判别学习的准确率往往较高；其次，可以有效简化学习问题，因为需要将输入 x 进行抽象、定义特征等步骤。常见的判别模型有线性回归、逻辑回归、随机森林、K 近邻法、支持向量机和传统的神经网络等。

逻辑回归（Logistic regression）相当于加入了一个逻辑函数的线性回归模型，因此弥补线性回归鲁棒性较差的缺陷，将预测值的范围减小（通常在 0 到 1 之间）的一种广义线性模型，假设 χ 与 γ 之间的关系是：

$$P(\gamma = T|\chi) = \frac{\exp(\chi^T \beta)}{1 + \exp(\chi^T \beta)}, \text{ 称之为 Sigmoid 函数。}$$

逻辑回归的决策方法是：对于 $\gamma = h_{dis}(\chi)$ 来说，如果 $l_{dis}(\chi) = \sum_{i=0}^d \beta_i \chi_i + \theta$ 大于 0，则预测 γ 为 T ，其中 θ 为阈值。因此逻辑回归是一个利用后验分配概率进行预测的判别模型。

K 近邻法（k Nearest Neighbors, kNN）通过距离度量来判别 χ 属于哪一个标签 γ 。给定一个训练数据集，经过训练将训练集数据分类，对于每个输入的测试数据，计算其与已分类的训练数据的空间距离，表决出其中 K 个最近的分类类别，用这 K 个训练样本进行

“多数表决”来决定测试样本的分类结果。

支持向量机（Supporting Vector Machine）的基本模型是二分类线性可分的，利用超平面将表示为点的输入样本以间隔最大化的方式分隔开，两个超平面之间的区域被称为间隔（margin），此区域正中间的超平面是最大间隔超平面。具体地说，输入样本首先被投射到一个空间中，用一个个的点表示，在空间中定义超平面的概念，用法向量 W 和 b 表示为 $X^T W + b = 0$ ，将该空间分为两部分，一侧为正向类一侧为负向类。为了求得间隔最大化的唯一解，需要选择两个可以分离空间中的点的平行超平面，使二个分类之间的距离尽可能地大。由数学距离计算公式可推广定义 margin 的值，即

$$\text{margin} = \rho = \frac{2}{\|W\|}.$$

由于我们的目标是使 ρ 最大化，因此可定义：

$$\max_{W,b} \rho \quad \max_{W,b} \rho^2 \quad \min_{W,b} \frac{1}{2} \|W\|^2.$$

加上约束条件后，间隔最大化的数学表达为：

$$\min_{W,b} h(W) = \min_{W,b} \frac{1}{2} \|W\|^2, \text{s.t. } \gamma_i (X_i^T W + b) \geq 1, i = 1, 2, \dots, n$$

2.3.2 生成学习算法

生成学习方法与判别学习方法最大的不同在于对联合概率 $P(\chi, \gamma)$ 的学习，得到的生成学习模型的一般形式为 $P(\gamma | \chi) = \frac{P(\chi, \gamma)}{P(\chi)}$ ，表示生成关系。因此，生成学习算法能够处理

输入 χ 标签缺失问题，并且当输入样本量增加时，模型能更快的收敛为符合真实的模型。朴素贝叶斯分类器、隐马尔科夫模型、高斯混合模型和受限波尔兹曼机等都是经典的生成模型。

朴素贝叶斯分类器（NB）是基于假设预测之间相互独立的贝叶斯理论的一个经典的生成学习模型。类似地，NB 算法假设在某一类的某一标签与该类其他标签独立，也就是说，NB 算法强调的是在特征向量中每一维度的不相关性，联合概率分布可得到简化：

$$P(\chi, \gamma) = P(\gamma) \times P(\chi_1, \chi_2, \dots, \chi_n | \gamma) = P(\gamma) \times \prod_{i=1}^n P(\chi_i | \gamma)$$

决策方法 $\gamma = h_{gen}(\chi)$ 可以通过对数似然比来解释：

$$l_{gen}(X) = \log \frac{P(Y=T | X)}{P(Y=F | X)} = \frac{P(Y=T) \times \prod_{i=1}^n P(X_i | Y=T)}{P(Y=F) \times \prod_{i=1}^n P(X_i | Y=F)}.$$

如果对数似然比大于零，即 $P(Y=T | X) > P(Y=F | X)$ ，将 X 分类为 T 。

基于 NB 算法衍生出的三个应用于不同分类场景的模型分别是高斯模型，多项式模型和伯努利模型。高斯模型较适用于连续分布的输入数据；而如果输入数据大多是离散的

话，则使用多项式模型；特别的，如果输入样本特征是二元或多元且稀疏的离散值，则会使用伯努利模型。

在自然语言处理中，隐马尔科夫模型是使用最为普遍的概率统计模型之一。该模型基于马尔可夫假设，其核心思想是假设在一个随机过程中的某一时刻的状态 q_i 只与前一时刻的状态 q_{i-1} 有关。模型中包含的五个组成部分：

初始概率分布 $\pi = \pi_1, \pi_2, \dots, \pi_n, s.t. \sum_{i=1}^n \pi_i = 1$ ，表示马尔可夫链开始于某一状态 i 的概率；

N 个状态用 $Q(q_i), i=1, 2, \dots, N$ 表示，相当于标签；

观测值 $O(o_T)$ 为可能的输入样本；

转移概率矩阵 $A(a_{ij}), s.t. \sum_j a_{ij} = 1 \forall i$ ，表示从状态 i 到 j 转移的概率；

输出概率 $B(b_i(o_T))$ ，每个概率是观测值 o_T 从状态 i 中产生的概率^[9]。

通过三个步骤构建隐马尔科夫模型，使用的算法分别为 forward 算法、Viterbi 算法和 forward-backward 算法，用来进行评估（Evaluation）即观测序列的概率、解码（Decoding）即预测、学习（Learning）即训练模型参数。

高斯混合模型是另一种典型的生成模型，将标签 y 看作是一个隐变量与 x 一起求联合概率分布。它通过求解多个子分布模型，设置阈值来判断输入的某一样本是否属于某一标签，因此特别适合于多分类问题。

2.3.3 深度学习分类算法

利用神经网络的深度学习算法事实上也可用判别学习和生成学习进行归类，例如卷积神经网络（Convolutional Neural Network）以及循环神经网络（Recurrent Neural Network）就属于判别模型，而一些最近热门的神经网络算法，如对抗神经网络（Generative Adversarial Neural Network）、变分自动编码器（Variational AutoEncoder）等利用了生成学习的思想。但由于在神经网络中利用生成模型解决的往往不是分类问题，而是与之相反的生成问题^[10]：通过学习时算法自动生成图像等，因此，本论文使用的神经网络还是基于判别模型的思想实现的。对于多标签分类问题，常常利用的神经网络主要是多标签卷积神经网络以及多标签循环生成网络，包括 LSTM 模型和 GRU 模型。

2.4 手工获取电视剧海报和简介数据集的方法描述

本文的数据集来源为 IMDB 和 TMDB。IMDB 是互联网中电影信息的主要信息源，但在该网站的数据库中也将电视剧作为其中的一个分类。每一个电影或电视剧的海报、评价、评分以及情节概述等其他信息基本上都可以从中找到。TMDB 也称为 The Movie DataBase，是 IMDB 的开源版本，允许注册用户使用 API 进行信息获取和调用。因此，使用该网站的信息，首先要注册一个免费帐号，以获取访问数据所需的 API。

以下是本论文数据收集的步骤概述：

在 TMDB 官网 (<https://www.themoviedb.org/>) 注册账号并申请 API 用于获取电视剧的信息。

设计函数获取 TMDB 的电视剧体裁、海报以及情节简介。

重复上一步骤获取 IMDB 中的相同信息。

在同一电视剧中对比从两个不同网站中获取信息的差异。根据对研究目标的考虑综合分析该两个数据源的优劣，从中选择一个数据库收集并建立最终的数据集。

2.5 多类别支持向量机模型描述

传统的支持向量机（SVM）是一类解决分类类别只有两种的问题的模型。为了解决多标签分类问题，许多学者不断在传统 SVM 模型上提出改进，几种常见的多类别支持向量机（Multi-class SVM）分类方法有一对余法（One-vs-Rest）、一对一法（One-vs-One）、有向非循环图法（Directed-Acyclic-Graph-SVM）和二叉树法（Binary Tree）等。这些改进的算法基本思想可大致分为两类：一是问题转换，即把多类别分类问题转化成二分类问题；另一种是首先构建多个子二分类器，通过将其组合解决问题。在^[11]这一论文中的实验研究表明，一对一法相比一对余法和有向非循环图法在现实应用下有着最好的实验结果。下面将介绍前两个算法的实现思想和特点。

一对余法的算法思想是对于有着 N 类的多标签分类问题需要创建 N 个 SVM 分类器，每个分类器 $N_i, i=1, 2, \dots, n$ 本质上仍是一个二分类器，将属于标签 i 的样本与不属于标签 i 的样本视为两类^[12]。这种方法的优势是只需要训练 N 个分类器就可以建立模型，但训练的效率会随训练集数上升而变差，因为每个子分类器都要训练全部的输入样本。

一对一法类似于一对余法，但是在每两个标签之间构建一个子分类器，因此一共需要的分类器个数为：

$$\frac{N(N-1)}{2}$$

这一方法在每个子分类器上只需训练两类样本，在训练集分类类别数较小时，训练速度比一对余算法快很多；但由于构建的子分类器数量更多，在有着大量分类类别的训练集上其训练效率和准确率都会降低。另外，一对一法容易受到不可区分特征的影响，如

图 2 中阴影部分。

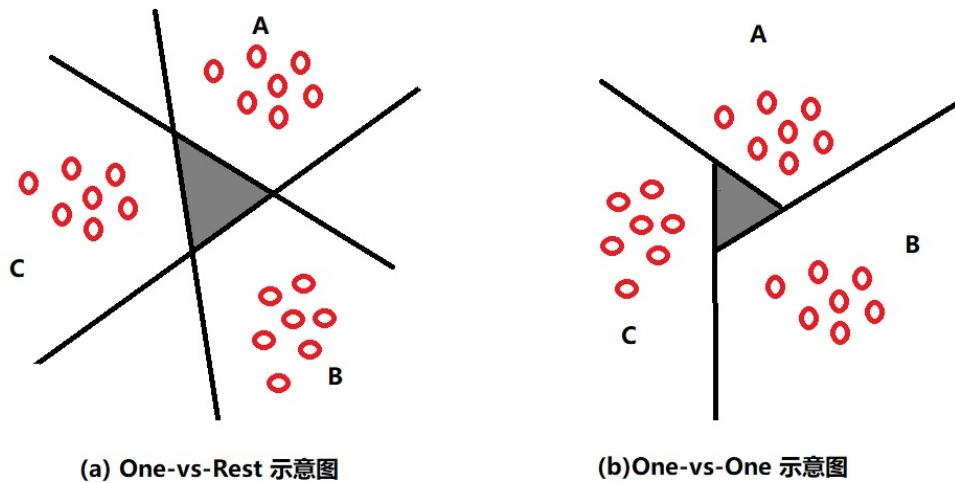


图 2 One-vs-One 与 One-vs-Rest 方法的分类示意图

2.6 多项式朴素贝叶斯模型描述

由于本论文的训练特征是离散的，在使用朴素贝叶斯模型时，将采用其多项式模型。该模型最重要的一个处理步骤是平滑处理，这是为了防止极大似然估计出的后验概率值为零，即某些特征并未在训练样本中出现。

平滑处理后的先验概率为：

$$P(\gamma_i) = \frac{N_{\gamma_i} + \lambda}{N + K\lambda}, i \in [1, K]$$

其中 N_{γ_i} 是标签 γ_i 的训练样本个数， N 是总训练样本个数， K 是总标签个数， λ 是平滑值。

平滑处理后条件概率为：

$$P(\chi_n | \gamma_i) = \frac{N_{\chi_n, \gamma_i} + \lambda}{N_{\gamma_i} + \alpha \lambda_{\gamma_i}}$$

其中 α 是特征的维数， N_{χ_n, γ_i} 是标签 γ_i 的样本中第 n 维特征值 χ_n 出现的次数， N_{γ_i} 是标签为 γ_i 的训练样本个数， λ 是平滑值。

平滑值需设一个大于等于零的值；特别地，平滑值为 0 时是极大似然估计，不做平滑处理；为 1 时称为拉普拉斯平滑；在 0 与 1 之间被称为 Lidstone 平滑。

2.7 神经网络模型描述

卷积神经网络在提取视觉特征方面显示了深远的研究价值，也不断取得了突破性的研究，成为了近年来应用最广、最热门的算法之一。在 2012 年后的 ImageNet 图像分类大赛^[14]中 AlexNet^[13]的卓越成果，使得深度学习这一领域开始蓬勃发展，之后出现的基于卷积神经网络模型如 Google 的 GoogLeNet^[15]、牛津大学与 DeepMind 的 VGG-Net^[16]和微软的 ResNet^[17]不断刷新图像识别领域的各项纪录，使得卷积神经网络的各种应用迅速发展。

2.7.1 运用于海报数据集的 VGG-Net 模型

由于传统的机器学习算法在提取图像特征上相比深度学习相关算法有着明显的劣势，本论文将使用其中的 VGG 卷积神经网络和迁移学习结合实现对电视剧海报的多标签分类。VGG-Net 探讨了如何提高卷积神经网络的深度来提高网络训练性能，它通过不断堆叠小尺寸的卷积层 (3×3) 和池化层 (2×2) 使得网络达到 16 至 19 层深。该网络不但在当年的 ImageNet 大赛中创下了记录，极大地降低了错误率，而且由于网络的泛化能力强，也很容易的拓展到其他图片的数据集。

VGG-Net 给出了五种不同深度的网络，这些网络结构图如图 3 所示。

卷积神经网络结构（在列中显示）。随着更多的层被加入（添加的层用粗体显示），结构的深度从左（A）到右（E）增加。卷积层参数表示为“卷积层<感受野大小>-<通道数>”。为了简洁，我们不展示 ReLU 激活函数。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					
参数数量（单位为百万）					
Network	A.A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

图 3 [16] 中给出的 VGG-Net 描述

VGG 的作者经过研究发现越深的网络训练效果越好，并且局部响应标准化对提高训练效果帮助不大，反而耗费更多训练时间； 3×3 的卷积层比起 1×1 的来说能学习到更大图像特征，即使是 3 个串联同大小卷积层也要比单一 7×7 卷积层有着更多迭代的非线性变换，即可以学习到更多特征。

现可直接通过 Keras 包来进行优化的 VGG 模型的使用，再利用迁移学习，便可只用几行代码完成对一个图像数据集的训练。具体来说，本论文将 VGG-Net 最后一层去掉，自定义最后一层全连接层，得到对某一电视剧在所有题材类别中的预测值，选取最可能的 3 个或 5 个体裁类别作为最终预测体裁。

2.7.2 运用于简介数据集的 Word2Vec 模型

类似地，本论文也将在文本数据集上运用神经网络算法来比较深度学习与传统机器学习算法。近年来一种十分有效的文本分类预测神经网络模型是 Word2Vec 模型。为了让深度学习能够像学习视觉特征一样学习文本特征，该模型将文本以分布式的方式表示出

来，形成类似于图像和音频类似地连续数据，具体来说，使用词向量（word embedding）将单词映射到高维空间，相近语义单词可以具有相似向量表示。这样一来，就可以在文本数据集中使用深度学习算法了。

为了将该模型应用于预测电视剧体裁，本论文同样要使用迁移学习，自定义输出层，获取体裁预测概率。

2.8 算法分类效果的评估指标

评估分类算法最基本的指标是用分类准确率(accuracy)来量化算法的性能^[18]，详细定义为分类器将测试样本预测为正确的分类的数量与总样本数之比。准确率确实在多数场合中能够衡量算法的有效性，但在多分类问题中，只用这一种判断方式则显得较为粗糙和单薄，往往不能准确合理的评估算法。因此，需要引入其他的评估方式：精确率（Precision）和召回率（Recall）。

首先需要对四种不同的分类指标进行定义。将某一分类场景所关注的标签集称为正向类（positives），其他标签称为负向类（negatives）：TP 是指将正向类预测为正向类数；FN 是指把正向类预测为负向类数；FP 是指把负向类预测为正向类数；TN 是指把负向类预测为负向类数^[18]。

精确率定义为

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

召回率定义为

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

此外，还可以用 F1-measure 来评估。F1 值就是精确率和召回率的调和均值，定义为

$$\frac{2}{F_1} = \frac{1}{P} + \frac{1}{R}$$

可以调整为

$$F_1 = \frac{2PR}{P+R} = \frac{2TP}{2TP + FP + FN}$$

F1 值随精确率和召回率正向变化。

以上的评估指标通常在二分类问题中出现，如果要应用于多分类问题，需结合宏平均、微平均两个概念，来考虑多标签分类问题在的分类器在不同分类标签的综合性能。

宏平均（Macro-averaging）是指将问题中每一分类的统计指标值求出一个算术均值^[19]，具体计算公式见表 2.

