密级：_____

浙江大学

硕 士 学 位 论 文

论文题目 基于卷积神经网络的视频分类检索

作者姓名 刘伟

指导教师 寿黎但教授

学科(专业) 计算机应用技术

所在学院 计算机科学与技术学院

提交日期 2017-1-5

A Dissertation Submitted to Zhejiang University for the Degree of Master of Engineering

TITLE: Video Classification and Retrieval Based on Convolutional Neural Networks

Author: Liu Wei

Supervisor: Prof. Shou Lidan

Subject: Computer Application Technology

College: Computer Science and Technology

Submitted Date: 2017-01-05

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：　　签字日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：　　导师签名：

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

学位论文作者毕业后去向：

工作单位：　　电话：

通讯地址：　　邮编

摘要

随着深度学习在图像识别领域的巨大成功，深度学习被应用到越来越多的领域，包括视频处理领域，文本处理领域以及音频处理领域。卷积神经网络是深度学习中非常重要的一类网络模型，它和传统神经网络的区别在于引入了卷积层、池化层。卷积神经网络在图像识别领域取得成功之后，研究人员开始逐步将其应用于视频分类任务中，并取得了分类效果上的提高，这充分说明了卷积神经网络在视频分类任务中起到的重要作用。

本文的研究内容在于实现一个通用且有效的分类检索模型。在图像研究领域有研究人员提出在卷积神经网络中引入哈希层来得到图像对应的 0-1 向量哈希码，使用这个哈希码来进行相似图片的查找，取得了不错的效果。本文的一个创新点在于将这种思路引入到了视频处理领域，验证了哈希层在这个应用场景中的有效性，继而提出通用且效果更好的分类检索模型；另外一个创新点在于采用了更为高效的检索算法来提高模型的可用性。

本文在调研了相关领域的工作之后，采用了基于 VGG-NET 神经网络模型的视频分类模型，在模型中引入哈希层来得到视频对应的 0-1 向量哈希码。在模型对视频数据分类功能方面，通过实验验证了分类模型的准确率；在得到哈希码后的模型检索部分，本文提出的改进的检索算法和朴素的依次比较算法相比在检索时间上有显著地改进。总的来说，本文实现的通用分类检索模型具有更好的分类准确率和更快的检索速度。

关键词： 卷积神经网络，视频分类，视频检索，哈希层，0-1 向量

Abstract

With the great success achieved by deep learning in image recognition research area, deep learning becomes more and more important in other research areas, including video classification, natural language processing, audio classification. One of the most important models used in deep learning is Convolutional Neural Network. Compared to the traditional neural network, deep neural network contains new layers and use new optimizer to train the model. After it's widely used in image recognition, the researchers begin to apply the Convolutional Neural Network to the video classification task. This paper also use the hash algorithms came up with by researcher in image retrieval research area.

One part of this paper is about video classification. The traditional video classification algorithm is based on the features extracted from key frames. But now, the deep learning is becoming a more powerful tool to solve this problem. The mainstream algorithm combines the Convolutional Neural Network and Recurrent Neural Network. The other part of this paper is about video retrieval. This paper use the algorithm which is used in image retrieval, and try to prove it is workable in video data also.

After analyzing the research work in related areas, we choose to use the video classification model based on VGG-NET which is one of the most successful Convolutional Neural Network models, and add a full-connected layer with Sigmoid function as activation function which is used to get binary hash code for each video. After we have got the binary hash code for each video, we come up with a new algorithm based on binary code and Trie to search for the similar videos. Compared to basic algorithms, it can improve the time used in similar videos searching effectively.

Keywords：CNN , Video classification , Video retrieval , Hash Layer , 0-1 feature vector

目录

图目录

表目录

第 1 章 绪论

1.1 课题背景

随着近年来数据量的剧增，以及数据处理能力的提高，机器学习算法在数据分析领域发挥着越来越重要的作用，其中的深度学习是近年来最为热门的研究领域[1]。深度学习是从传统神经网络演化而来的一种新的分类预测解决方案，本质上和传统的神经网络一样是一个用于分类或者预测的神经网络模型。不同的地方在于深度神经网络引入了很多改进[2]，针对不同的应用场景提供了很多不同的特定的模型，使得深度神经网络和过去的神经网络相比分类效果更好，模型训练更容易。随着数据量的增大、服务器性能的提高，深度神经网络已经成为了如今图像、文本、视频、音频研究领域的主流解决方案。

神经网络是源于生物学的一种机器学习模型，里面引入神经元的概念来模拟人大脑中的神经元。神经元接受输入，通过特定的权值和激活函数来得到对应的输出，进而作为下一个神经元的输入。神经网络出现以后一直没有得到重视的原因有以下几点：一个是当时的数据量不能满足神经网络训练的需要，数据量不够的情况下网络的训练会存在过拟合的问题。随着大数据时代的到来，神经网络所需要的数据能够得到保证。二是训练存在问题，之前的神经网络存在很多训练方面的问题，比如因为梯度发散导致模型训练失败。最近提出的 Dropout，Batch Normalization 层较好的解决了这类问题，使得深度神经网络能够成功训练。

深度学习最开始取得巨大成功是在图像识别领域[3]，使用了卷积神经网络的分类模型对 ImageNet 图像数据集的识别准确率比过去的模型有了很大提高。在图像领域取得成功之后，深度神经网络开始被应用到文本，音频，视频领域。在文本处理中，循环神经网络因为处理时序递推信息的优越性，被广泛应用。在视频领域中，基于单帧的卷积神经网络分类算法，扩展的处理多帧的卷积神经网络分类算法，以及结合了卷积神经网络和循环神经网络的视频分类算法都有很好的分类效果。

在之前的相关研究中，视频分类算法和图片分类算法类似[4]，需要把视频作为一帧一帧的图片来处理。在得到图片帧之后，对图片提取特征，利用词袋模型得到图片对应的一个固定长度的实数向量，利用支持向量机进行训练，训练之后得到的支持向量机分类器就可以对新的帧进行分类。卷积神经网络在图像分类领域取得更好的效果之后，研究人员用卷积神经网络取代过去基于支持向量机的图像分类算法，提出了基于单帧的视频分类算法。这个算法的缺点在于没有考虑到帧与帧之间的关联性。为了能够利用时间上的连续性，研究人员提出了三维卷积层，一个卷积的处理对象不再是一帧，而是连续的几帧，这样就可以将连续的帧作为输入进行处理，从而利用了帧与帧之间的连续性，预测效果进一步提高。

本论文中进一步扩展了用于视频分类的网络结构，引入了哈希层来实现视频哈希。在图像研究领域，有研究人员提出可以在图像分类模型中引入 Hash Layer[5]，扩展的模型的分类准确率和检索准确率都很高，且整个模型具有简单高效的优点。在本论文中将证明在视频分类的网络模型中同样可以引入 Hash Layer，且能够起到较好的视频哈希和视频检索的效果。

1.2 本文的工作与贡献

本文所做的工作有两点，第一是将卷积神经网络用于视频分类，在网络层次中引入一个全连接层作为哈希层，从哈希层中得到视频对应的哈希码，进而实现视频之间基于汉明距离的相似度比较，最终完成相似视频的检索。第二是有关相似视频的查找，为了更好的解决近邻查找的问题，论文提出了基于 Trie 树的相似哈希码检索算法来加速近邻的查找。同时因为神经网络本身是用于视频分类的，训练得到的网络是可以用于视频分类的。总的来说，论文最终得到了一个可以用于分类的深度神经网络，通过网络可以对视频进行分类，也可以对视频进行哈希。另外提出了基于 Trie 树的相似哈希码检索算法，提高了相似视频的检索速度。

具体的网络模型结构设计方面，在调研了已有的网络模型结构之后，采用经典分类模型 VGG-NET 作为该论文中的视频分类模型。为了能够顺利训练模型，在模型中引入了 Dropout 来防止过拟合，调小了学习速率来实现更为精细的模型

学习。此外，在网络层次中引入了 Hash Layer，这一层是整个网络结构的最后一层，所有的网络输出都注入到这一层。通过设置阈值将这一层的输出转换成 Binary Hashcode。最后一层是 Softmax 层，用于得到对不同类别进行分类的概率。整个模型训练的过程中需要做精确的参数调整，包括学习速率的调整，批量训练数据集大小的设置以及优化方法的设置。

论文从两个方面证明模型的有效性。一个是模型分类的准确性。本文首先会对整个数据集进行划分，分为训练集，验证集和测试集。在验证集上调参得到最好的模型之后，在测试集上进行结果的测试，得到的准确率作为模型分类准确率的评估。另一方面是模型检索的准确性和检索效率。检索的准确性用检索出来的视频的类别和当前视频真实类别一致的比例来衡量，比如检索最相似的 n 个视频，里面和当前视频真实类别一致的视频个数为 m，m / n 即为检索准确率，准确率越高说明检索的效果越好；检索的效率用找出最相似的 n 个视频的时间来评估，花费的时间越少检索的效率越高。此外实验基于公开数据集，实验的结果具有通用性和可靠性。