表 2 宏平均指标计算公式表

公式名称	公式
宏精确率 (Macro-Precision)	$P_{marco} = \frac{1}{n} \sum_{i=1}^n P_i$
宏召回率 (Macro-Recall)	$R_{marco} = \frac{1}{n} \sum_{i=1}^n R_i$
宏 F 值 (Macro-F Score)	$F_{marco} = \frac{2 \times P_{marco} \times R_{marco}}{P_{marco} + R_{marco}}$

微平均 (Micro-averaging) 是统计每个样本的各项指标，无视分类的不同，以此形成全局混淆矩阵^[19]，计算公式见表 3

表 3 微平均指标计算公式表

公式名称	公式
微精确率 (Macro-Precision)	$P_{micro} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i}$
微召回率 (Macro-Recall)	$R_{micro} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i}$
微 F 值 (Macro-F Score)	$F_{micro} = \frac{2 \times P_{micro} \times R_{micro}}{P_{micro} + R_{micro}}$

宏平均和微平均的不同之处在于权重设置的不同。宏平均将每个分类的指标赋予同一权重值，而微平均则是赋予每个样本同一权重。换句话说，在后者中占主导的是样本统计值多的类别。因此，宏平均更能反映在测试数据集中分类器对类别样本少的预测有效性，而微平均则通常反映分类器对大类别的预测有效性。再实际研究中一般多采用微平均。

本论文最终对比算法的统一评估指标是微平均下的精确率和召回率，详细设计见 3.5 节。这是因为，需用精确率衡量正确的预测结果，即预测正确的正向类占总数的百分比，是模型查准率的指标；召回率的作用则是考虑模型查全率的指标，代表了正确地被预测出的正向样本占所有应被预测出的正向样本数的比例。理论上一个模型的精确率和召回率应越高越好，但现实情况下由于多标签分类的天然属性，即每个样本没有一个固定的预测标签个数，这两个指标往往是“此消彼长”。举例来说，如果数据集中大多电视剧

只有一个体裁标签，那么精确率与召回率的比值就会非常低，因为模型会给大多数电视剧贴上多个体裁标签。所以设置两个评估指标才能更好的反映模型性能表现。

2.9 项目部署介绍

本论文的模型在 python 语言上编写，代码将部署在 Google 的开源环境 Colabulatory 上。Colabulatory 基于 Jupyter Notebook 平台开发，最大的优势是提供云端 GPU 计算服务。这一强大的计算资源是本论文可以灵活、快速研究问题的关键。

Jupyter Notebook 是一个 python 的 Web 应用程序，支持多达 40 中不同的编程语言，通过交互的方式实现实时代码编写、数据分析和可视化等功能^[20]，近年来受到广泛应用。

本论文的代码编写形式与传统 python 项目有些许差异，将以“教程”（tutorial）的形式展现。也就是说，会将程序分隔在一个个“cell”中，每个 cell 完成一个或几个简单的功能。这是由于出于对代码可读性和简洁性的考虑，使得任何人都可以用该 tutorial 轻松完成对本研究问题的实际操作，包括生成数据集和算法实现。因此，即便不能使用 Colabulatory，该 tutorial 也可以在本地环境运行。

在本地运行时，推荐利用开源程序 Anaconda 下载 Jupyter Notebook 以及所依赖的各种 python 包，该程序对 python 环境的管理和配置也十分友好。

2.10 本章小结

第二章主要从如何设计数据集、使用的算法以及评估的方式对研究开展时的思路进行了总结。

在 2.1~2.3 节中，强调了研究中数据集的重要性，展示了本论文是如何从数据的代表性入手，进行数据集的选择。接下来，2.4-2.7 节具体的从理论上描述了本论文将使用的三类算法，分别是支持向量机、多项式朴素贝叶斯和卷积神经网络。接着 2.8 节介绍将在算法设计中使用的评估指标。最后介绍了代码的编写语言和环境。

第三章 文本多标签分类模型算法设计

3.1 生成数据集的前期准备

数据集的前期准备是在 TMDB 网站上获取 API。在 TMDB 官网获得一个免费帐号后点击右上角的账号头像并选择“setting”。进入设置页面后，选择 API 选项，即可看到申请 API 的界面。天下玩必填信息并提交，就可以得到一个新生成的个人 API。但使用 API 访问 TMDB 时需注意它只允许每十秒钟 40 次请求的频率。因此当使用 API 请求数据时，遵循这个要求最好的方法是在循环时使用命令 time.sleep(1)，并且使用 python 的 try/except 模块能够很好的实现这一目的，保证在第一次请求失败后第二次的“try”首先使用 python 的 sleep 函数暂停请求然后再次尝试连接请求成功获得数据。示例如下：

```
1. try:  
2.     search.tvshow(query=tvshow)  
3. except:  
4.     try:  
5.         time.sleep(1)  
6.         search.tvshow(query=<i>tvshow_name</i>)  
7.     except:  
8.         print("Failed second attempt too, check if there's any error in request")
```

3.1.1 获取数据的开源 python 包描述

在访问 IMDB 与 TMDB 网站的数据库时，需要使用的 python 开源包主要有 IMDbPY 和 tmdbsimple。

IMDbPY 是一个能获取 IMDB 数据库中几乎所有媒体信息的 python 包，并且允许使用者将数据保存至本地。

使用时，需要先安装 imd bpy 包，然后再导入；值得注意的是，本论文使用的是 6.6 版本的 imd bpy。通过创建一个 IMDB 目标来接入 IMDB 的数据库，之后可以根据需要获取信息。从 IMDB 数据库获取电视剧‘Game of thrones’的示例如下：

```
1. from imdb import IMDb  
2. imbd_object = IMDb()  
3. results = imbd_object.search_movie('Game of Thrones')
```

tmdb simple 与 IMDbPY 略有不同，如名所示，是一个用 python 写的 TMDB 的 API 的封装包。该作者利用 tmdb 网站上原本定义的一些接口函数进行重新封装定义，以达到更简洁的使用感，简化使用者的代码并且同时让用户得到全面的关于电影和电视剧的相关信息。TMDB 的函数与 tmdb simple 的封装类实际上也还是一一映射的，只不过利用了

python 类的特性使数据的获得更加简单。

使用 tmdbsimple 前也需要安装，并且定义 API 的密钥使必须的。以下是一个利用 tmdbsimple 进行电影名的搜索示例：

```

1. import tmdbsimple as tmdb
2. tmdb.API_KEY = 'YOUR_API_KEY_HERE'
3. movie = tmdb.Movies(603)
4. response = movie.info()
5. movie.title

```

3.1.2 IMDB 与 TMDB 数据的对比抉择

这一步的目的是为了在构建数据集时选择数据的来源数据库，因为该两个网站相同电影电视剧的信息是重合的。通过分别获取 IMDB 与 TMDB 字典类型的目标所定义的键名以及对类别的不同定义方式，来进行抉择。具体区别如表 4 所示。可见，IMDB 对体裁的定义是缺乏体裁类别的 ID 编号的，会复杂化训练数据的矩阵转换，因此本论文将使用 TMDB 数据库进行数据集的构建。

表 4 IMDB 和 TMDB 字典目标在键名与体裁定义上的不同对比

以电视剧 'Game Of Thrones'举例	键名	类别定义方式
IMDB	['title', 'kind', 'year', 'cast', 'genres', 'runtimes', 'countries', 'country codes', 'language codes', 'color info', 'aspect ratio', 'sound mix', 'certificates', 'number of seasons', 'rating', 'votes', 'cover url', 'plot outline', 'languages', 'series years', 'akas', 'seasons', 'writer', 'production companies', 'distributors', 'special effects', 'other companies', 'plot', 'synopsis', 'canonical title', 'long imdb title', 'long imdb canonical title', 'smart canonical title', 'smart long imdb canonical title', 'full-size cover url']	['Action', 'Adventure', 'Drama', 'Fantasy', 'Romance']]
TMDB	['backdrop_path', 'created_by', 'episode_run_time', 'first_air_date', 'genres', 'homepage', 'id', 'in_production', 'languages', 'last_air_date', 'last_episode_to_air', 'name', 'next_episode_to_air', 'networks', 'number_of_episodes', 'number_of_seasons', 'origin_country', 'original_language', 'original_name', 'overview', 'popularity', 'poster_path', 'production_companies', 'seasons', 'status', 'type', 'vote_average', 'vote_count']	[{'id': 10765, 'name': 'Sci-Fi & Fantasy'}, {'id': 18, 'name': 'Drama'}, {'id': 10759, 'name': 'Action & Adventure'}]]

3.1.3 构建数据集的方法定义及所需工具

首先，获取数据时需要访问的 url 定义如下,其中 api_key 是每个人自己的 API:“url = ‘https://api.themoviedb.org/3/movie/1581?api_key=’ + ‘api_key’ ”;读取数据需要使用 urllib.response 包访问该地址。

其次，需要定义构建 TMDB 数据集时所需的一些封装函数以便获取电视剧的某一具体所需信息，函数的定义名规定为 get_movie_requiredInfo(),其中 requiredInfo 是可替换字段。这是因为使用 tmdbSimple 的搜索类得到的是一个电影或电视剧目标的实例，之后还需要通过调用方法得到字典类型的值。因此定义函数将会使代码更加简洁明了。所定义的四个函数分别是获取电视剧的海报，ID 代码，字典类型的全部详细信息以及所属体裁：

1. grab_poster_tmdb()
2. get_tvshow_id_tmdb()
3. get_tvshow_info_tmdb()
4. get_tvshow_genres_tmdb()

具体地说，数据集中选取的电视剧是在 TMDB 网站上最热门排名的前 10000 部电视剧，使用的具体方法是 popular()。需注意，由于利用 popular()返回得到的电视剧目标对体裁的描述使用的是‘genre_ids’，即使用的是体裁类别的 ID 编码，因此还需要创建一个对应体裁名称与其 ID 的字典，将其定义为‘Genre_ID_to_name’。

另外需要介绍的一个保存抓取到的电视剧数据集的 python 包是 pickle。利用 pickle 可以实现字典、列表的压缩，保存和提取。使用时先以“写”的方式打开一个自定义 pickle 后缀的文件，然后使用 pickle.dump 将需要保存的文件内容写入该文件。利用 pickle 提取文件时则只需以“读”的方式打开目标文件，然后用 pickle.load 将文件赋予给一个变量进行操作。

电视剧数据集获取的伪代码如下：

1. all_tvshows=tmdb.TV()
2. top10000_tvshows=[]
3. **for** i **in** range(1,51):
4. tvshows_on_this_page=all_tvshows.popular(page=i)[‘results’]
5. top10000_tvshows.extend(tvshows_on_this_page)
6. f3=open(‘tvshow_list.pkl’,‘wb’)
7. pickle.dump(top10000_tvshows,f3)
8. f3.close()

3.2 构建文本数据集的详细设计

用电视剧的情景简介作为体裁预测的输入 X 和用电视剧的体裁作为输出 Y 的关键的一步是数据清理，即将简介和体裁表示转换成可训练的矩阵，它包括数据的转换和修正。

3.2.1 二进制向量表示电视剧体裁标签得到输出矩阵 Y

二进制向量（Binarized vector）表示是一个非常常见的机器学习存储和表示数据的方式，可以有效地将 n 个分类标签的变量减少到 n 个二进制指标变量。比如输入样本 A, B 的标签为[(3),(2,4)]，指样本 A 有一个标签 3，样本 B 有两个标签 2 和 4。二进制向量表示就是用 1 来表示样本有此标签，0 表示无。因此，二进制版本的上述列表为 [(0,0,1,0),(0,1,0,1)]。将电视剧的体裁转换为二进制向量表示得到输出 Y 的伪代码如下：

```
1. genres=[]
2. all_ids=[]
3. for i in range(len(tvshows_with_overviews)):
4.     tvshow=tvshows_with_overviews[i]
5.     id=tvshow['id']
6.     genre_ids=tvshow['genre_ids']
7.     genres.append(genre_ids)
8.     all_ids.extend(genre_ids)
9. from sklearn.preprocessing import MultiLabelBinarizer
10. mlb=MultiLabelBinarizer()
11. Y=mlb.fit_transform(genres)
```

接下来的一步是检查类别名称是否与 ID 编号一一对应。如果 Y 的形状[a,b]中的 b 大于在之前 3.1.3 节中定义的‘Genre_ID_to_name’字典的键名个数，则说明在数据集中出现了新的体裁 ID 编号没有被转换成可读的英文题材名称。此时需要通过人工搜寻找到新出现的 ID 编号在 TMDB 网站中对应的体裁，将其手动加入该字典。

3.2.2 预处理电视剧简介输入矩阵 X

最后的步骤是得到输入数据 X 的矩阵，即把电视剧的简介存入矩阵中。这一方法被称为词袋建模（Bag Of Words），在自然语言处理领域很常见。词袋模型是使用一组忽略语义和语序的单词（tokens or words）表示一段文本，甚至一篇文章。具体地，将所有原文本集合中的所有不同单词都看作可能出现在目标文本中的不同对象，然后可以形象的将每一个电影简介文本都看作一个“袋子”，将会装入不同的对象。其中二进制向量表示将再次被使用，如同转化 Y 矩阵那样。但在里面的代码实现采用 scikit-learn 的封装函数 CountVectorizer([parameters])。

但由于这样表示文本时对象数量常常达到上万个，而维度太高将会降低训练的表现

[22]，为了解决高维度这一问题，还需引入另外一个常用的语言模型 TF-IDF，即词频-逆向文档频率模型（Term Frequency-Inverse Document Frequency model）。该模型认为决定一个文本文件（在本论文中将每个简介视为一个文本文件）特征的对象是那些在该文本文件出现频繁而在其他文件中偶尔出现的单词。举例来说，在科幻电影黑客帝国的简介中“computer”出现了两次，但这一情况在其他电影或电视剧中很少存在。因此利用这一思想可以有效地降低维度，通过 TF-IDF 模型去除对特征决定影响极小的单词对象。Python 实现这一目的只需在 CountVectorizer() 中定义两个参数 max_df 与 min_df： min_df 参数表示将出现概率极小的单词去除， max_df 参数是为了去除那些在每个电视剧简介中几乎都有的单词。本论文使用的参数是 max_df=0.97 与 min_df=0.05。

这一步骤的伪代码如下，先建立一个只包含单词的电视剧简介的列表，之后用上述函数 CountVectorizer() 转换成目标二进制矩阵 X：

```
1. from sklearn.feature_extraction.text import CountVectorizer
2. import re
3. content=[]
4. for i in range(len(tvshows_with_overviews)):
5.     tvshow=tvshows_with_overviews[i]
6.     id=tvshow['id']
7.     overview=tvshow['overview']
8.     overview=overview.replace(';', '')
9.     overview=overview.replace('.', '')
10.    content.append(overview)
11. vectorize=CountVectorizer(max_df=0.97, min_df=0.005)
12. X=vectorize.fit_transform(content)
```

3.3 判别模型和生成模型训练过程

3.3.1 使用 TF-IDF 模型对词袋中的词设置权重

电视剧简介介绍了电视剧的主要故事情节，因此可以用于对电视剧体裁预测的依据内容。但文本的特性决定了不同词之间的重要程度是由区别的，所以需要给能强烈暗示主题且出现频率较高的词赋予更高的权重。TF-IDF 模型，如上一节介绍，也可以用来决定词袋中不同词的权重值。详细来说，IDF（Inverse Document Frequency）部分做的工作是发现一些在大多文本中普遍存在的单词，如“a, an, the”。这些单词一般没有任何特征指示性。TF（Term Frequency）部分所做的任务正好相反，是去给一段文本中特征指示性较强的词更高的权重，而指示型稍弱词赋予较低的权重。Sklearn 提供了方法 TfidfTransformer() 用于实现此目的。

3.3.2 支持向量机判别模型代码设计

首先，将全部文本数据集分为训练集和测试集，比例为 7: 3。

为了保持代码的简洁，本论文将直接使用 sklearn 的 multiclass 多标签分类器进行一对
一和一对余支持向量机模型建构。同时，使用精确率、召回率和 F1-score 对每个标签的
分类结果进行观测。

特别地，机器学习算法需定义一个网格搜索函数进行最优参数的查找并同时进行训练
后的预测，该函数将同时运用于判别模型和生成模型，其伪代码如下：

```

1. def grid_search(train_x, train_y, test_x, test_y, genre_names, parameters, pipeline):
2.     grid_search_tune = GridSearchCV(pipeline, parameters, cv=2, n_jobs=3, verbose=10)
3.     grid_search_tune.fit(train_x, train_y)
4.     best_clf = grid_search_tune.best_estimator_
5.     predictions = best_clf.predict(test_x)
6.     print(classification_report(test_y, predictions, target_names=genre_names))