总的来说，本论文的工作包括以下几点；

1. 分析了图像中的分类方法和视频中的分类方法，基于图像中的经典分类模型实现本文中的视频分类模型
2. 将图像中的哈希策略引入视频分类模型，验证哈希层在视频数据上的可用性，测试检索的准确性
3. 提出了一个更为高效的相似视频检索策略，验证检索策略的准确性、可行性，测试检索策略的时间效率
4. 在公开数据集上做以上工作，验证算法的可行性

1.3 本文组织和结构

本文一共分为七章。第一章是绪论，简单介绍了论文涉及到的研究方向，包括当前的研究状况以及研究进展，简要说明了一下论文中的工作以及创新点，同时包括整篇论文的组织和结构。第二章对论文相关的工作做了一个综述，包括图

像中使用的分类检索算法和技术，视频中使用的分类检索算法和技术。第三章定义了论文希望解决的问题，分析了解决问题时的难点，以及论文预期实现的目标。第四章给出了基于卷积神经网络的视频分类技术，具体包括使用的 VGG-NET 网络结构；具体定义了哈希层；对近邻检索的相关技术做了一个综述，包括近似近邻检索以及 KD 树等检索技术，同时给出了本文提出的基于 Trie 树的相似视频检索算法。第五章提出了两个视频分类检索模型的应用场景，即摘要视频生成算法和推荐系统模块。第六章分析实验的结果，包括说明实验进行的环境，进行实验的数据集，和传统的视频分类算法的对比实验结果，以及利用哈希码做检索的准确率以及时间效率。第七章对整篇论文的工作做一个总结，总结论文中的工作以及创新点，分析论文中解决方案的优缺点，说明今后的研究方向。

第 2 章 相关工作

本章主要介绍和本文工作相关的研究以及发展情况。本章主要从两个方面进行分析，一个是视频分类检索算法，一个是传统的卷积神经网络以及层次扩展后的卷积神经网络。同时本章对卷积神经网络在视频分类上的应用做了一定的总结和介绍，分析卷积神经网络在视频分类检索上的应用以及发展历程。

2.1 视频分类检索

2.1.1 概述

视频分类和视频检索是视频处理中的两个常见任务，视频分类用于对新的视频的类别进行分类，视频检索用于检索给定视频的相似视频[6]。视频分类主要是基于视频的内容进行处理，通过对视频进行逐帧分析，提取帧对应的特征，用于视频分类；视频检索主要是通过提取视频对应的指纹，通过比较指纹的相似度来评估视频之间的相似度，进而检索相似的视频。

2.1.2 图像分类算法

图像分类算法主要分为两大块[7]，一种是基于各种经典特征的词袋模型，使用支持向量机或者其它的分类器进行分类器的训练。在进行分类操作的时候，首先得到图像的特征，然后根据词袋模型得到对应的向量，最后将向量作为分类器的输入，输出对于图像的分类结果。另外一种是基于卷积神经网络的分类模型，直接采用卷积神经网络进行图片的处理，将图片直接作为神经网络的输入，输出即为图片的分类结果。

如果采用基于特征的分类方法，主要会采用以下几种特征：GIST 特征，SIFT 特征以及 HOG 特征[8]。GIST 是图像前一种全局特征，旨在通过图像的光谱信息反映其整体布局，而 SIFT 则是图像的一种局部特征，主要是通过统计图像关键点邻域内的梯度方向信息来反应邻域内的局部结构。HOG 是一种局部特征，侧重于通过统计图像像素块的边缘梯度方向信息来刻画目标的形状结构。

采用卷积神经网络进行图片分类的算法[9]，流程和传统的基于特征的图像分

类算法有所不同。最大的不同之处在于无需对图像做特征提取的操作。在卷积神经网络中，输入是图像的所有像素，即 WIDTH x HEIGHT x CHANNEL 的像素矩阵，输出即为分类结果。

2.1.2.1 GIST 特征

GIST 是图像前一种全局特征，旨在通过图像的光谱信息反映其整体布局，是一种常用的图像特征。该特征对外景分类效果较好，室内场景效果较差，是图像前一种全局特征，忽略了物体或背景的细微纹理信息。