```

定义好分类器的 pipeline 和 parameters 的参数后，就可以直接进行训练了。

3.3.3 多项式朴素贝叶斯生成模型代码设计

多项式朴素贝叶斯使用的是 sklearn 的多项式贝叶斯封装函数，方法名为
MultinomialNB()。它的参数一共有三个，参数 α 即平滑参数 λ ，默认为 1；fit_prior 是一个布尔值参数，表示是否考虑先验概率，默认为 TRUE，当然也可以用第三个自定义参数
class_prior 设置先验概率，否则模型将自己根据输入数据计算先验概率，模型先验概率的
计算公式为：

$$P(\gamma = C_k) = \frac{N_k}{N}, \text{ 其中 } N_k \text{ 为 } k \text{ 标签的样本统计值, } N \text{ 是训练样本数。}$$

本论文的该模型参数定义为：

```

1. pipeline = Pipeline([
2.     ('clf', OneVsRestClassifier(MultinomialNB(
3.         fit_prior=True, class_prior=None))),
4. ])
5. parameters = {
6.     'clf__estimator_alpha': (1e-2, 1e-3)
7. }

```

3.3.4 Word2Vec 迁移学习模型的训练过程设计

本论文使用 Word2Vec 神经网络模型抽取文本特征。但因为单词数较小，该网络甚至
不需要前向传播算法，并且特征向量也直接存储在一个字典里，直接使用这个所有人都能
下载的开源字典便可以获得所需的单词的 Word2Vec 特征。建模方式是使用在 gensim

包中的 `models.KeyedVectors.load_word2vec_format()` 方法。应用后，每个单词在数学上从 3-4 个维度长度增加到了 300 维度。

下一步是对简介文本预处理。本论文所做的唯一预处理是使用 python 的开源包的 NLTK 删除在大多数文本中普遍存在的单词，如‘a’，‘but’，也可以称之为间隔词。之后删除数据集中简介只包括间隔词或者没有单词包含 Word2Vec 特征的电视剧。训练集是随机挑选 70% 的预处理后数据集，其余的归为测试集。

神经网络只有一层，经过 ReLU 输出激活函数和 softmax 函数得到最后的体裁预测概率损失函数使用二进制交叉熵。接着用方法 `.fit(training features, training labels, epochs, batch_size, verbose)` 训练模型二次。之前得到的样本特征矩阵的 70% 作为训练集，也就是第一个参数值，第二个参数是其对应的训练标签。模型一行行地载入训练数据时需要分批，这是因为 RAM 空间有限。后两个参数则是规定了每批数据的大小——`epochs` 是每个样本总共被载入模型的次数，也就是训练次数，`batch size` 为一次训练分成的批数。`Verbose` 值只能为 1 或 0，表示是否打印训练过程到屏幕。采用的第一次训练的参数值为 `epochs=10, batch_size=500`；第二次训练的参数值为 `epochs=10000, batch_size=500`。

对训练模型的评估将使用 `model.predict()`，它可以返回训练后模型在训练集上准确率；实际预测时使用 sklearn 的封装函数 `model.predict()`，返回预测标签概率矩阵。

第四章 图片多标签分类模型算法设计

4.1 构建海报图片数据集的详细设计

海报图片数据集的构建较为简单。主要是访问图片连接将其下载到本地的循环构建。首先，创立一个子文件夹存放所有海报。之后在每个不同的体裁分类下获取相同数量的电视剧，保存到字典变量中。之后定义一个循环获取该字典中所有电视剧的海报，这一步利用到 3.1.3 节中定义的 `grab_poster_tmdb()` 函数方法。注意有些电视剧可能会没有海报，因此需要将这些不符合要求的电视剧去除，以得到最后所需的有海报图片的电视剧列表，再次用 `pickle` 包保存。

4.2 VGG-Net 迁移学习模型训练过程

4.2.1 特征抽取方式

本论文所使用的 Keras-VGG 模型是一个已经优化好的视觉特征提取预训练模型，同时使用迁移学习重新设计最后全连接层的功能，使其可以计算出每个输入样本属于某个体裁标签的可能性。因此，所要做的工作是重新设计全连接层并决定最后输出前几个概率最大的体裁预测。

在大型数据集中已训练好的预训练模型的结构和权重可以通过迁移学习直接应用到本论文所研究的问题上，这带来的最大优势是可以节省训练成本，包括时间和内存，同时获得更优的训练结果。

那么具体用该预训练模型做怎么样的特征抽取呢？当海报进入模型后，网络学习特征的方式是让包含相似元素越多的图片空间距离越近。虽然说这样的特征抽取方式对于题材预测来说不是最理想的，但是可以获得有意义的训练结果的。比如海报上同时有枪和汽车的电视剧可以被预测为是‘Action’体裁的。这是因为一般的电视剧海报设计者想通过海报吸引观众的眼球，即用视觉最大化的展现电视剧的故事语义特征。所以说用海报作为体裁预测的依据是有代表性的。

4.2.2 VGG-Net 的迁移学习代码实现

代码设计的第一步是定义一个 VGG 模型变量，调用 Keras 中的 VGG 预训练模型；接着对每张图片进行特征抽取预处理，得到输入样本的特征矩阵 X，每张图片的特征维度为 25088。伪代码如下：

```
1. model = VGG16(weights='imagenet', include_top=False)
2. for mov in poster_movies:
3.     x = img_to_array(mov_img)
4.     x = np.expand_dims(x, axis=0)
5.     x = preprocess_input(x)
```

```
6.     features = model.predict(x)
7.     feature_list.append(features)
8. X = get_feature_array(feature_list)
```

最后一层全连接层，在Keras也被称为dense层，设计为先用ReLU激活函数把特征值降维到1024维和256维，最后用Sigmoid函数将对每一个体裁的预测概率限制在0和1之间，具体代码设计见图7：

```
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras import optimizers
model_visual = Sequential([
    Dense(1024, input_shape=(25088,)),
    Activation('relu'),
    Dense(256),
    Activation('relu'),
    Dense(28),
    Activation('sigmoid'),
])
opt = optimizers.rmsprop(lr=0.0001, decay=1e-6)

model_visual.compile(optimizer=opt,
                      loss='binary_crossentropy',
                      metrics=['accuracy'])
```

图7 使用迁移学习的VGG-Net全连接层详细设计

最后还是用.fit()方法和predict()进行训练和预测，两次训练过程的参数分别为epochs=10, batch_size=64; epochs=50, batch_size=64。

4.3 所有算法的统一评估设计

本论文所采用的所有算法将使用同一个自定义函数在同一尺度上进行评估。现在每一类中进行精确率和召回率的统计，最后求得算法平均值，这是参考了微平均的定义确定的评估方法。其中函数precision_recall()将同一标签的下所有TP(true positive)、FP(false positive)、FN(false negative)的值进行统计，利用2.8小节中对精确率和召回率

的定义进行计算。代码设计如图 6:

```

precs=[]
recs=[]
for i in range(len(test_movies)):
    if i%1==0:
        pos=test_movies[i]
        test_movie=movies_with_overviews[pos]
        gtids=test_movie['genre_ids']
        gt=[]
        for g in gtids:
            g_name=Genre_ID_to_name[g]
            gt.append(g_name)
        #      print predictions[i], movies_with_overviews[i]['title'], gt
        a,b=precision_recall(gt, predictions[i])
        precs.append(a)
        recs.append(b)

print np.mean(np.asarray(precs)), np.mean(np.asarray(recs))

```

图 6 算法统一评估代码设计

其中取得样本精确率和召回率的函数为:

```

1. def precision_recall(ground_truth,predictions):
2.     TP, FP, FN=0
3.     for t in ground_truth:
4.         if t in predictions:
5.             TP+=1
6.         else:
7.             FN+=1
8.     for p in predictions:
9.         if p not in ground_truth:
10.            FP+=1
11.    if TP+FP==0:
12.        precision=0
13.    else:
14.        precision=TP/float(TP+FP)
15.    if TP+FN==0:
16.        recall=0
17.    else:
18.        recall=TP/float(TP+FN)
19.    return precision, recall

```

第五章 算法测试以及结果分析

5.1 数据集介绍

5.1.1 数据集的统计数据

为了比照在不同数据集大小上算法表现的差异，本论文的数据集采用多个数量级进行研究。具体统计表格如下表 5。

表 5 本论文使用的训练数据集大小统计数据

数据集名称	数据集样本矩阵大小（样本数，特征维度）	数据集样本标签矩阵大小（样本数，标签数）
TMDB 电视剧简介数据集 (10,000 数量级)	(9802, 2259) (9727, 300)*	(9802, 27) (9727, 27)*
TMDB 电视剧简介数据集 (30,000 数量级)	(28203, 2166) (28185, 300)*	(28203, 29) (28185, 29)*
TMDB 电视剧海报数据集 (1,000 数量级)	(1277, 25088)	(1277, 22)
TMDB 电视剧海报数据集 (10,000 数量级)	(11362, 25088)	(11362, 27)

*用于 Word2Vec 模型的数据集经过词向量映射和预处理，因此大小略有差异

需要解释的是由于数据集的增大，标签数量也会不可避免地增加。这是因为当更多的电视剧包含进来时，会出现一些体裁较为小众的电视剧，比如‘TV movie’，‘sport’等。另外特征维度的不同是因为将输入样本用参数相同 TF-IDF 模型进行了特征筛选，删除的特征数受输入样本单词总量影响，因此会出现差异。作为参考，TMDB 官网大约收录了 81240 部电视剧。

5.1.2 利用成对比较分析数据集

由于本论文所研究的问题多标签分类，如果只看这些体裁标签的数量统计结果意义不大。因此需要使用成对比较（pairwise comparisons）^[21]来更好的展示体裁类别的内在联系，也可以由此看出数据集中的一些内在的偏重。举例来说，通常情况下浪漫与喜剧同时出现的概率要比‘浪漫’与‘恐怖’标签要高。这样做的目的是为了观察内在偏重对研究可能造成的影响，以后可能需要解决由于这样的数据集的不平衡性所造成的问题，或者如果并不需要做出减少影响的措施，确定本文假设建立在一个不平衡的数据集上将并不会影响最终结果也是很重要的。

对于 3.1 节所得到的万部电视剧本文将用体裁分布做成对比较分析。主要目标是观察哪些体裁同时出现在一部电视剧的概率较大。首先需要定义一个函数生成所有可能出现

的配对。之后将每个电视剧中的所有配对转成成列表进行统计并生成统计图。

生成的热力图将使用 seaborn 这个 python 包的 heatmap()方法。横纵坐标均是体裁名称，利用色块的深浅来表示数量的多少。结果如下图 4。

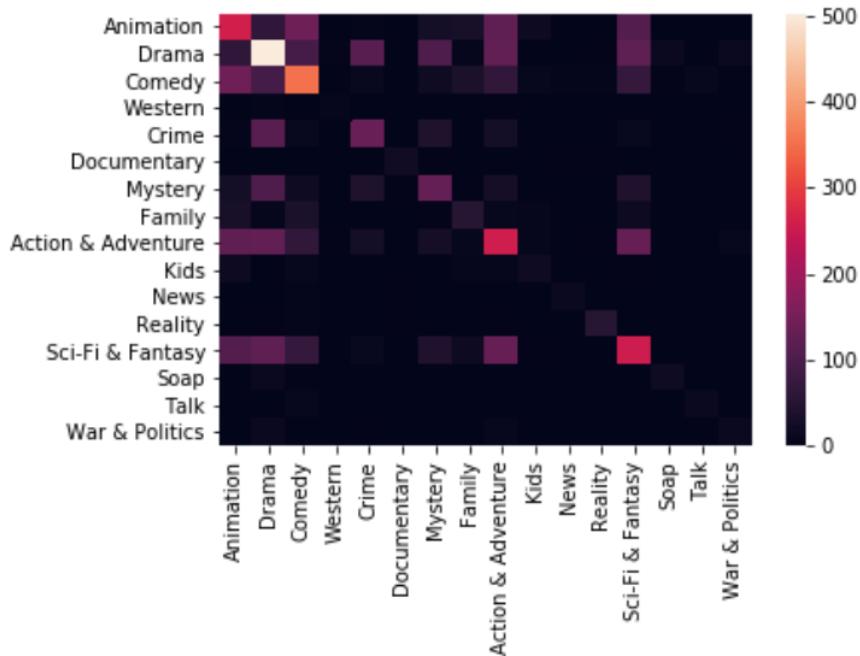


图 4 体裁类别成对分析的热力图（以 9802 部 TMDB 电视剧简介数据集为例）

上示热力图需注意对角线上的体裁。对角线上的点实际上均是自称对，如剧情与剧情同时出现的数量。因此，对角线上的数量统计仅表示某一体裁出现的次数。可以清晰的看出，drama，也就是剧情这一体裁出现次数最多，但这也是一个较为宽泛的描述标签。多部分的标签由于数据集较小，均表示为最深色。为了将统计数据以更好的逻辑表示，使用一种新的聚类算法——双聚类（Biclustering）。该方法可以对数据矩阵的两个坐标进行同时聚类，形成一个子矩阵。使用双聚类后可得到图 5 的热力图。可以看出，体裁相关性的观察变得更为简单了。同时出现次数较多的类有 Animation, Drama, Comedy, Action

& Adventure, Sci-Fi & Fantasy, 符合实际情况。

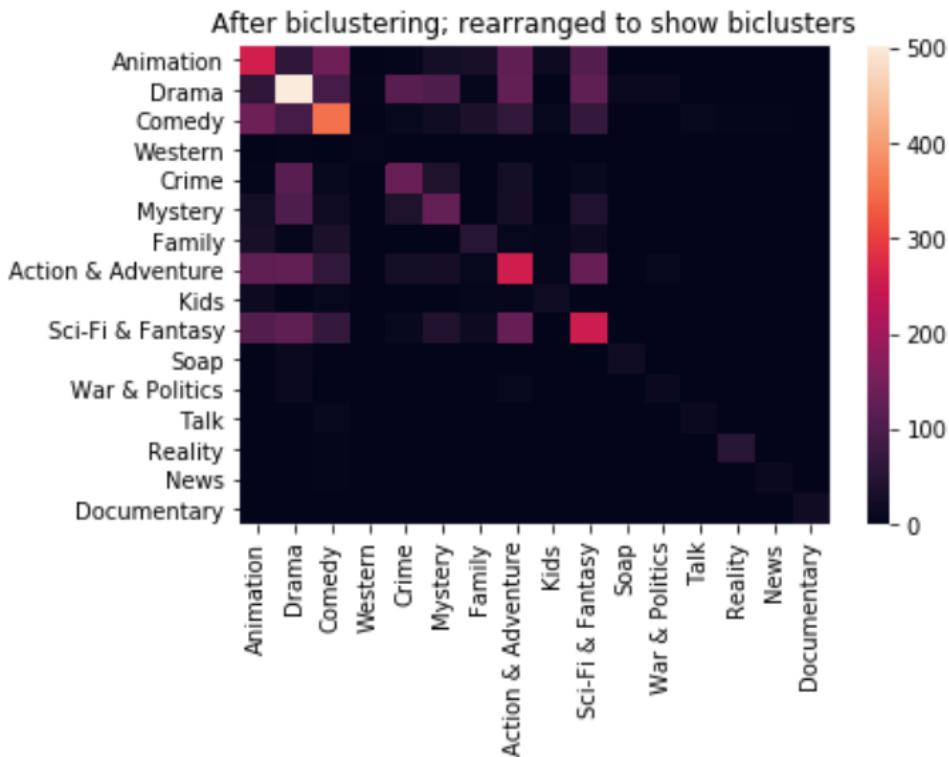


图 5 使用双聚类后的体裁类别成对分析的热力图（以 9802 部 TMDB 电视剧简介数据集为例）

事实上，双聚类得到的分类特征能给多标签分类问题提供新的启示。比如将这些体裁分成四大类（分类并不唯一）：Drama(Drama, Soap), Uplifting(Animation, Comedy, Family, Action & Adventure, Sci-Fi & Fantasy, Kids), Exciting(Western, Crime, Mystery, War & Politics), Real(Talk, Reality, news, Documentary)，转化为唯一标签分类问题。这样的优势是可以将增加数据集的平衡性，算法在分类时可以只需将一部电影分类为一种体裁，提高算法表现。还有，基于对数据集成对比较的研究，为我们在第五章的结果分析提供了重要的依据，它展示了为什么算法会经常把一些体裁一同预测出来。

5.2 多标签分类模型的评估指标统计

本小节将统计第三、四章中介绍的在电视剧的简介和海报数据集上预测体裁的所有算法的测试结果，以及使用同一评估指标（指标设计见 4.3 节）得到的精确率和召回率。特别需要说明的是，5.1.1 至 5.1.3 节的机器学习算法测试结果的评估报告是用 classification report 方法生成的。在 5.1.4 节评估总结中，虽然缺乏对电视剧信息进行研究的文献，但为了更好的展示算法的有效性，加入了一些参考数据——电影的简介和图片数据集在同一体裁或多个体裁分类的 state-of-the-art 的模型测试结果。

5.2.1 多类别支持向量机使用参数及测试结果

在 9802 条 TMDB 电视剧简介数据集上使用一对一和一对余的多类别支持向量机模型的最佳参数以及每一体裁分类的评估指标如图 8:

```
OneVsRestClassifier(estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovo', degree=3, gamma='auto', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False),
    n_jobs=1)
```

	precision	recall	f1-score	support
Action & Adventure	0.00	0.00	0.00	3
Animation	0.00	0.00	0.00	8
Comedy	0.84	0.65	0.73	524
Crime	0.69	0.59	0.63	933
Documentary	0.00	0.00	0.00	0
Drama	0.00	0.00	0.00	5
Family	0.00	0.00	0.00	6
Kids	0.74	0.30	0.42	632
Mystery	0.00	0.00	0.00	2
News	0.88	0.52	0.65	69
Reality	0.00	0.00	0.00	3
Sci-Fi & Fantasy	0.66	0.40	0.50	321
Soap	0.78	0.42	0.55	342
Talk	0.00	0.00	0.00	4
War & Politics	0.55	0.09	0.16	288
Western	0.00	0.00	0.00	4
Romance	0.00	0.00	0.00	21
Science Fiction	0.60	0.02	0.04	320
Music	0.00	0.00	0.00	2
Action	0.62	0.19	0.29	446
Horror	0.46	0.04	0.07	161
Fantasy	0.95	0.74	0.83	279
Adventure	0.79	0.41	0.54	301
War	0.72	0.25	0.37	408
TV Movie	0.81	0.31	0.45	126
NULL	0.81	0.31	0.45	148
History	0.59	0.10	0.17	104
Thriller	0.00	0.00	0.00	0
avg / total	0.71	0.37	0.46	5460

```

Best parameters set:
[('clf', OneVsRestClassifier(estimator=LinearSVC(C=1.3, class_weight=None, dual=True, fit_intercept=True,
intercept_scaling=1, loss='squared_hinge', max_iter=1000,
multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
verbose=0),
n_jobs=3))]

Applying best classifier on test data:
      precision    recall   f1-score   support

Action & Adventure     0.00     0.00     0.00       3
  Animation     0.00     0.00     0.00       8
    Comedy      0.78     0.71     0.75     524
    Crime       0.66     0.61     0.63     933
  Documentary     0.00     0.00     0.00       0
    Drama       0.00     0.00     0.00       5
    Family       0.00     0.00     0.00       6
    Kids        0.56     0.44     0.50     632
  Mystery       0.00     0.00     0.00       2
    News         0.84     0.54     0.65      69
  Reality       0.00     0.00     0.00       3
Sci-Fi & Fantasy     0.59     0.42     0.49     321
  Soap          0.67     0.52     0.59     342
    Talk         0.00     0.00     0.00       4
War & Politics      0.44     0.23     0.30     288
  Western       0.00     0.00     0.00       4
  Romance       0.00     0.00     0.00      21
Science Fiction      0.51     0.22     0.31     320
  Music          0.00     0.00     0.00       2
  Action         0.49     0.32     0.39     446
  Horror         0.42     0.22     0.29     161
  Fantasy        0.93     0.74     0.82     279
  Adventure      0.69     0.50     0.58     301
    War           0.57     0.41     0.48     408
TV Movie          0.72     0.40     0.51     126
  NULL           0.65     0.41     0.50     148
  History         0.51     0.23     0.32     104
  Thriller        0.00     0.00     0.00       0

avg / total     0.62     0.47     0.53     5460

```

图 8 One-vs-One（上）和 One-vs-Rest（下） SVM 模型最佳训练参数以及 sklearn 评估报告（1）

在 28203 部 TMDB 电视剧简介数据集上的最佳参数以及每一体裁分类的评估指标如图 9：

OneVsRestClassifier(estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False), n_jobs=1)				
	precision	recall	f1-score	support
Action & Adventure	0.00	0.00	0.00	11
Animation	0.00	0.00	0.00	14
Comedy	0.86	0.70	0.77	1339
Crime	0.74	0.64	0.69	2952
Documentary	0.00	0.00	0.00	1
Drama	0.00	0.00	0.00	3
Family	0.00	0.00	0.00	15
Kids	0.86	0.53	0.65	2628
Mystery	0.00	0.00	0.00	11
News	0.89	0.42	0.57	76
Reality	0.00	0.00	0.00	5
Sci-Fi & Fantasy	0.58	0.19	0.29	475
Soap	0.82	0.65	0.72	1369
Talk	0.00	0.00	0.00	6
War & Politics	0.00	0.00	0.00	354
Western	0.00	0.00	0.00	0
Romance	0.00	0.00	0.00	13
Science Fiction	0.00	0.00	0.00	74
Music	0.00	0.00	0.00	441
Action	0.00	0.00	0.00	3
Horror	0.50	0.02	0.03	773
Fantasy	0.00	0.00	0.00	152
Adventure	0.87	0.66	0.75	245
War	0.78	0.35	0.48	640
TV MOVIE	0.67	0.13	0.21	666
Musical	0.69	0.07	0.12	137
History	0.71	0.08	0.14	131
Thriller	0.00	0.00	0.00	123
Sport	0.00	0.00	0.00	1
avg / total	0.69	0.45	0.52	12658

```

Best parameters set:
[('clf', OneVsRestClassifier(estimator=LinearSVC(C=1.3, class_weight=None, dual=True, fit_intercept=True,
                                                intercept_scaling=1, loss='squared_hinge', max_iter=1000,
                                                multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
                                                verbose=0),
   n_jobs=3))]

Applying best classifier on test data:
      precision    recall  f1-score   support

Action & Adventure     0.00     0.00     0.00      11
  Animation     0.00     0.00     0.00      14
    Comedy     0.81     0.72     0.76    1339
    Crime     0.72     0.65     0.68    2952
 Documentary     0.00     0.00     0.00       1
    Drama     0.00     0.00     0.00       3
   Family     0.00     0.00     0.00      15
    Kids     0.76     0.58     0.66    2628
  Mystery     0.00     0.00     0.00      11
    News     0.83     0.46     0.59       76
  Reality     0.00     0.00     0.00       5
Sci-Fi & Fantasy     0.57     0.30     0.40     475
   Soap     0.77     0.68     0.72    1369
   Talk     0.00     0.00     0.00       6
War & Politics     0.38     0.08     0.13     354
  Western     0.00     0.00     0.00       0
  Romance     0.00     0.00     0.00      13
Science Fiction     0.09     0.01     0.02      74
   Music     0.39     0.06     0.11     441
   Action     0.00     0.00     0.00       3
   Horror     0.41     0.18     0.25    773
  Fantasy     0.36     0.07     0.11     152
Adventure     0.83     0.69     0.75     245
   War     0.61     0.43     0.50     640
 TV MOVIE     0.54     0.25     0.34     666
  Musical     0.60     0.15     0.24     137
  History     0.48     0.17     0.25     131
Thriller     0.36     0.07     0.12     123
   Sport     0.00     0.00     0.00       1

 avg / total     0.67     0.50     0.56    12658

```

图 9 One-vs-One (上) 和 One-vs-Rest (下) SVM 模型最佳训练参数以及 sklearn 评估报告 (2)

由两个数据集上的 sklearn 参考评估指标可以看出，一对一向量机在准确率上略优于一对余方法，但一对余方法的微平均 F1-score 的确总高于一对一方法，这应该是一对一方法受到某些体裁标签特征不可分的影响，比如‘Horror’和‘Thriller’两个类型，一对一模型就无法进行分类。

5.2.2 多项式朴素贝叶斯使用参数及测试结果

在 9802 条 TMDB 电视剧简介数据集上使用多项式朴素贝叶斯模型的最佳参数以及每一体裁分类的评估指标如图 10，在 28203 部 TMDB 电视剧简介数据集上的最佳参数以及每一体裁分类的评估指标如图 11：

```
Best parameters set:  
[('clf', OneVsRestClassifier(estimator=MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True),  
n_jobs=1))]  
Applying best classifier on test data:  
precision    recall    f1-score   support  
  
Action & Adventure      0.00     0.00     0.00      7  
    Animation      0.00     0.00     0.00     11  
    Comedy         0.88     0.50     0.63    543  
    Crime          0.64     0.57     0.60    976  
    Documentary     0.00     0.00     0.00      1  
    Drama           0.00     0.00     0.00      5  
    Family          0.00     0.00     0.00      6  
    Kids            0.71     0.13     0.22    691  
    Mystery          0.00     0.00     0.00      4  
    News             0.79     0.34     0.48    67  
    Reality          0.00     0.00     0.00      3  
Sci-Fi & Fantasy       0.72     0.28     0.40    329  
    Soap            0.87     0.26     0.41    367  
    Talk             0.00     0.00     0.00      7  
War & Politics          0.53     0.10     0.17    299  
    Western          0.00     0.00     0.00      4  
    Romance          0.50     0.05     0.08    22  
Science Fiction          0.62     0.05     0.10    302  
    Music            0.00     0.00     0.00      4  
    Action           0.68     0.16     0.26    436  
    Horror           0.00     0.00     0.00    165  
    Fantasy          0.86     0.73     0.79    236  
    Adventure        0.88     0.31     0.45    321  
    War              0.69     0.21     0.32    389  
    TV Movie         0.94     0.24     0.38    124  
    NULL             0.84     0.12     0.21    135  
    History          0.75     0.03     0.06    98  
    Thriller         0.00     0.00     0.00      0  
  
avg / total      0.71     0.30     0.38    5552
```

图 10 MultinomialNB 模型最佳训练参数以及 sklearn 评估报告 (1)

```

Best parameters set:
[('clf', OneVsRestClassifier(estimator=MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True),
                               n_jobs=None))]

Applying best classifier on test data:
      precision    recall   f1-score   support
Action & Adventure     0.00     0.00     0.00      10
      Animation     0.00     0.00     0.00      13
      Comedy       0.91     0.43     0.59    1299
      Crime        0.73     0.53     0.62    2932
  Documentary     0.00     0.00     0.00       0
      Drama        0.00     0.00     0.00       5
      Family        0.00     0.00     0.00      14
      Kids         0.83     0.40     0.54    2602
      Mystery       0.00     0.00     0.00      15
      News          0.82     0.17     0.29      80
      Reality       0.00     0.00     0.00       7
Sci-Fi & Fantasy      0.55     0.17     0.25     428
      Soap          0.90     0.46     0.61    1384
      Talk          0.00     0.00     0.00       5
War & Politics        0.44     0.02     0.04     359
      Western       0.00     0.00     0.00       1
      Romance       0.00     0.00     0.00      15
Science Fiction        0.20     0.10     0.13      89
      Music          0.00     0.00     0.00     450
      Action         0.00     0.00     0.00       3
      Horror         0.49     0.05     0.09    776
      Fantasy        0.00     0.00     0.00     153
      Adventure      0.87     0.61     0.72    243
      War            0.82     0.16     0.27    592
TV MOVIE                 0.70     0.05     0.09    701
      Musical        0.71     0.04     0.07    136
      History        0.75     0.05     0.09    133
      Thriller       0.00     0.00     0.00    112
      Sport          0.00     0.00     0.00       1

  micro avg       0.79     0.34     0.47    12558

```

图 11 MultinomialNB 模型最佳训练参数以及 sklearn 评估报告（2）

5.2.3 Word2Vec 模型测试示例及结果

在 9802 条 TMDB 电视剧简介数据集上训练得到的评估准确率为 0.9403，利用 Word2Vec 模型分别取前 1 个、前 3 个和前 5 个预测概率最高的体裁作为电视剧体裁预测结果，可得到如图 12 的测试示例，预测电视剧选取原则是每预测 500 部输出一部的预测结果：

A

Our predictions for the tvshows are -

```
Predicted: ['Action & Adventure'] Actual: ['Animation', 'Comedy', 'Action & Adventure']
Predicted: ['Drama'] Actual: ['Drama']
Predicted: ['Drama'] Actual: ['Drama', 'Crime']
Predicted: ['Reality'] Actual: ['Mystery', 'Reality']
Predicted: ['Documentary'] Actual: ['Animation', 'Family', 'Kids', 'Sci-Fi & Fantasy']
Predicted: ['Crime'] Actual: ['Reality']
```

B

Our predictions for the tvshows are -

```
Predicted: ['Comedy', 'Animation', 'Action & Adventure'] Actual: ['Animation', 'Comedy', 'Action & Adventure']
Predicted: ['Comedy', 'Crime', 'Drama'] Actual: ['Drama']
Predicted: ['Sci-Fi & Fantasy', 'Action & Adventure', 'Drama'] Actual: ['Drama', 'Crime']
Predicted: ['Mystery', 'Documentary', 'Reality'] Actual: ['Mystery', 'Reality']
Predicted: ['Sci-Fi & Fantasy', 'Drama', 'Documentary'] Actual: ['Animation', 'Family', 'Kids', 'Sci-Fi & Fantasy']
Predicted: ['Documentary', 'Drama', 'Crime'] Actual: ['Reality']
```

C

Our predictions for the tvshows are -

```
Predicted: ['Crime', 'Drama', 'Comedy', 'Animation', 'Action & Adventure'] Actual: ['Animation', 'Comedy', 'Action & Adventure']
Predicted: ['Documentary', 'Mystery', 'Comedy', 'Crime', 'Drama'] Actual: ['Drama']
Predicted: ['Mystery', 'Comedy', 'Sci-Fi & Fantasy', 'Action & Adventure', 'Drama'] Actual: ['Drama', 'Crime']
Predicted: ['News', 'Talk', 'Mystery', 'Documentary', 'Reality'] Actual: ['Mystery', 'Reality']
Predicted: ['Animation', 'Action & Adventure', 'Sci-Fi & Fantasy', 'Drama', 'Documentary'] Actual: ['Animation', 'Family', 'Kids', 'Sci-Fi & Fantasy']
Predicted: ['Action', 'Comedy', 'Documentary', 'Drama', 'Crime'] Actual: ['Reality']
```

图 12 利用 Word2Vec 模型在相同 6 部电视剧上进行题材预测测试示例，红框标注表示预测正确：A、B、C 分别为选取预测概率最高的，前三的和前五的体裁作为最终预测结果（1）

训练 28203 部 TMDB 电视剧简介数据集的模型评估准确率为 0.9586，利用该模型分别取前 1 个、前 3 个和前 5 个预测概率最高的体裁作为电视剧体裁预测结果，可得到如图 13 的测试示例，选取示例方式与上述相同：

A

Our predictions for the tvshows are -

```

Predicted: ['Action & Adventure'] Actual: ['Animation', 'Drama', 'Comedy']
Predicted: ['Animation'] Actual: ['Animation', 'Comedy', 'Sci-Fi & Fantasy']
Predicted: ['Animation'] Actual: ['Animation']
Predicted: ['Drama'] Actual: ['Drama', 'Action & Adventure']
Predicted: ['Drama'] Actual: ['Drama']
Predicted: ['Drama'] Actual: ['Drama', 'Action', 'History', 'Crime', 'Romance']
Predicted: ['Drama'] Actual: ['Drama']
Predicted: ['Drama'] Actual: ['Drama']
Predicted: ['Comedy'] Actual: ['Comedy']
Predicted: ['Talk'] Actual: ['Comedy', 'Talk']
Predicted: ['Comedy'] Actual: ['Comedy', 'Kids']
Predicted: ['Comedy'] Actual: ['Comedy']
Predicted: ['Drama'] Actual: ['Documentary']
Predicted: ['Documentary'] Actual: ['Documentary']
Predicted: ['Documentary'] Actual: ['Documentary']
Predicted: ['Comedy'] Actual: ['Action & Adventure', 'Kids']
Predicted: ['Documentary'] Actual: ['Reality']

```

B

Our predictions for the tvshows are -

```

Predicted: ['Comedy', 'Animation', 'Action & Adventure'] Actual: ['Animation', 'Drama', 'Comedy']
Predicted: ['Action & Adventure', 'Comedy', 'Animation'] Actual: ['Animation', 'Comedy', 'Sci-Fi & Fantasy']
Predicted: ['Comedy', 'Drama', 'Animation'] Actual: ['Animation']
Predicted: ['Action & Adventure', 'Crime', 'Drama'] Actual: ['Drama', 'Action & Adventure']
Predicted: ['Crime', 'Comedy', 'Drama'] Actual: ['Drama']
Predicted: ['Comedy', 'Action & Adventure', 'Drama'] Actual: ['Drama', 'Action', 'History', 'Crime', 'Romance']
Predicted: ['Crime', 'Drama', 'Mystery'] Actual: ['Drama']
Predicted: ['Comedy', 'Sci-Fi & Fantasy', 'Drama'] Actual: ['Drama']
Predicted: ['Animation', 'News', 'Comedy'] Actual: ['Comedy']
Predicted: ['Comedy', 'Reality', 'Talk'] Actual: ['Comedy', 'Talk']
Predicted: ['Talk', 'Family', 'Comedy'] Actual: ['Comedy', 'Kids']
Predicted: ['Documentary', 'Drama', 'Comedy'] Actual: ['Comedy']
Predicted: ['Comedy', 'Reality', 'Drama'] Actual: ['Documentary']
Predicted: ['Drama', 'Reality', 'Documentary'] Actual: ['Documentary']
Predicted: ['Crime', 'Drama', 'Documentary'] Actual: ['Documentary']
Predicted: ['Reality', 'Sci-Fi & Fantasy', 'Comedy'] Actual: ['Action & Adventure', 'Kids']
Predicted: ['Drama', 'Family', 'Documentary'] Actual: ['Reality']