2.1.2.2 SIFT 特征

SIFT 是一种监测局部特征的算法，该算法通过求一张图片中的特征点机器有关尺度和朝向的描述子得到特征并进行图像特征点匹配，能够获得良好的效果。整个算法分为以下几个部分：

1. 构建尺度空间。这是一个初始化的操作，尺度空间理论目的是模拟图像数据的多尺度特征。

2. 找到关键点。即检测 DOG 尺度空间极值点。为了寻找尺度空间的极值点，每一个采样点要和它所有的相邻点比较，看其是否比它的图像域和尺度域的相邻点大或者小。

3. 除去不好的特征点，这一步本质上要去掉 Dog 局部曲率非常不对称的像素。通过拟合三维二次函数以精确确定关键点的位置和尺度，同时去除低对比度的关键点和不稳定的边缘响应点（因为 DoG 算子会产生较强的边缘响应），以增强匹配稳定性、提高抗噪声能力。

4. 给特征点赋值一个 128 维方向参数。上一步中确定了每幅图中的特征点，为每个特征点计算一个方向，依照这个方向做进一步的计算，利用关键点邻域像素的梯度方向分布特性为每个关键点指定方向参数，使算子具备旋转不变性。至此图像的关键点检测完毕，每个关键点有三个信心：位置、所处尺度、方向，由此可以确定一个 SIFT 特征区域。

5. 生成关键点的描述子，利用公式求得每个像素的梯度幅值与梯度方向，箭头方向代表该像素的梯度方向，箭头长度代表梯度模值，然后用高斯窗口对其

进行加权运算。

6. 根据 SIFT 特征进行匹配，生成了图片对应的描述子之后，就将两图中各个尺度的描述子进行匹配，匹配上所有的维度即可表示两个特征点成功匹配。

2.1.2.3 HOG 特征

方向梯度直方图特征是一种在计算机视觉和图像处理中用来进行物体监测的特征描述子。它通过计算和统计图像局部区域的梯度方向直方图来构成特征。HOG 特征结合支持向量机已经被广泛应用于图像识别中，尤其在行人检测中获得了极大的成功。

在一张图片中，局部目标的表象和形状能够被梯度或边缘的方向密度分布很好地描述。它的本质是梯度的统计信息，而梯度主要存在于边缘的地方。

HOG 特征提取算法的实现过程大致如下：

1. 灰度化，即将图像看做一个 x ， y ， z 的三维图像

2. 采用 Gamma 校正法对输入图像进行颜色空间的标准化，目的是调节图像的对比度，减低图像局部的阴影和光照变化所造成的影响，同时可以抑制噪音的干扰

3. 计算图像每个像素的梯度，包括大小和方向，主要是为了捕获轮廓信息，同时进一步弱化光照的干扰

4. 将图像划分成小单元，单元即为很小的一个像素矩阵

5. 统计每个单元的梯度直方图，即为不同梯度的个数，即可形成每个单元的描述子

6. 将每几个单元组成一个块，比如 3 * 3 个单元/块，一个块内所有的单元的特征描述子串联起来便得到该块的 HOG 特征描述子

7. 将图像内的所有块的 HOG 特征描述子串联起来就可以得到该图像的 HOG 特征描述子，这个就是最终的可供使用的特征向量

2.1.2.4 卷积神经网络分类算法

基于卷积神经网络的图像分类算法只需要将图像转化成统一的大小，作为模型的输入，进行模型的训练即可[10]。模型中包括卷积层，池化层，Dropout 层以

及最后的 softmax 层用于得到输入属于哪一类的概率，算法一般选择预测概率最大的作为分类的结果。

卷积神经网络主要用于识别、缩放以及其它形式扭曲不变性的的二维图形[11]。由于卷积神经网络的特征检测层通过训练数据进行学习，所以在使用卷积神经网络时，避免了显示的特征抽取，隐式地从训练数据中进行学习。由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相较于神经元彼此互连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性，其布局更接近于实际的生物神经网络，权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。

总的来说，卷积神经网络有以下特点[12]：输入图像和网络的拓扑结构能很好地吻合；特征提取和模式分类同时进行，并同时在训练中产生；权值共享可以减少网络的训练参数，使神经网络结构变得更简单，适应性更强。

图像分类领域一直在不断地发展，最近得到广泛应用的卷积神经网络更是进一步推动了图像分类领域的发展。在 ImageNet 等国际图像识别比赛中，不断涌现着新的网络模型和新的图像处理算法。图像处理和视频处理也是密不可分的两个领域，相信随着图像分类领域的发展，视频分类领域也会发生很大的改变。