```

C

Our predictions for the tvshows are -

```

Predicted: ['Drama', 'Sci-Fi & Fantasy', 'Comedy', 'Animation', 'Action & Adventure'] Actual: ['Animation', 'Drama', 'Comedy']
Predicted: ['Sci-Fi & Fantasy', 'Kids', 'Action & Adventure', 'Comedy', 'Animation'] Actual: ['Animation', 'Comedy', 'Sci-Fi & Fantasy']
Predicted: ['Sci-Fi & Fantasy', 'Mystery', 'Comedy', 'Drama', 'Animation'] Actual: ['Animation']
Predicted: ['Comedy', 'Mystery', 'Action & Adventure', 'Crime', 'Drama'] Actual: ['Drama', 'Action & Adventure']
Predicted: ['Mystery', 'Reality', 'Crime', 'Comedy', 'Drama'] Actual: ['Drama']
Predicted: ['Romance', 'Mystery', 'Comedy', 'Action & Adventure', 'Drama'] Actual: ['Drama', 'Action', 'History', 'Crime', 'Romance']
Predicted: ['Action & Adventure', 'Soap', 'Crime', 'Drama', 'Mystery'] Actual: ['Drama']
Predicted: ['Action & Adventure', 'Mystery', 'Comedy', 'Sci-Fi & Fantasy', 'Drama'] Actual: ['Drama']
Predicted: ['Drama', 'Family', 'Animation', 'News', 'Comedy'] Actual: ['Comedy']
Predicted: ['Documentary', 'Kids', 'Comedy', 'Reality', 'Talk'] Actual: ['Comedy', 'Talk']
Predicted: ['Animation', 'Reality', 'Talk', 'Family', 'Comedy'] Actual: ['Comedy', 'Kids']
Predicted: ['Crime', 'Animation', 'Documentary', 'Drama', 'Comedy'] Actual: ['Comedy']
Predicted: ['Action & Adventure', 'Documentary', 'Comedy', 'Reality', 'Drama'] Actual: ['Documentary']
Predicted: ['News', 'Animation', 'Drama', 'Reality', 'Documentary'] Actual: ['Documentary']
Predicted: ['News', 'Comedy', 'Crime', 'Drama', 'Documentary'] Actual: ['Documentary']
Predicted: ['Animation', 'Family', 'Reality', 'Sci-Fi & Fantasy', 'Comedy'] Actual: ['Action & Adventure', 'Kids']
Predicted: ['Comedy', 'Action & Adventure', 'Drama', 'Family', 'Documentary'] Actual: ['Reality']

```

图 13 利用 Word2Vec 模型在相同 17 部电视剧上进行题材预测测试示例，红框标注表示预测正确：

A、B、C 分别为选取预测概率最高的，前三的和前五的体裁作为最终预测结果（2）

5.2.4 VGG-Net 模型测试示例及结果

在 1277 张 TMDB 电视剧海报数据集上训练的 VGG 模型分别取前 3 个和前 4 个预测概率最高的体裁作为电视剧体裁预测结果，可得到如图 14 的测试示例，预测电视剧选取原则是每预测 500 部输出一部的预测结果

A

Predicted: Kids, Comedy, Animation Actual: Animation, Comedy
 Predicted: Talk, War & Politics, Drama Actual: Western, Action & Adventure
 Predicted: Documentary, Mystery, Drama Actual: Drama, Crime, Mystery, Sci-Fi & Fantasy
 Predicted: Sci-Fi & Fantasy, Family, Animation Actual: Animation, Family, Kids
 Predicted: Sci-Fi & Fantasy, Mystery, Drama Actual: Drama, Mystery, Sci-Fi & Fantasy

B

Predicted: War & Politics, Kids, Comedy, Animation Actual: Animation, Comedy
 Predicted: Fantasy, Talk, War & Politics, Drama Actual: Western, Action & Adventure
 Predicted: War & Politics, Documentary, Mystery, Drama Actual: Drama, Crime, Mystery, Sci-Fi & Fantasy
 Predicted: Kids, Sci-Fi & Fantasy, Family, Animation Actual: Animation, Family, Kids
 Predicted: Soap, Sci-Fi & Fantasy, Mystery, Drama Actual: Drama, Mystery, Sci-Fi & Fantasy

图 14 利用 Word2Vec 模型在相同 6 部电视剧上进行题材预测测试示例，红框标注表示预测正确：A、B 分别为选取预测概率前三的和前四的体裁作为最终预测结果（1）

在 11362 张 TMDB 电视剧海报数据集上训练的 VGG 模型分别取前 3 个和前 4 个预测概率最高的体裁作为电视剧体裁预测结果，可得到如图 14 的测试示例，预测电视剧选取原则是每预测 500 部输出一部的预测结果：

A

Predicted: Kids, Animation, Soap Actual: Animation, Comedy, Kids, Soap
 Predicted: Soap, Comedy, Animation Actual: Animation, Comedy
 Predicted: Drama, Kids, Animation Actual: Animation, Mystery
 Predicted: Kids, Animation, Soap Actual: Animation, Drama, Kids, Soap
 Predicted: Drama, Animation, Soap Actual: Animation, Drama, Comedy, Kids, Soap

B

Predicted: Comedy, Kids, Animation, Soap Actual: Animation, Comedy, Kids, Soap
 Predicted: TV Movie, Soap, Comedy, Animation Actual: Animation, Comedy
 Predicted: Soap, Drama, Kids, Animation Actual: Animation, Mystery
 Predicted: Drama, Kids, Animation, Soap Actual: Animation, Drama, Kids, Soap
 Predicted: Kids, Drama, Animation, Soap Actual: Animation, Drama, Comedy, Kids, Soap

图 15 利用 Word2Vec 模型在相同 5 部电视剧上进行题材预测测试示例，红框标注表示预测正确：A、B 分别为选取预测概率前三的和前四的体裁作为最终预测结果（2）

在 5.1.3 和 5.1.4 节中的预测参考可以证明本论文对数据集的成对分析的重要性。许多题材之间的内在偏重使得它们总是会被标注在同一电视剧中，如‘Drama’和‘soap’，‘Animation’和‘Comedy’。

5.2.5 评估总结

对于本论文所有算法在同一评估指标的统计结果见表 6：

表 6 算法结果统计 (TOPⁿ 是指以排名前 n 的预测体裁类别作为最终预测值)

数据集	算法	精确率	召回率
9802 部 TMDB 电视剧简介数据集	一对余支持向量机	0.55407	0.50363
	一对一支持向量机	0.54080	0.41120
	多项式贝叶斯分类	0.47350	0.70015
	Word2Vec (TOP ¹)	0.64132	0.40859
	Word2Vec (TOP ³)	0.43371	0.73251
	Word2Vec (TOP ⁵)	0.32026	0.86608
28203 部 TMDB 电视剧简介数据集	一对余支持向量机	0.54391	0.58469
	一对一支持向量机	0.61103	0.51730
	多项式贝叶斯分类	0.48299	0.69853
	Word2Vec (TOP ¹)	0.64595	0.50273
	Word2Vec (TOP ³)	0.37409	0.79162
	Word2Vec (TOP ⁵)	0.26403	0.90071
1277 张 TMDB 电视剧海报数据集	VGG-Net (TOP ³)	0.53012	0.60020
	VGG-Net (TOP ⁴)	0.44277	0.66526
11362 张 TMDB 电视剧海报数据集	VGG-Net (TOP ³)	0.54762	0.68609
	VGG-Net (TOP ⁴)	0.45416	0.75378

可做参考的文献数据如表 7, 所参考的文献见表中的引用:

表 7 在电影数据集上利用与本论文相似算法的一些评估结果统计

数据集	算法	精确率	召回率
158,840 部 IMDB 电影简介数据集 ^[23]	One-vs-Rest SVM	0.66141	0.39572
	One-vs-Rest SVM (weighted)	0.47689	0.62533
	神经网络 (单层 100 个隐层单元, 利用 PCA, 降维维度数为 1000)	0.67630	0.41513
2400 部 TMDB 电影简介数据集 (仅 4 个体裁标签) ^[24]	RBF 核函数*的 SVM	0.7325	
2400 部 TMDB 电影海报数据集 (仅 4 个体裁标签) ^[24]	RBF 核函数的 SVM	0.6075	
338,789 部 IMDB 电影简介数据集 ^[25]	Multinomial Naive Bayes	0.507	0.563

*在[24]中使用的是用于单标签分类的线性不可分问题的 SVM 函数, 使用的核函数为 RBF(Radial Based Function), 作用是将训练样本映射到更高维空间, 使其可区分。

综合表 6 和表 7 来看，在传统机器学习算法中，一对一判别模型在精确率上优势明显，而生成模型能够取得较高的召回率——在 10000 数量级的电视剧简介数据集上使用朴素贝叶斯的召回率约为 70%；但明显地可以看到利用神经网络的深度学习模型的优越性，使用该模型可以取得约 65% 的精确率和高达 90% 的召回率。参考文献统计数据表 7，可以得出本论文所设计的电视剧体裁多标签分类模型是十分有效的。

5.3 结果分析

5.3.1 传统机器学习模型对比分析

在两个大小不同的数据集上训练得到的结果可以发现，判别模型和生成模型分别在精确率和召回率上有着各自的优势。从数学角度分析，这是由于两类模型解决问题的思路不同导致的。支持向量机算法思想是寻找最大决策超平面尽量区分数据，鲁棒性较强，是准确率较高的主要原因，由于一对训练的子分类器更多，因此精确率较一对余来说更高；而朴素贝叶斯通过条件概率做出分类，会受到特征之间相关性的影响^[26]，但能很好的利用题材之间的依赖性获得较高的召回率。从这一方面来看，就可以很好的理解这两种算法在评估结果上的差异了。但要承认的是，简介的体裁事实上是一类相对不明确的特征，分类的界定本身就是存在难度的。此外，朴素贝叶斯模型训练时速度比支持向量机稍快，训练效率较高，因为它不需要训练子分类器。

5.3.2 神经网络模型结果分析

用神经网络训练数据集的明显特征是数据量越大，训练模型表现会越好。特别地，使用迁移学习在预训练模型上进行针对具体问题搭建的模型，不仅训练结果明显超过机器学习算法，而且解决了深度学习对内存的依赖和时间的消耗。但与传统机器学习相比，其准确率要依赖于更大的数据集。

5.4 创新点总结

本论文选用了一个有趣的数据集进行多标签分类问题的研究，进行创新的方面可总结为以下三点：

- 1) 绝大多数在视觉媒体信息上的多标签分类研究是使用的电影信息数据集，但本论文从数学角度讨论数据集的选取并创新的尝试了更为困难的电视剧信息数据集作为训练数据，输入样本特征指示性更弱，但最终在精确率和召回率上表现出色，参考了电影数据集的 state-of-the-art。
- 2) 将使用的模型分为判别模型和生成模型这两大分类模型，对比它们在多标签分类问题上的表现。证明了在电视剧简介和海报数据集这样分类困难的数据中两类模型各自的优劣。
- 3) 利用迁移学习优化了多标签分类神经网络模型。不仅减少构建算法所需工作量，并同

时保证较好的训练结果。

总结

本论文从构思到完成总计花费近半年的时间，对多标签分类问题选取了一个极具研究价值的数据集，并最终设计出了效果较好的模型。研究课题难度较大，因为需要花费大量的时间学习机器学习知识，了解算法思想。论文全面的从数据集的选取、处理，算法理论分析和实际模型设计介绍了问题的研究过程，并将代码制作成清晰易操作的学习类教程模式，对以后相关课题的研究具有参考价值。

参考文献

- [1] Schapire, R.E. & Singer, Y. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 2000, 39:135-168.
- [2] 张微微. 基于标签相关性的多标签分类算法研究[D]. 浙江师范大学, 2016.
- [3] Breiman L. Random Forests[J]. *Machine Learning*, 2001, 45(1):5-32.
- [4] Boutell M R, Luo J, Shen X, Brown C M. Learning multi-label scene classification. *Pattern Recognition*, 2004, 37(9):1757-1771.
- [5] Grigoris Tsoumakas, Ioannis Vlahavas. Random k-Labelsets: An Ensemble Method for Multilabel Classification. In: Proceedings of the 18th European conference on Machine Learning (ECML '07), Joost N. Kok, Jacek Koronacki, Raomon Lopez Martaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron (Eds.). Berlin, Heidelberg: Springer-Verlag, 2007, 4701:406-417.
- [6] Zhang M-L, Zhou Z-H. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 2007, 40(7):2038-2048.
- [7] Elisseeff A, Weston J. A kernel method for multi-labelled classification. In: Dietterich T G, Becker S, Ghahramani Z, (Eds.). *Advances in Neural Information Processing Systems 14 (NIPS'01)*, Cambridge, MA: MIT Press, 2002, pp.681-687.
- [8] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01), T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.). Cambridge, MA, USA: MIT Press, 2001, pp.841-848.
- [9] Dan Jurafsky, James H. Martin. *Speech and Language Processing*.3rd edition. USA:Stanford University, 2018
- [10]Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, Gang Hua. CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training. *ICCV*, 2017,299:2764-2773
- [11]Hsu, C.-W & Chang, C.-C & Lin, C.-J. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University. 2003,101:1396-1400.
- [12]陈玉芹. 多类别科技文献自动分类系统[D]. 华中科技大学, 2008.
- [13]Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In: *Proceedings of Advances in Neural Information Processing Systems 25*.

- Lake Tahoe, Nevada, USA: Curran Associates, Inc., 2012, pp.1097-1105
- [14]Deng J, Dong W, Socher R, Li L J, Li K, Li F F. ImageNet: a large-scale hierarchical image database. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL: IEEE, 2009, pp.248-255
- [15]Szegedy C, Liu W, Jia Y Q, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA: IEEE, 2015, pp.1-9
- [16]Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv, 2016, pp.1409-1556
- [17]He Kai-Ming, Zhang Xiang-Yu, Ren Shao-Qing, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA:IEEE, 2016,pp.770-778
- [18]李航. 统计学习方法[M]. 清华大学出版社, 2012
- [19]廖一星. 文本分类及其特征降维研究[D]. 浙江大学, 2012.
- [20]薛煜阳. Jupyter Notebook 在 Python 教学中的应用探索[J]. 信息技术与信息化, 2018, 7:168-169.
- [21]Janez Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, J. Mach. Learn. Res. 2006:1532-4435 Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res., 2006, 7:1-30.
- [22]Pedro Domingos. A few useful things to know about machine learning. Commun. ACM, 2012, 55(10):78-87.
- [23]K W Ho. Movies' Genres Classification by Synopsis. USA: Stanford University, 2011.
- [24]Fu Z., Li B., Li J., Wei S. Fast Film Genres Classification Combining Poster and Synopsis. In: Intelligence Science and Big Data Engineering. Image and Video Data Engineering. He X. et al. (eds). IScIDE 2015. Lecture Notes in Computer Science, Springer, Cham, 2015, 9242(1):72-81.
- [25]Hoang, Quan. Predicting Movie Genres Based on Plot Summaries. CoRR, 2018, pp.1801-4813
- [26]Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 2014, 15(1):3133-3181

致谢

本论文的顺利完成离不开导师，同学还有母校四川大学的各方面支持和帮助，特别是张建州老师一直以来的督促和悉心指导，还有四川大学所提供的文献资源和学习环境，让我在论文设计期间能够不断弥补知识和经验的不足，不断明确研究目标、完善研究框架，一步步地拆分课题研究，尽最大的努力圆满完成这篇毕业论文。

附录 1

英文原文

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* & Andrew Zisserman*

Visual Geometry Group, Department of Engineering Science, University of Oxford

{Karen,az}@robots.ox.ac.uk

[Abstract] In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision..

1. INTRODUCTION

Convolutional networks (ConvNets) have recently enjoyed a great success in large-scale image and video recognition (Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014; Simonyan & Zisserman, 2014) which has become possible due to the large public image repositories, such as ImageNet(Deng et al., 2009), and high-performance computing systems, such as GPUs or large-scale distributed clusters (Dean et al., 2012). In particular, an important role in the advance of deep visual recognition architectures has been played by the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2014), which has served as a test bed for a few generations of large-scale image classification systems, from high-dimensional shallow feature encodings (Perronnin et al., 2010) (the winner of ILSVRC-2011) to deep ConvNets (Krizhevsky et al., 2012) (the winner of ILSVRC-2012).

With ConvNets becoming more of a commodity in the computer vision field, a number of attempts have been made to improve the original architecture of Krizhevsky et al. (2012) in a bid to achieve better accuracy. For instance, the best-performing submissions to the ILSVRC- 2013 (Zeiler & Fergus, 2013; Sermanet et al., 2014) utilised smaller receptive window size and smaller stride of the first convolutional layer. Another line of improvements dealt with training and testing the networks densely over the whole image and over multiple scales (Sermanet et al., 2014; Howard, 2014). In this paper, we address another important aspect of ConvNet architecture design – its depth. To this end, we fix other parameters of the architecture, and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3×3) convolution filters in all layers.

As a result, we come up with significantly more accurate ConvNet architectures, which not only achieve the state-of-the-art accuracy on ILSVRC classification and localisation tasks, but are also applicable to other image recognition datasets, where they achieve excellent performance even when used as a part of a relatively simple pipelines (e.g. deep features classified by a linear SVM without fine-tuning). We have released our two best-performing models¹ to facilitate further research.

The rest of the paper is organised as follows. In Sect. 2, we describe our ConvNet configurations. The details of the image classification training and evaluation are then presented in Sect. 3, and the configurations are compared on the ILSVRC classification task in Sect. 4. Sect. 5 concludes the paper. For completeness, we also describe and assess our ILSVRC-2014 object localisation system in Appendix A, and discuss the generalisation of very deep features to other datasets in

Appendix B. Finally, Appendix C contains the list of major paper revisions.

2 CONVNET CONFIGURATIONS

To measure the improvement brought by the increased ConvNet depth in a fair setting, all our ConvNet layer configurations are designed using the same principles, inspired by Ciresan et al. (2011); Krizhevsky et al. (2012). In this section, we first describe a generic layout of our ConvNet configurations (Sect.2.1) and then detail the specific configurations used in the evaluation (Sect.2.2). Our design choices are then discussed and compared to the prior art in Sect. 2.3.

2.1 ARCHITECTURE

During training, the input to our ConvNets is a fixed-size 224×224 RGB image. The only pre-processing we do is subtracting the mean RGB value, computed on the training set, from each pixel. The image is passed through a stack of convolutional (conv.) layers, where we use filters with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations we also utilise 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1 pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2.

A stack of convolutional layers (which has a different depth in different architectures) is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU (Krizhevsky et al., 2012)) non-linearity. We note that none of our networks (except for one) contain Local Response Normalisation (LRN) normalisation (Krizhevsky et al., 2012): as will be shown in Sect. 4, such normalisation does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time. Where applicable, the parameters for the LRN layer are those of (Krizhevsky et al., 2012).

2.2 CONFIGURATIONS

The ConvNet configurations, evaluated in this paper, are outlined in Table 1, one per column. In the following we will refer to the nets by their names (A–E). All configurations follow the generic design presented in Sect. 2.1, and differ only in the depth: from 11 weight layers in the network A (8 conv. and 3 FC layers) to 19 weight layers in the network E (16 conv. and 3 FC layers). The width of conv. layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

In Table 2 we report the number of parameters for each configuration. In spite of a large depth, the number of weights in our nets is not greater than the number of weights in a more shallow net with larger conv. layer widths and receptive fields (144M weights in (Sermanet et al., 2014)).

2.3 DISCUSSION

Our ConvNet configurations are quite different from the ones used in the top-performing entries of the ILSVRC-2012 (Krizhevsky et al., 2012) and ILSVRC-2013 competitions (Zeiler & Fergus, 2013; Sermanet et al., 2014). Rather than using relatively large receptive fields in the first conv. layers (e.g. 11×11 with stride 4 in (Krizhevsky et al., 2012), or 7×7 with stride 2 in (Zeiler & Fergus, 2013; Sermanet et al., 2014)), we use very small 3×3 receptive fields throughout the whole net, which are convolved with the input at every pixel (with stride 1). It is easy to see that a stack of two 3×3 conv. layers (without spatial pooling in between) has an effective receptive field of 5×5 ; three such layers have a 7×7 effective receptive field. So what have we gained by using, for instance, a stack of three 3×3 conv. layers instead of a single 7×7 layer? First, we incorporate three non-linear rectification layers instead of a single one, which makes the decision function more discriminative. Second, we decrease the number of parameters: assuming that both the input and the output of a three-layer 3×3 convolution stack has C channels, the stack is parametrised by $3*3^2*C^2 = 27*C^2$ weights; at the same time, a single 7×7 conv. layer would require $7^2*C^2 = 49*C^2$ parameters, i.e. 81% more. This can be seen as imposing a regularisation on the 7×7 conv. filters, forcing them to have a decomposition through the 3×3 filters (with non-linearity injected in between).

The incorporation of 1×1 conv. layers (configuration C, Table 1) is a way to increase the non-linearity of the decision function without affecting the receptive fields of the conv. layers. Even though in our case the 1×1 convolution is essentially a linear projection onto the space of the same dimensionality (the number of input and output channels is the same), an additional non-linearity is introduced by the rectification function. It should be noted that 1×1 conv. layers have

recently been utilised in the “Network in Network” architecture of Lin et al. (2014).

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as “conv receptive field size - number of channels”. The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A.A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

The incorporation of 1×1 conv. layers (configuration C, Table 1) is a way to increase the non-linearity of the decision function without affecting the receptive fields of the conv. layers. Even though in our case the 1×1 convolution is essentially a linear projection onto the space of the same dimensionality (the number of input and output channels is the same), an additional non-linearity is introduced by the rectification function. It should be noted that 1×1 conv. layers have recently been utilised in the “Network in Network” architecture of Lin et al. (2014). Small-size convolution filters have been previously used by Ciresan et al. (2011), but their nets are significantly less deep than ours, and they did not evaluate on the large-scale ILSVRC

dataset. Goodfellow et al. (2014) applied deep ConvNets (11 weight layers) to the task of street number recognition, and showed that the increased depth led to better performance. GoogLeNet (Szegedy et al., 2014), a top-performing entry of the ILSVRC-2014 classification task, was developed independently of our work, but is similar in that it is based on very deep ConvNets (22 weight layers) and small convolution filters (apart from 3×3 , they also use 1×1 and 5×5 convolutions). Their network topology is, however, more complex than ours, and the spatial resolution of the feature maps is reduced more aggressively in the first layers to decrease the amount of computation. As will be shown in Sect. 4.5, our model is outperforming that of Szegedy et al. (2014) in terms of the single-network classification accuracy.

3 CLASSIFICATION FRAMEWORK

In the previous section we presented the details of our network configurations. In this section, we describe the details of classification ConvNet training and evaluation.

3.1 TRAINING

The ConvNet training procedure generally follows Krizhevsky et al. (2012) (except for sampling the input crops from multi-scale training images, as explained later). Namely, the training is carried out by optimising the multinomial logistic regression objective using mini-batch gradient descent (based on back-propagation (LeCun et al., 1989)) with momentum. The batch size was set to 256, momentum to 0.9. The training was regularised by weight decay (the L2 penalty multiplier set to 5×10^{-4}) and dropout regularization for the first two fully-connected layers (dropout ratio set to 0.5). The learning rate was initially set to 10 -2 , and then decreased by a factor of 10 when the validation set accuracy stopped improving. In total, the learning rate was decreased 3 times, and the learning was stopped after 370K iterations (74 epochs). We conjecture that in spite of the larger number of parameters and the greater depth of our nets compared to (Krizhevsky et al., 2012), the nets required less epochs to converge due to (a) implicit regularization imposed by greater depth and smaller conv. filter sizes; (b) pre-initialisation of certain layers.

The initialisation of the network weights is important, since bad initialisation can stall learning due to the instability of gradient in deep nets. To circumvent this problem, we began with training the configuration A (Table 1), shallow enough to be trained with random initialisation. Then, when training deeper architectures, we initialised the first four convolutional layers and the last three fully-connected layers with the layers of net A (the intermediate layers were initialised randomly). We did not decrease the learning rate for the pre-initialised layers, allowing them to change during learning. For random initialisation (where applicable), we sampled the weights from a normal distribution with the zero mean and 10^{-2} variance. The biases were initialised with zero. It is worth noting that after the paper submission we found that it is possible to initialise the weights without pre-training by using the random initialisation procedure of Glorot & Bengio (2010).

To obtain the fixed-size 224 \times 224 ConvNet input images, they were randomly cropped from rescaled training images (one crop per image per SGD iteration). To further augment the training set, the crops underwent random horizontal flipping and random RGB colour shift (Krizhevsky et al., 2012). Training image rescaling is explained below.

Training image size. Let S be the smallest side of an isotropically-rescaled training image, from

which the ConvNet input is cropped (we also refer to S as the training scale). While the crop size is fixed to 224×224 , in principle S can take on any value not less than 224: for $S = 224$ the crop will capture whole-image statistics, completely spanning the smallest side of a training image; for $S \gg 224$ the crop will correspond to a small part of the image, containing a small object or an object part.

We consider two approaches for setting the training scale S . The first is to fix S , which corresponds to single-scale training (note that image content within the sampled crops can still represent multi-scale image statistics). In our experiments, we evaluated models trained at two fixed scales: $S = 256$ (which has been widely used in the prior art (Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014)) and $S = 384$. Given a ConvNet configuration, we first trained the network using $S = 256$. To speed-up training of the $S = 384$ network, it was initialised with the weights pre-trained with $S = 256$, and we used a smaller initial learning rate of 10^{-3} . The second approach to setting S is multi-scale training, where each training image is individually rescaled by randomly sampling S from a certain range $[S_{MIN}, S_{MAX}]$ (we used $S_{MIN} = 256$ and $S_{MAX} = 512$). Since objects in images can be of different size, it is beneficial to take this into account during training. This can also be seen as training set augmentation by scale jittering, where a single model is trained to recognise objects over a wide range of scales. For speed reasons, we trained multi-scale models by fine-tuning all layers of a single-scale model with the same configuration, pre-trained with fixed $S = 384$.

3.2 TESTING

At test time, given a trained ConvNet and an input image, it is classified in the following way. First, it is isotropically rescaled to a pre-defined smallest image side, denoted as Q (we also refer to it as the test scale). We note that Q is not necessarily equal to the training scale S (as we will show in Sect. 4, using several values of Q for each S leads to improved performance). Then, the network is applied densely over the rescaled test image in a way similar to (Sermanet et al., 2014). Namely, the fully-connected layers are first converted to convolutional layers (the first FC layer to a 7×7 conv. layer, the last two FC layers to 1×1 conv. layers). The resulting fully-convolutional net is then applied to the whole (uncropped) image. The result is a class score map with the number of channels equal to the number of classes, and a variable spatial resolution, dependent on the input image size. Finally, to obtain a fixed-size vector of class scores for the image, the class score map is spatially averaged (sum-pooled). We also augment the test set by horizontal flipping of the images; the soft-max class posteriors of the original and flipped images are averaged to obtain the final scores for the image.

Since the fully-convolutional network is applied over the whole image, there is no need to sample multiple crops at test time (Krizhevsky et al., 2012), which is less efficient as it requires network re-computation for each crop. At the same time, using a large set of crops, as done by Szegedy et al. (2014), can lead to improved accuracy, as it results in a finer sampling of the input image compared to the fully-convolutional net. Also, multi-crop evaluation is complementary to dense evaluation due to different convolution boundary conditions: when applying a ConvNet to a crop, the convolved feature maps are padded with zeros, while in the case of dense evaluation the padding for the same crop naturally comes from the neighbouring parts of an image (due to both the convolutions and spatial pooling), which substantially increases the overall network receptive field, so more context is captured. While we believe that in practice the increased computation time of multiple crops does not justify the potential gains in accuracy, for reference we also evaluate our networks using 50 crops per scale (5×5 regular grid with 2 flips), for a total of 150 crops over 3 scales, which is comparable to 144 crops over 4 scales used by Szegedy et al. (2014).

3.3 IMPLEMENTATION

Our implementation is derived from the publicly available C++ Caffe toolbox (Jia, 2013) (branched out in December 2013), but contains a number of significant modifications, allowing us to perform training and evaluation on multiple GPUs installed in a single system, as well as train and evaluation full-size (uncropped) images at multiple scales (as described above). Multi-GPU training exploits data parallelism, and is carried out by splitting each batch of training images into several GPU batches, processed in parallel on each GPU. After the GPU batch gradients are computed, they are averaged to obtain the gradient of the full batch. Gradient computation is synchronous across the GPUs, so the result is exactly the same as when training on a single GPU. While more sophisticated methods of speeding up ConvNet training have been recently proposed (Krizhevsky, 2014), which employ model and data parallelism for different layers of the net, we have found that our conceptually much simpler scheme already provides a speedup of 3.75 times on an off-the-shelf 4-GPU system, as compared to using a single GPU. On a system equipped with four NVIDIA Titan Black GPUs, training a single net took 2–3 weeks depending on the architecture.

4 CLASSIFICATION EXPERIMENTS

Dataset. In this section, we present the image classification results achieved by the described ConvNet architectures on the ILSVRC-2012 dataset (which was used for ILSVRC 2012–2014 challenges). The dataset includes images of 1000 classes, and is split into three sets: training (1.3M images), validation (50K images), and testing (100K images with held-out class labels). The classification performance is evaluated using two measures: the top-1 and top-5 error. The former is a multi-class classification error, i.e. the proportion of incorrectly classified images; the latter is the main evaluation criterion used in ILSVRC, and is computed as the proportion of images such that the ground-truth category is outside the top-5 predicted categories.

For the majority of experiments, we used the validation set as the test set. Certain experiments were also carried out on the test set and submitted to the official ILSVRC server as a “VGG” team entry to the ILSVRC-2014 competition (Russakovsky et al., 2014).

4.1 SINGLE SCALE EVALUATION

We begin with evaluating the performance of individual ConvNet models at a single scale with the layer configurations described in Sect. 2.2. The test image size was set as follows: $Q = S$ for fixed S , and $Q = 0.5(S_{min} + S_{max})$ for jittered $S \in [S_{min}, S_{max}]$. The results of are shown in Table 3.

First, we note that using local response normalisation (A-LRN network) does not improve on the model A without any normalisation layers. We thus do not employ normalisation in the deeper architectures (B–E).

Second, we observe that the classification error decreases with the increased ConvNet depth: from 11 layers in A to 19 layers in E. Notably, in spite of the same depth, the configuration C (which contains three 1×1 conv. layers), performs worse than the configuration D, which uses 3×3 conv. layers throughout the network. This indicates that while the additional non-linearity does help (C is better than B), it is also important to capture spatial context by using conv. filters with non-trivial receptive fields (D is better than C). The error rate of our architecture saturates when the depth reaches 19 layers, but even deeper models might be beneficial for larger datasets. We also compared the net B with a shallow net with five 5×5 conv. layers, which was derived from B by replacing each pair of 3×3 conv. layers with a single 5×5 conv. layer (which has the same receptive field as explained in Sect. 2.3). The top-1 error of the shallow net was measured to be 7% higher than that of B (on a center crop), which confirms that a deep net with small

filters outperforms a shallow net with larger filters.

Finally, scale jittering at training time ($S \in [256; 512]$) leads to significantly better results than training on images with fixed smallest side ($s = 256$ or $s = 384$), even though a single scale is used at test time. This confirms that training set augmentation by scale jittering is indeed helpful for capturing multi-scale image statistics.

Table 3: ConvNet performance at a single test scale.

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256:512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256:512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256:512]	384	25.5	8.0

4.2 MULTI-SCALE EVALUATION

Having evaluated the ConvNet models at a single scale, we now assess the effect of scale jittering at test time. It consists of running a model over several rescaled versions of a test image (corresponding to different values of Q), followed by averaging the resulting class posteriors.

Considering that a large discrepancy between training and testing scales leads to a drop in performance, the models trained with fixed S were evaluated over three test image sizes, close to the training one: $Q = \{S - 32, S, S + 32\}$. At the same time, scale jittering at training time allows the network to be applied to a wider range of scales at test time, so the model trained with variable $S \in [S_{\min}; S_{\max}]$ was evaluated over a larger range of sizes

$$Q = \{S_{\min}, 0.5(S_{\min} + S_{\max}), S_{\max}\}.$$

The results, presented in Table 4, indicate that scale jittering at test time leads to better performance (as compared to evaluating the same model at a single scale, shown in Table 3). As before, the deepest configurations (D and E) perform the best, and scale jittering is better than training with a fixed smallest side S. Our best single-network performance on the validation set is 24.8%/7.5% top-1/top-5 error (highlighted in bold in Table 4). On the test set, the configuration E achieves 7.3% top-5 error.

Table 4: ConvNet performance at multiple test scales.

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
C	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	24.8	7.5
D	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	24.8	7.5
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	24.8	7.5

4.3 MULTI-CROP EVALUATION

In Table 5 we compare dense ConvNet evaluation with multi-crop evaluation (see Sect. 3.2 for details). We also assess the complementarity of the two evaluation techniques by averaging their soft-max outputs. As can be seen, using multiple crops performs slightly better than dense evaluation, and the two approaches are indeed complementary, as their combination outperforms each of them. As noted above, we hypothesize that this is due to a different treatment of convolution boundary conditions.