2.1.3 图像检索算法

2.1.3.1 概述

图像检索算法指的是进行相似图片检索的算法，算法的具体应用场景是对一张图片快速找出数据库中所有相似的图片。针对这个问题研究人员提出了很多基于哈希的检索策略，如平均值 hash，感知 hash，差异 hash 算法等[13]。在对每张图片得到哈希值之后，可以通过查找相似的哈希值去查找对应的相似图片。在卷积神经网络研究领域，有研究人员将卷积神经网络和图片哈希结合起来提出了新的算法，取得了不错的结果。

2.1.3.2 平均 hash 算法

此算法是基于比较灰度图每个像素与平均值来实现的，最适用于缩略图，放大图搜索。整个算法的步骤如下：

1. 缩放图片：为了保留结构去掉细节，去除大小、横纵比的差异，把图片统一缩放到 8 * 8，共 64 个像素的图片

2. 转化为灰度图：把缩放后的图片转化成 256 阶的灰度图

3. 计算平均值：计算进行灰度处理后图片的所有像素点的平均值

4. 比较像素灰度值：遍历灰度图片每一个像素，如果大于平均值记录为 1，否则为 0

5. 得到信息指纹：组合 64 个 bit 位，顺序随意保持一致性即可

6. 对比指纹：计算两幅图片的指纹，计算汉明距离（从一个指纹到另一个指纹需要变几次），汉明距离越大则说明图片越不一致，反之，汉明距离越小则说明图片越相似，当距离为 0 时，说明完全相同。（通常认为距离 > 10 就是两张完全不同的图片）

2.1.3.3 感知 hash 算法

平均 hash 算法过于严格，不够精确，更适合搜索缩略图，为了获得更精确的结果可以选择感知 hash 算法，它采用 DCT（离散余弦变换）来降低频率。具体算法流程如下：

1. 缩小图片：32 * 32 是一个较好的大小，这样方便 DCT 计算

2. 转化为灰度图：把缩放后的图片转化为 256 阶的灰度图

3. 计算 DCT：DCT 把图片分离成分率的集合

4. 缩小 DCT：DCT 是 32 * 32，保留左上角的 8 * 8，这些代表的图片的最低频率

5. 计算平均值：计算缩小 DCT 后的所有像素点的平均值

6. 进一步减小 DCT：大于平均值记录为 1，反之记录为 0

7. 得到信息指纹：组合 64 个人信息位，顺序随意保持一致性即可

8. 对比指纹：计算两幅图片的指纹，计算汉明距离（从一个指纹到另一个指纹需要变几次），汉明距离越大则说明图片越不一致，反之，汉明距离越小

则说明图片越相似，当距离为 0 时，说明完全相同。（通常认为距离 ＞10 就是两张完全不同的图片）

2.1.3.4 差异 hash 算法

相比感知 hash，差异 hash 的速度要更快；相比平均 hash，差异 hash 在效率几乎相同的情况下效果要更好，它是基于渐变实现的。算法的执行流程如下：

1. 缩小图片：收缩到 9 * 8 的大小，一遍它有 72 的像素点

2. 转化为灰度图：把缩放后的图片转化为 256 阶的灰度图

3. 计算差异值：差异 hash 算法工作在相邻像素之间，这样每行 9 个像素之间产生了 8 个不同的差异，一共 8 行，则产生了 64 个差异值

4. 获得指纹：如果左边的像素比右边的更亮，则记录为 1，否则为 0

需要说明的是，这种指纹算法不仅可以应用于图片搜索，同样适用于其他多媒体形式。除此之外，图片搜索特征提取方法有很多，很多算法还有很多可以改进的地方，比如对于人物可以先进行人脸识别，再在面部区域进行局部的 hash，或者背景是纯色的可以先过滤剪裁等等，最后在搜索的结果中还可以根据颜色、风景、产品等进行过滤

2.1.3.5 基于卷积神经网络的 Hash 算法

最近研究人员提出，可以将图片分类和图片哈希结合起来，即在卷积神经网络网络模型中加入一层全连接层，将该层输出结果通过一个阈值得到 Binary Hashcode，这个哈希码即为这张图片的哈希结果。论文中讨论了将这个哈希码作为图片的哈希码进行相似图片检索的准确性，以及原本的模型的分类准确性。结果表明加入的全连接层不会影响模型分类的准确性，通过引入 Hash Layer 得到的哈希码可以作为很好的图片的指纹，指纹间的相似度可以表示图片之间的相似度。这样就可以利用指纹之间的汉明距离评估图片之间的相似度了。