Table 5: ConvNet evaluation techniques comparison. In all experiments the training scale S was sampled

from [256;512], and three test scales Q were considered: {256, 384, 512}.

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
		dense	24.8
D	multi-crop	24.6	7.5
	multi-crop & dense	24.4	7.2
	dense	24.8	7.5
E	multi-crop	24.6	7.4
	multi-crop & dense	24.4	7.1
	dense	24.8	7.5

4.4 CONVNET FUSION

Up until now, we evaluated the performance of individual ConvNet models. In this part of the experiments, we combine the outputs of several models by averaging their soft-max class posteriors. This improves the performance due to complementarity of the models, and was used in the top ILSVRC submissions in 2012 (Krizhevsky et al., 2012) and 2013 (Zeiler & Fergus, 2013; Sermanet et al., 2014).

The results are shown in Table 6. By the time of ILSVRC submission we had only trained the single-scale networks, as well as a multi-scale model D (by fine-tuning only the fully-connected layers rather than all layers). The resulting ensemble of 7 networks has 7.3% ILSVRC test error. After the submission, we considered an ensemble of only two best-performing multi-scale models (configurations D and E), which reduced the test error to 7.0% using dense evaluation and 6.8% using combined dense and multi-crop evaluation. For reference, our best-performing single model

achieves 7.1% error (model E, Table 5).

Table 6: Multiple ConvNet fusion results.

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	23.7	6.8	6.8

4.5 COMPARISON WITH THE STATE OF THE ART

Finally, we compare our results with the state of the art in Table 7. In the classification task of ILSVRC-2014 challenge (Russakovsky et al., 2014), our “VGG” team secured the 2nd place with 7.3% test error using an ensemble of 7 models. After the submission, we decreased the error rate to 6.8% using an ensemble of 2 models.

As can be seen from Table 7, our very deep ConvNets significantly outperform the previous generation of models, which achieved the best results in the ILSVRC-2012 and ILSVRC-2013 competitions. Our result is also competitive with respect to the classification task winner (GoogLeNet with 6.7% error) and substantially outperforms the ILSVRC-2013 winning submission Clarifai, which achieved 11.2% with outside training data and 11.7% without it. This is remarkable, considering that our best result is achieved by combining just two models--significantly less than used in most ILSVRC submissions. In terms of the single-net performance, our architecture achieves the best result (7.0% test error), outperforming a single GoogLeNet by 0.9%. Notably, we did not depart from the classical ConvNet architecture of LeCun et al. (1989), but improved it by substantially increasing the depth.

Table 7: Comparison with the state of the art in ILSVRC classification. Our method is denoted as “VGG”.

Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	23.7	6.8	6.8
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-		7.9
GoogLeNet (Szegedy et al., 2014) (7 nets)	-		6.7
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

5 CONCLUSION

In this work we evaluated very deep convolutional networks (up to 19 weight layers) for large-scale image classification. It was demonstrated that the representation depth is beneficial for the classification accuracy, and that state-of-the-art performance on the ImageNet challenge dataset can be achieved using a conventional ConvNet architecture (LeCun et al., 1989; Krizhevsky et al., 2012) with substantially increased depth. In the appendix, we also show that our models generalise well to a wide range of tasks and datasets, matching or outperforming more complex recognition pipelines built around less deep image representations. Our results yet again confirm the importance of depth in visual representations.

6 REFERENCES

- Bell, S., Upchurch, P., Snavely, N., and Bala, K. Material recognition in the wild with the materials in context database. CoRR, abs/1412.0623, 2014.
- Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In Proc. BMVC., 2014.
- Cimpoi, M., Maji, S., and Vedaldi, A. Deep convolutional filter banks for texture recognition and segmentation. CoRR, abs/1411.6836, 2014.
- Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In IJCAI, pp. 1237–1242, 2011.
- Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., and Ng, A. Y. Large scale distributed deep networks. In NIPS, pp. 1232–1240, 2012.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proc. CVPR, 2009.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. CoRR, abs/1310.1531, 2013.
- Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. The Pascal visual object classes challenge: A retrospective. IJCV, 111(1):98–136, 2015.
- Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In IEEE CVPR Workshop of Generative Model Based Vision, 2004.
- Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524v5, 2014. Published in Proc. CVPR, 2014.
- Gkioxari, G., Girshick, R., and Malik, J. Actions and attributes from wholes and parts. CoRR, abs/1412.2604, 2014.
- Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proc. AISTATS, volume 9, pp. 249–256, 2010.
- Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In Proc. ICLR, 2014.
- Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
- He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR, abs/1406.4729v2, 2014.
- Hoai, M. Regularized max pooling for image categorization. In Proc. BMVC., 2014.
- Howard, A. G. Some improvements on deep convolutional neural network based image classification. In Proc. ICLR, 2014.

- Jia, Y. Caffe: An open source convolutional architecture for fast feature embedding.
<http://caffe.berkeleyvision.org/>, 2013.
- Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. CoRR, abs/1412.2306, 2014.
- Kiros, R., Salakhutdinov, R., and Zemel, R. S. Unifying visual-semantic embeddings with multimodal neural language models. CoRR, abs/1411.2539, 2014.
- Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. CoRR, abs/1404.5997, 2014.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In NIPS, pp. 1106–1114, 2012.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551, 1989.
- Lin, M., Chen, Q., and Yan, S. Network in network. In Proc. ICLR, 2014.
- Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.
- Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In Proc. CVPR, 2014.
- Perronnin, F., Sánchez, J., and Mensink, T. Improving the Fisher kernel for large-scale image classification. In Proc. ECCV, 2010.
- Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. CNN Features off-the-shelf: an Astounding Baseline for Recognition. CoRR, abs/1403.6382, 2014.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet large scale visual recognition challenge. CoRR, abs/1409.0575, 2014.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In Proc. ICLR, 2014.
- Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. CoRR, abs/1406.2199, 2014. Published in Proc. NIPS, 2014.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.
- Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., and Yan, S. CNN: Single-label to multi-label. CoRR, abs/1406.5726, 2014.
- Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. CoRR, abs/1311.2901, 2013. Published in Proc. ECCV, 2014.

ACKNOWLEDGEMENTS

This work was supported by ERC grant VisRec no. 228180. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

附录 2

中文译文

大型图片识别的超深层卷积神经网络

Karen Simonyan* & Andrew Zisserman*

牛津大学，工程科学系，视觉识别组

{Karen,az}@robots.ox.ac.uk

陈怡凡译

[摘要]本文研究了在大规模图片识别环境中卷积网络深度对其识别精度的影响。我们的主要贡献是使用一个非常小的(3*3)卷积过滤器结构对增加深度的网络进行了全面的评估，表明了通过将深度加至16-19个权重层后可以对现有技术配置进行显著的改进。这些发现是我们在2014年ImageNet挑战赛提交结果的基础，在这个比赛中我们的团队分别获得了定位项目的第一名和分类项目的第二名。我们还展示了在我们的成果对于其他数据集也普遍使用，并且实现了具有领先水平的结果。我们已经公开了两个性能最好的卷积神经网络模型，以便在计算机视觉中深度视觉表征的更进一步的研究。

1.引言

卷积神经网络近年来在大规模图像和视频识别方面取得了巨大的成功(Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014; Simonyan & Zisserman, 2014), 这得益于大规模的公共图像数据库, 如 ImageNet(Deng et al., 2009),还有高性能计算系统, 如 GPU (图形处理器) 或大规模分布式集群(Dean et al., 2012)。尤其是 ImageNet 大规模视觉识别挑战赛(ILSVRC) (Russakovsky et al., 2014)在深度视觉识别体系的发展中发挥出了重要作用, 而且它也已经成为了几代大规模图像分类系统的试验平台, 从高维浅特征编码(Perronnin et al., 2010)(ILSVRC-2011 的优胜者)到深度卷积神经网络(Krizhevsky et al., 2012) (ILSVRC-2012 的优胜者)。

随着卷积神经网络在计算机视觉领域更多的以一种商品的形式出现, 为了达到更好的准确度, 人们对卷积神经网络的原有结构进行了一些改进。例如, 对 ILSVRC-2013 的最佳上传结果(Zeiler & Fergus, 2013; Sermanet et al., 2014)使用了更小的接收窗口尺寸和更小的第一层卷积层的步长。另一项改进是在整个图像还有多个尺度上密集地对网络进行训练和测试(Sermanet et al., 2014; Howard, 2014)。在本论文中, 我们重点解决了卷积神经网络体系结构设计的另一个重要方面——深度。为此, 我们固定了体系结构的其他参数, 并通过增加更多的卷积层来稳定地增加网络的深度, 这是可行的, 因为在所有层中都使用了非常小的(3×3)卷积过滤器。

因此, 我们提出了更加准确的卷积神经网络等的结构, 它不仅能够达到 ILSVRC 分类和定位任务的最高准确度, 而且也适用于其他图像识别数据集, 即使只是应用于一个相对简单的流水线的一部分(例如, 用线性支持向量机对深度特征分类而且没有微调), 它们也能够获得优异的表现。为了便于进一步的研究, 我们发布了我们两种最好的模型。

论文的其余部分组织如下。在第二部分, 我们描述了我们的卷积神经网络结构。对图像分类训练和评估的方法的详细介绍在第三部分, 并且在第四部分对国内外 ILSVRC 分类任务的结构进行了比较。第五部分总结了全文。为了完整性起见, 我们还在附录 A 中描述和评估了我们的 ILSVRC-2014 目标定位系统, 并在附录 B 中讨论了对其他数据集中非常深层的特性的概括。附录 C 包含了关键的论文修订的记录。

2 卷积神经网络的结构

秉承公平的原则，在度量增加卷积神经网络的深度所带来的改进时，我们所有的卷积神经网络层结构都是使用相同的原则设计的，灵感来自 Ciresan et al. (2011); Krizhevsky et al. (2012)。在本节中，我们首先描述了我们的卷积神经网络结构的通用布局(2.1 节)。然后详细说明评估中所使用的具体配置方案(2.2 节)。.我们的设计选择方案会在之后进行讨论并且在 2.3 节中与现有技术进行了比较。

2.1 架构体系

在训练期间，我们的卷积神经网络的输入是一个固定大小的 224×224 RGB 图像。我们所做的唯一预处理是对每个像素减去基于训练集计算出的平均 RGB 值。该图像通过一个系列的卷积层(conv.)传递，在其中，我们使用非常小的感受野的过滤器: 3×3 (这是用来捕捉左/右，上/下，中心方位的最小的过滤器)。在其中一种配置中，我们还使用了 1×1 的卷积过滤器，这可以被看作是输入通道的线性变换(接着是非线性)。卷积步长被固定为 1 个像素，卷积层输入空间的填充要满足卷积之后空间分辨率被保留，比如，在 3×3 的卷积层填充 1 个像素。空间池化是通过五个最大池化层来实现的，在其之后是一些常规的卷积层。(不是所有的卷积层之后都是最大采样)。最大采样是使用一个 2×2 像素的窗口执行，步长为 2。

一系列卷积层(在不同的结构中具有不同的深度)之后是三个全连接(FC)层:前两层有 4096 个通道，第三层执行 1000 种 ILSVRC 分类，因此包含 1000 个通道(每个代表一种分类)。最后一层是 softmax 层。在所有网络中，完全连接的层的结构都是相同的。

所有隐藏层都用了非线性校正(ReLU (Krizhevsky et al., 2012))，并对校正结果进行了分析，得到了各隐层的校正结果。我们注意到我们的网络(除了一个)没有包含局部响应规范化(LRN)的标准化(Krizhevsky et al., 2012): 这将在第四部分中展示，这种规范化不会提高在 ILSVRC 数据集上的性能，但会增加内存消耗和计算时间。如果可以适用，LRN 层的参数是(Krizhevsky et al., 2012)给出的参数。

2.2 配置

本文中评估的卷积神经网络配置在表 1 中的每一列被列出了。在下文中，我们将用层的名称(A-E)来称呼这些网络。所有配置都遵循 2.1 节中介绍的通用设计，并且只是深度有所不同:从 11 层权重层的网络 A(8 层卷积层和 3 层全连接层)到 19 层权重层的网络 E(16 层卷积层和 3 层全连接层)。卷积层的宽度(指的是通道数量)是相当小的，从第一层只有 64 个开始，然后在每个最大池化层之后增加 2 倍，直到达到 512 个。

在表 2 中，我们给出了每个配置的参数数量。尽管有一个较大的深度，但是在我们的网络中权重的数量并不比在一个有着更大的卷积层宽度和感受野的更浅的网络中的权重的数量大多(在 (Sermanet et al., 2014)有 1.44 亿个权重)。

2.3 讨论

我们的卷积神经网络配置与在 ILSVRC-2012(Krizhevsky et al., 2012)和 ILSVRC-2013(Zeiler & Fergus, 2013; Sermanet et al., 2014)竞赛中表现最好的参赛作品使用的配置完全不同。在第一部分的卷积层中我们没有使用相对较大的感受野(比如在(Krizhevsky et al., 2012) 11×11 的大小以及步长为 4, 或者在 (Zeiler & Fergus, 2013; Sermanet et al., 2014) 中 7×7 的大小以及步长为 2),, 而是在整个网络中使用了非常小的 3×3 的感受野, 这些感受野随着每个像素的输入而被卷积(步长为 1)。显而易见的是, 一个由两个 3×3 大小的卷积层组成的栈 (其中没有空间池化) 有着一个大小为 5×5 的有效感受野; 3 个这样的层有一个 7×7 大小的有效感受野。那么, 例如一个 3×3 的卷积层的栈, 通过使用它而不是单一的 7×7 大小的卷积层, 我们能得到什么? 首先, 采用三个非线性校正层代替单个校正层, 会使决策函数更具有判别性。其次, 我们减少了参数的数量:假设一个三层 3×3 卷积栈的输入和输出具有 C 个通道, 这个栈的由 $3 * 3^2 * C^2 = 27 * C^2$ 个权重构成; 同时, 一个单一 7×7 卷积层会需要 $7^2 * C^2 = 49 * C^2$ 个参数, 即多出 81%。这可以被看作是对 7×7 卷积过滤器的规范化, 使他们通过 3×3 过滤器得到分解(把非线性注入其间)。

表 1:卷积神经网络结构(在列中显示)。随着更多的层被加入(添加的层用粗体显示), 结构的深度从左(A)到右(E)增加。卷积层参数表示为"卷积层<感受野大小>-<通道数>"。为了简洁, 我们不展示 ReLU 激活函数。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

表 2: 参数数量 (单位为百万)

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

1×1 卷积层(结构 C,表 1)是一种增加决策函数的非线性但不影响卷积层的感受野的方法。即使在我们的例子中 1×1 卷积实质上是在同一维度空间上的线性投影(输入通道和输出通道的数目相同), 一种附加的非线性应由校正函数引入。应该注意的是 1×1 卷积层最近已被用于 Lin et al. (2014)的"网络中的网络"的架构。

小尺寸卷积过滤器以前曾被 Ciresan et al. (2011)使用, 但他们的网络明显没我们的深, 并且他们没有在大规模的 ILSVRC 数据集上评估。Goodfellow et al. (2014)将 11 个权重层的深度卷积神经网络应用于街道号码识别任务中, 结果表明, 增加深度可以获得更好的识别性能。GoogLeNet(Szegedy et al., 2014)是 ILSVRC-2014 大赛分类任务中表现最好的一个参赛作品, 独立于我们的作品开发的, 但类似的是, 它也是基于非常深的卷积神经网络(22

个权重层)和小型卷积过滤器(除了 3×3 大小，他们也使用 1×1 和 5×5 的卷积层)。然而，他们的网络拓扑结构比我们的更复杂，并且特征映射的空间分辨率在第一层被更加猛烈地降低以减少计算量。如 4.5 节所示，我们的模型在单网络分类准确率方面要优于 Szegedy et al. (2014)。

3 分类框架

在前面的章节中，我们介绍了网络结构的细节。在这一部分中，我们详细地描述了分类卷积神经网络的训练和评估。

3.1 训练

卷积神经网络的训练过程通常遵循 Krizhevsky et al. (2012) 的步骤(除了从多尺度训练图像中抽取输入的裁剪后的图片，后面将会解释)。也就是说，训练是通过使用带动量的小批量梯度下降法优化多项逻辑回归目标 (基于(LeCun et al., 1989)的反向传播)。批尺寸被设置为 256，动量设置为 0.9。训练通过权重衰减(L2 惩罚乘数设置为 5×10^{-4})和前两个全连接层的丢弃 (dropout) 正规化(丢弃率设置为 0.5)来调整。学习率最初设置为 10^{-2} ，当验证集的准确率停止提高时，学习率以 10 倍的速率下降。总的来说，学习率降低了 3 次，学习在 370K 次迭代(74 个 epochs)后停止。我们推测，尽管与(Krizhevsky et al., 2012)的网络相比，我们的网络参数数量和深度更多更大，但由于以下原因，网络为了收敛所需的 epoch 更少：
(a)更深和更小的过滤器尺寸引起隐式正则化;(b)某些层的预初始化。