整个算法的执行流程大致如下：

1. 在已有的经典卷积神经网络分类模型，如 VGG-NET 上加上冗余的一层作为哈希层。这一层是普通的全连接层，激活函数采用经典的 Sigmoid 函数。不采用 ReLU 的原因是 Sigmoid 函数能够得到一个(0 , 1)实数，这样就能够

通过一个阈值将实数向量转成 0-1 向量，相较于 ReLU 更为方便。

2. 利用训练数据训练模型，得到调参良好的模型

3. 模型在 Hash Layer 的输出是一个实数向量，通过设置阈值，将实数向量转成 0-1 向量。论文中采用的方案是如果超过阈值的即为 1，没有超过阈值的即为 0

4. 对于输入的图片，算法可以从 Hash Layer 得到对应的 Binary Hashcode，即图片的哈希码。通过查找相似的哈希码来查找相似的图片，进而完成相似图片的查找工作

2.1.4 视频分类算法

目前视频分类的主流解决思路是采用基于视频内容的处理和检索[14]，主要通过以下两种方式实现：

1. 部分区域固定特征监测方式。即通过逐帧的分析视频图像，定位图像中需要查询的相关特征，根据此特征的匹配程度，确定图形和视频是否属于某类视频。此类方法的代表性处理手段有：匹配标识，匹配服饰特征，匹配特定性人物。通常要实现此类方法，需要采用人脸识别、纹理识别、相似度计算等方法。该方法的缺点是由于过度依赖固定特征和固定区域，当视频内容发生变化时，难以有效区分。

2. 图像整体特征方法。即通过图像特征提取算法，获得图像整体系统特征，通过大量数据的训练，获得分类器模型，利用此分类器模型实现对后续的图像视频的分类识别。此类方法中采用的图像提取算法通常有：SIFT 算法、灰度共生矩阵法、傅里叶功率谱法等。该方式的缺点是由于采用的是固定特征提取算法，当视频中掺杂了干扰数据后，此类算法将会获得大量带噪音的特征，极大降低分类的效果。

基于深度学习的视频分类算法[15]大致有以下几种：

1. 基于单帧的识别方法。算法首先将视频进行截帧，然后基于图像粒度进行卷积神经网络的分类。这种分类算法存在一个很大的缺陷，因为一帧图片是整个

视频很小的一部分，如果这帧图片没有什么区分度，或者是和视频主题无关的图片，就会极大地影响分类的效果。此外如果针对单独的帧进行处理，就会无法利用帧之间的连续性。算法的流程如图 2.1 所示。

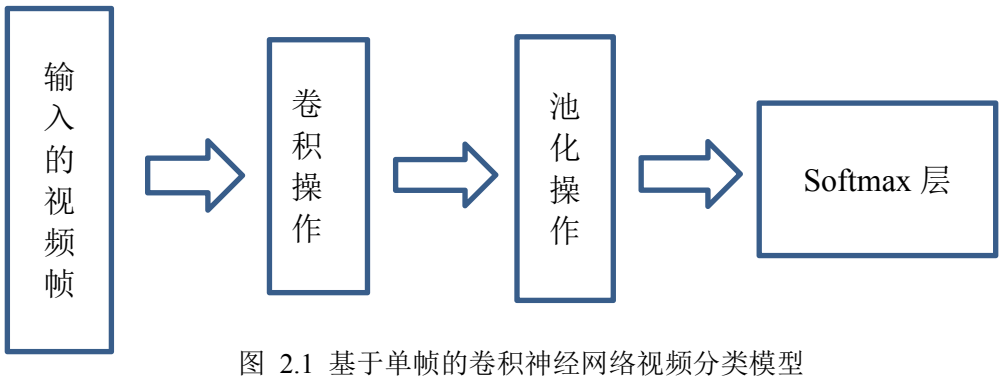

图 2.1 基于单帧的卷积神经网络视频分类模型

2. 因为单帧识别存在着很多的问题，有研究人员提出了一种卷积层的扩展，即将卷积操作从 3 维扩展到 4 维，第四维表示是连续的哪几帧。这样就可以利用连续帧之间的关联关系，更好的表示这个视频。结果也表明这样的解决方案要比单帧的识别方法更好。扩展的卷积层示意图如图 2.2 所示。

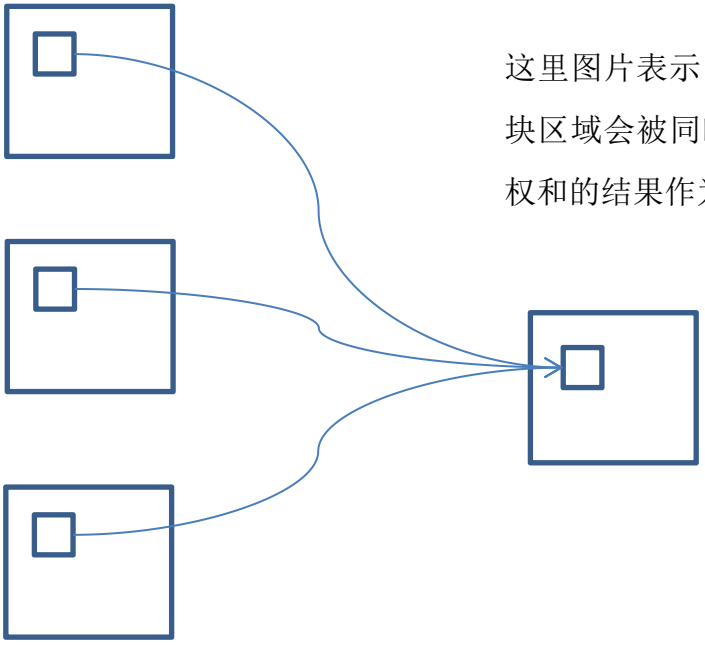

图 2.2 基于扩展的卷积神经网络模型的视频分类算法

3. 因为以上两种方法都有自己的优势，有研究人员提出将两种方法合二为

一，融合两个模型的结果，进而提出双路卷积神经网络的识别算法。简单来说，就是独立的训练两个网络，最后将神经元连接到同一个 Softmax 层，进行模型输出的融合。多路卷积神经网络分类算法如图 2.3 所示。

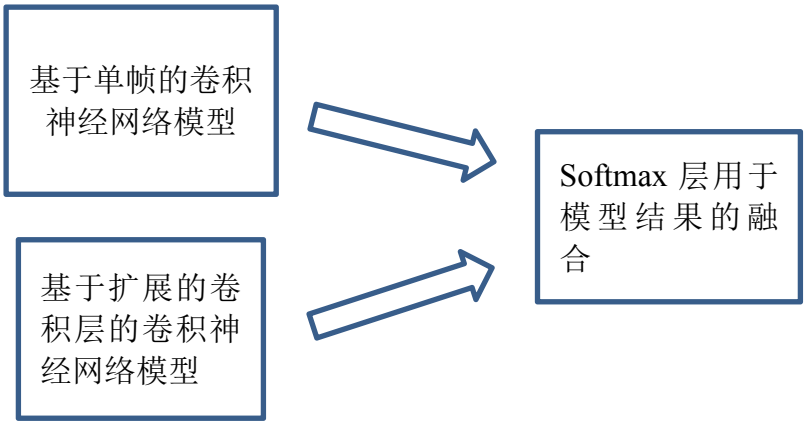

图 2.3 双路卷积神经网络分类算法

4. 近几年研究人员提出可以将循环神经网络引入视频分类之中[16]。算法基本思想是用循环神经网络对最后一层的激活在时间轴上进行整合。这里算法没有使用卷积神经网络全连接层后的最后特征进行融合，是因为全连接层后的高层特征进行池化已经丢失了控件特征在时间轴上的信息。网络中通过引入循环神经网络的记忆单元，可以有效地表达帧的先后顺序。

总的来说，视频分类算法是依赖于图像分类算法的。不管是基于图像特征进行视频分类的分类算法，还是使用卷积神经网络或者扩展的卷积神经网络进行视频分类的分类算法，都是基于图像分类算法实现的。

2.1.5 视频检索算法

视频检索算法指的是快速检索出相似视频的算法，其中包含两个关键问题，一个是相似性度量，而是快速检索方法。近年来，在视频特征提取和相似性度量方面研究者们提出了很多有意义的方法[17]，例如基于片段的视频摘要方法，通过次采样帧和层次累计聚类提取视频特征，用比例化最大权二分图匹配实现相似性度量；随机化视频特征提取算法，将特征投影到基于位置敏感哈希的直方图上，并采用基于 Kernel 的相似度度量方法；基于颜色矩的特征提取方法，基于二分图