网络权值的初始化很重要，因为不好的初始化会由于深度网络中梯度的不稳定而导致学习失败。为了避免这个问题，我们从结构 A(表 1)开始训练，足够的浅来达到通过随机初始化进行训练。然后，当训练更深层的架构时，我们初始化第一个 4 层卷积层和最后三个全连接的层与网络 A 的层(中间的层是随机初始化的)。我们没有降低预初始化层的学习率，并允许他们在学习过程中改变。对于随机初始化(如果适用)，我们从一个带有零均值和 10^{-2} 方差的正态分布采样权重。这些偏差都初始化为零。值得注意的是，在提交论文之后，我们发现可以使用 Glorot & Bengio (2010)的随机的初始化过程来初始化权重，并不需要预训练。

为了获得固定大小的 224×224 大小的卷积神经网络输入图片，这些图像从归一化的训练图像中随机裁剪(每个图像在每次 SGD 迭代裁剪一次)。为了进一步增强训练集，裁剪图片经历了随机水平翻转和随机 RGB 颜色变化(Krizhevsky et al., 2012)。训练图像的归一化说明如下。

训练图片尺寸。设 S 是等轴归一化的训练图片的最小边，从该图片裁出卷积神经网络输入(我们也称 S 为训练尺度)。虽然裁剪大小固定为 224×224 ，原则上 S 可以取任何不小于 224 的值:对于 $S = 224$ ，裁剪步骤将考虑整个图片的统计数据，完全扩展到训练图像的最小边；对于 $>>224$ 的 S ，裁剪这一步将对应于图片的一小部分，包括一个小物体或一个物体的部分。

我们考虑了两种设置训练量 S 的方法。第一种是固定 S ，它对应于单尺度训练(注意，

采样裁剪中的图片内容仍然可以代表多尺度图片统计)。在我们的实验中,我们评估了在两个固定尺度上训练的模型: $S = 256$ (在现有技术中被广泛使用(Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014))和 $S = 384$ 。给定一个卷积神经网络结构,我们首先使用 $S = 256$ 对网络进行训练。为了提高 $S = 384$ 网络的训练速度,它采用 $S = 256$ 的预训练的权重进行初始化,我们使用的初始学习率较小,为 10^{-3} 。

第二种设置 S 的方法是多尺度训练,每个训练图片在一定的范围内 $[S_{MIN}, S_{MAX}]$ (我们使用 $S_{MIN} = 256$ 和 $S_{MAX} = 512$)通过随机抽样 S 分别归一化。因为图片中的物体可以是不同大小的,所以在训练中考虑这一点是有益的。这也可以看作是通过尺度抖动来进行训练集增强,其中一个单一的模型经过训练能够识别不同尺度下的物体。出于速度方面的考虑,我们使用固定的 $S = 384$ 的预训练以及相同的结构来微调单尺度模型的所有层来训练多尺度模型。

3.2 测试

在测试时,给定一个训练好的卷积神经网络和一个输入图片,按以下方式对其进行分类。首先,它被重新定义为一个预先定义的最小图片边,表示为 Q (我们也称之为测试尺度)。我们注意到 Q 不一定等于训练尺度 S (我们将在第四部分中展示,对每个 S 使用多个 Q 值可以产生性能的提高)。然后,以类似(Sermanet et al., 2014)的方式在归一化的测试图像上密集地应用在网络中。也就是说,首先将全连接层转化为卷积层(第一个全连接层转化为 7×7 卷积层,最后两个全连接层转化为 1×1 卷积层)。然后,由此产生的完全卷积网络应用于整个(未裁剪)图片。结果是一个类评分图,其中通道数等于分类的数量,并且空间分辨率是可变的,取决于输入图像的大小。最后,对类得分图进行空间平均(和池化),得到一个固定大小的图片的类别得分的向量。我们也会通过图像的水平翻转来增强测试集;对原始图像和翻转图像的 softmax 类后验进行平均以获得图像的最终得分。

由于全卷积网络应用于整个图像,因此不需要在测试时对多种裁剪进行采样(Krizhevsky et al., 2012),而这种采样效率较低,因为它需要对每种裁剪进行网络的重新计算。与此同时,使用大量的裁剪图片,如 Szegedy et al. (2014)所做的,可以提高准确性,因为它的结果相比于完全卷积网络来说,输入图片有着更精细的采样。此外,由于不同的卷积边界条件,多裁剪图片评估与密集评估是互补的:当将卷积神经网络应用于裁剪图片时,卷积特征映射被零填充,而在密集评估的情况下,同一裁剪图片的填充自然来自于图片的邻近部分(由于卷积和空间池化),这大大增加了整个网络的感受野,因此接收到了更多的内容。虽然我们相信在实践中多种裁剪图片的计算时间增加并不能证明潜在的精确性增加是合理的,但作为参考,我们也评估我们的网络通过每尺度使用 50 次裁剪 (5×5 规则网格, 2 次翻转),一共在 3 个尺度上有 150 次裁剪,这相当于 Szegedy et al. (2014) 的在 4 个尺度上的 144 次裁剪。

3.3 实施细节

我们的实现来源于公开的 C++ Caffe 工具箱(Jia, 2013) (2013 年 12 月被扩展), 但是包含了一些重要的改进, 允许我们在安装于单个系统中的多个 GPU 上进行训练和评估, 以及在全尺寸(未裁剪)图像上执行训练和评估(如上所述)。多 GPU 训练利用数据并行性, 并通过将每批训练图像分割成若干 GPU 批次, 在每个 GPU 上并行处理来实现。在计算 GPU 批梯度后, 对它们进行平均, 以获得完整的批梯度。梯度计算在 GPU 之间是同步的, 所以和在一个 GPU 上训练的结果是完全一样的。

虽然最近提出了一些更为复杂的加快卷积神经网络训练的方法(Krizhevsky, 2014),, 这些方法对网络的不同层使用模型和数据并行性, 但是我们发现我们在概念上更为简单的方案已经在现成的 4-GPU 系统上提供了 3.75 倍的加速程度, 而不是使用单一的 GPU。在一个配备了四个英伟达泰坦黑色 GPU 的系统上, 根据不同的架构, 训练一个网络需要 2-3 个工作周期。

4 分类实验

数据集。这一部分中，我们介绍了在 ILSVRC-2012 数据集(用于 ILSVRC2012-2014 挑战赛)上所描述的卷积神经网络结构实现的图片分类结果。该数据集包括 1000 个分类的图片，并被分成三组:训练集(130 万张图片)，验证集(5 万张图片)和测试集(留有类标签的 10 万张图片)。采用 TOP-1 和 TOP-5 错误率这两种测量方法对分类性能进行了评估。前者是一个多类别的分类错误率，即在正确被分类的图片的比例；后者是主要在 ILSVRC 使用的评估标准，并且用于计算图片真实类别在前 5 个预测类别之外的图片比例。

对于大多数实验，我们使用验证集作为测试集。某些实验甚至是在测试集上进行的，并作为"VGG"团队参加 ILSVRC-2014 竞赛的竞赛作品提交给了官方 ILSVRC 服务器。

4.1 单一尺度评估

我们从评估单个卷积神经网络模型在单一尺度上的性能开始，使用 2.2 节中描述的网络层结构。测试图片的尺寸设定为:固定 $Q = S$, $Q = 0.5(SMIN + SMAX)$, 当抖动 $s \in [SMIN, SMAX]$ 。结果如表所示。

首先，当没有任何归一化层的情况下，我们注意到使用局部响应归一化(A-LRN 网络)对模型 A 没有改善。因此，我们在更深层的结构(B-E)中不使用归一化。

其次，我们观察到随着卷积神经网络深度的增加，分类错误率减小:从 A 的 11 层到 E 的 19 层。值得注意的是，除了深度相同，结构 C(包含 3 个 1×1 卷积层)表现比结构 D 差，D 使用的全是 3×3 卷积层。这表明，虽然附加的非线性确实有帮助(C 优于 B)，但通过使用带非平凡感受野的卷积过滤器获得空间上下文也很重要(D 优于 C)。当深度达到 19 层时，我们的架构错误率就会饱和，但是更深的模型对于更大的数据集来说可能更有用。我们还比较了网络 B 和有着 5 个 5×5 卷积层的浅层网络的差异，这个网络来源于 B 但把每对 3×3 卷积层都换成了一个单一的 5×5 卷积层(与 2.3 节解释的感受野相同)。测量结果表明，浅层网络的 TOP-1 错误率比 B (在中心裁剪图片中) 高 7%，证实了有着小过滤器深层网络性能优于有着大过滤器的浅层网络。

最后，训练时的尺度抖动($S \in [256; 512]$)比固定最小边($s = 256$ 或 $s = 384$)的图片训练效果明显要好，即使在测试时使用的是单一尺度。这证实了基于尺度抖动的训练集增强对于多尺度图像统计数据的获取确实是有帮助的。

表 3: 卷积神经网络在单个测试范围内的性能

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	25.5	8.0

4.2 多尺度评估

在单一尺度上评估卷积神经网络模型后，我们现在评估测试时尺度抖动的影响。它包括在一张测试图像的几个归一化版本上运行模型（对应于不同的 Q 值），然后对所得到的类别后验进行平均，用固定的 S 训练的模型在三个测试图片尺度上进行了评估，接近于训练一次： $Q = \{S - 32, S, S + 32\}$ 。同时，训练时的尺度抖动允许网络在测试时应用于更广的尺度范围，所以用变量 $S \in [S_{\min}; S_{\max}]$ 训练的模型在更大尺寸范围 $Q = \{S_{\min}, 0.5(S_{\min} + S_{\max}), S_{\max}\}$ 上进行评估。

表 4 中给出的结果表明，测试时的尺度抖动得出了更好的性能（与在单一尺度上相同模型的评估相比，如表 3 所示）。如前所述，最深的结构（D 和 E）表现最佳，并且尺度抖动优于使用固定最小边 S 的训练。我们在验证集上的最佳单网络性能为 24.8%/7.5% top-1/top-5 的错误率（在表 4 中用粗体突出显示）。在测试集上，配置 E 实现了 7.3% top-5 的错误率。

表 4: 在多个测试尺度上的卷积神经网络性能

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	24.8	7.5
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	24.8	7.5

4.3 多裁剪图片评估

在表 5 中，我们将稠密卷积神经网络评估与多裁剪图片评估进行比较（细节参见 3.2 节）。我们还通过平均其 soft-max 输出来评估两种评估技术的互补性。如我们所见，使

用多裁剪图像表现比密集评估略好，而且这两种方法确实是互补的，因为它们的组合优于其中的每一种。如上所述，我们假设这是由于卷积边界条件的不同处理。

表 5：卷积神经网络评估技术比较。在所有的实验中训练尺度 S 从[256;512]采样，三个测试尺度 Q 被考虑为：{256, 384, 512}。

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	24.4	7.2
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	24.4	7.1

4.4 卷积网络融合

到目前为止，我们评估了卷积神经网络模型的性能。在这部分实验中，我们通过对 softmax 类别后验进行平均，结合了几种模型的输出。由于模型的互补性，这样提高了性能，并且在了 2012 年(Krizhevsky et al., 2012)和 2013 年(Zeiler & Fergus, 2013; Sermanet et al., 2014)ILSVRC 的最佳提交作品中使用。

结果如表 6 所示。在 ILSVRC 提交的时候，我们只训练了单一尺度网络，以及一个多尺度模型 D（仅在全连接层进行微调而不是所有层）。由此产生的 7 个网络组合具有 7.3% 的 ILSVRC 测试错误率。在提交之后，我们考虑了只有两个表现最好的多尺度模型（配置 D 和 E）的组合，它使用密集评估将测试错误率降低到 7.0%，使用密集评估和多裁剪图像评估将测试错误率降低到 6.8%。作为参考，我们表现最佳的单一模型达到 7.1% 的错误率（模型 E，表 5）。

表 6：多个卷积网络融合结果

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	23.7	6.8	6.8

4.5 与最新技术比较

最后，我们在表 7 中与领先水平的技术比较我们的结果。在 ILSVRC-2014 挑战赛的分类任务(Russakovsky et al., 2014)，中，我们的“VGG”团队获得了第二名，使用 7 个模型的组合取得了 7.3% 测试错误率。提交后，我们使用 2 个模型的组合将错误

率降低到 6.8%。

从表 7 可以看出，我们非常深的卷积神经网络显著优于前一代模型，在 ILSVRC-2012 和 ILSVRC-2013 竞赛中都取得了最好的结果。我们的结果对于分类任务获胜者（GoogLeNet 具有 6.7% 的错误率）也具有竞争力，并且大大优于 ILSVRC-2013 获胜者 Clarifai 的提交作品，其使用外部训练数据取得了 11.2% 的错误率，没有外部数据则为 11.7%。这是非常卓越的，考虑到我们最好的结果是仅通过组合两个模型实现的——明显少于大多数 ILSVRC 提交作品。在单一网络性能方面，我们的架构取得了最好结果（7.0% 的测试错误率），超过单个 GoogLeNet 0.9% 的百分比。值得注意的是，我们并没有偏离 LeCun (1989) 等人经典的卷积神经网络架构，但通过大幅增加深度改善了它。

表 7：在 ILSVRC 分类中与最新技术比较。我们的方法表示为“VGG”。报告的结果没有使用外部数据。

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	23.7	6.8	6.8
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-		7.9
GoogLeNet (Szegedy et al., 2014) (7 nets)	-		6.7
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

5 结论

在这项工作中，我们评估了用于大规模图片分类的超深度卷积网络（最多 19 个权重层）。已经证明，表达深度有利于分类精度，并且深度大大增加的传统卷积神经网络架构 (LeCun et al., 1989; Krizhevsky et al., 2012) 可以实现 ImageNet 挑战数据集上的最佳性能。在附录中，我们还显示了我们的模型很好地泛化到各种各样的任务和数据集上，可以匹敌或超越更复杂的识别流程，其构建围绕着不那么深的图片表示。我们的结果再次证实了深度在视觉表示中的重要性。

6 参考文献

- Bell, S., Upchurch, P., Snavely, N., and Bala, K. Material recognition in the wild with the materials in context database. CoRR, abs/1412.0623, 2014.
- Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In Proc. BMVC., 2014.
- Cimpoi, M., Maji, S., and Vedaldi, A. Deep convolutional filter banks for texture recognition and segmentation. CoRR, abs/1411.6836, 2014.
- Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In IJCAI, pp. 1237–1242, 2011.
- Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., and Ng, A. Y. Large scale distributed deep networks. In NIPS, pp. 1232–1240, 2012.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proc. CVPR, 2009.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. CoRR, abs/1310.1531, 2013.
- Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. The Pascal visual object classes challenge: A retrospective. IJCV, 111(1):98–136, 2015.
- Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In IEEE CVPR Workshop of Generative Model Based Vision, 2004.
- Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524v5, 2014. Published in Proc. CVPR, 2014.
- Gkioxari, G., Girshick, R., and Malik, J. Actions and attributes from wholes and parts. CoRR, abs/1412.2604, 2014.
- Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proc. AISTATS, volume 9, pp. 249–256, 2010.
- Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., and Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In Proc. ICLR, 2014.
- Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.
- He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR, abs/1406.4729v2, 2014.
- Hoai, M. Regularized max pooling for image categorization. In Proc. BMVC., 2014.
- Howard, A. G. Some improvements on deep convolutional neural network based image classification. In Proc. ICLR, 2014.

- Jia, Y. Caffe: An open source convolutional architecture for fast feature embedding.
<http://caffe.berkeleyvision.org/>, 2013.
- Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. CoRR, abs/1412.2306, 2014.
- Kiros, R., Salakhutdinov, R., and Zemel, R. S. Unifying visual-semantic embeddings with multimodal neural language models. CoRR, abs/1411.2539, 2014.
- Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. CoRR, abs/1404.5997, 2014.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In NIPS, pp. 1106–1114, 2012.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551, 1989.
- Lin, M., Chen, Q., and Yan, S. Network in network. In Proc. ICLR, 2014.
- Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.
- Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In Proc. CVPR, 2014.
- Perronnin, F., Sánchez, J., and Mensink, T. Improving the Fisher kernel for large-scale image classification. In Proc. ECCV, 2010.
- Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. CNN Features off-the-shelf: an Astounding Baseline for Recognition. CoRR, abs/1403.6382, 2014.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet large scale visual recognition challenge. CoRR, abs/1409.0575, 2014.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In Proc. ICLR, 2014.
- Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. CoRR, abs/1406.2199, 2014. Published in Proc. NIPS, 2014.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.
- Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., and Yan, S. CNN: Single-label to multi-label. CoRR, abs/1406.5726, 2014.
- Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. CoRR, abs/1311.2901, 2013. Published in Proc. ECCV, 2014.

致谢

这项工作得到 ERC 授权的 VisRec 编号 228180 的支持.我们非常感谢 NVIDIA 公司捐赠 GPU 为此研究使用。