匹配和图变换的由粗到细的检索方法，利用最大值匹配和次最相似匹配对检索结果进行过滤。

本文中做了一个新的尝试，在基于卷积神经网络的视频分类模型中引入哈希层实现视频哈希。结果表明这是一个有效的视频指纹生成策略，能够很好的应用在视频数据集中。

2.2 卷积神经网络

2.2.1 卷积神经网络概述

2.2.1.1 基本概念

卷积神经网络是引入了很多新特征的一种神经网络[18]，网络结构中包括传统神经网络的所有基本元素，同时相较于传统神经网络做出了很多的改进。卷积神经网络中引入了卷积层进行卷积操作，引入了池化操作进行特征的聚合统计，引入了 Dropout 层防止过拟合。最终的网络结构不再是神经元与神经元之间一一相连的关系，而是交错相连的关系。此外卷积神经网络在处理图片的时候把整张图片直接作为输入，避免了特征提取的过程。根据生物学上的研究表明[19]，人看东西的过程实际上就是不断总结不断聚合特征的过程，卷积神经网络就是在模拟这个过程，所以卷积神经网络的前几层输出实际上是这幅图片非常好的一个特征的表述。这是卷积神经网络的优势，可以从图片中提取出非常有效的特征。所以卷积神经网络不仅是分类器，也是特征提取器。

2.2.1.2 核心思想

除了引入了很多新的层次，如卷积层，池化层等等。卷积神经网络真正变得可用得益于几个重要思想的提出[20]，它们让参数众多，极其复杂的卷积神经网络的训练变得可行，同时能够更好的提取图片的特征。卷积操作的大致示意图如图 2.4 所示。

1. 局部感受。一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性较弱。因此每个神经元没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部

的信息综合起来就得到了全局的信息。

2. 参数共享。研究人员认为图像的一部分的统计特征与其他部分是一样的，这意味着在一部分学习的特征也能用在另一部分上。所以对于图像的所有位置，都能使用同样的学习特征。

3. 多卷积核。如果只采用一个卷积核进行卷积操作，特征的提取不是那么的充分。如果添加多个卷积核，就可以学习多种特征。每个卷积核都会将图像生成为另一幅图像，这样产生的不同图像可以看做是一张图片的不同通道。

4. 池化操作。在通过卷积获得特征之后，需要使用这些特征完成分类操作。理论上算法可以使用全部的特征做分类，但是这样计算量会过于庞大。池化聚合操作能够很好地减低特征的数目，同时保留特征的表示能力。

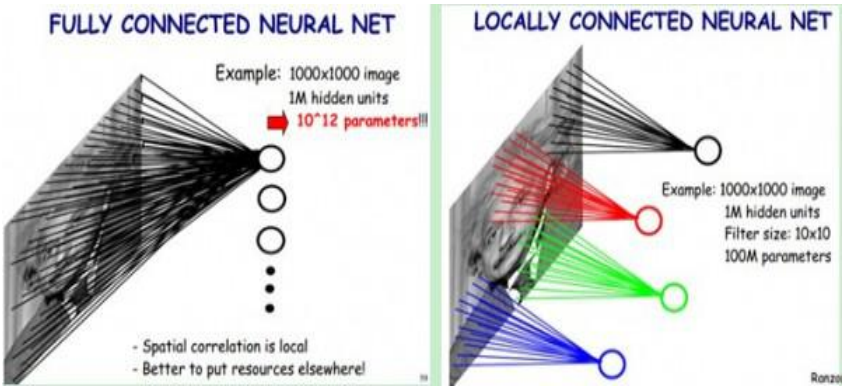

图 2.4 卷积操作示意图

2.2.1.3 发展历史

卷积神经网络是近年发展起来，并引起广泛重视的一种高效识别方法[21]。20 世纪 60 年代，Hubel 和 Wiesel 在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现其独特的网络结构可以有效地降低反馈神经网络的复杂性，继而提出了卷积神经网络。现在卷积神经网络已经成为众多科学领域的研究热点之一，特别是在模式分类领域，由于该网络避免了对图像的复杂前期预处理，可以直接输入原始图像，因而得到了广泛的应用。K.Fukushima 在 1980 年提出的新识别机是卷积神经网络的第一个实现网络。随后更多的科研工作者对该网络进行了改进，其中具有代表性的研究成果是 Alexander 和 Taylor 提出的“改进认知机”，该方法

综合了各种改进方法的优点并避免了耗时的误差反向传播。

2.2.2 卷积神经网络结构

卷积神经网络的结构和普通神经网络类似，包括很多的层次，每个层次里面包括很多的神经元。存在一个输入层，一个输出层，中间的所有层次都作为隐层。主要的不同之处在于卷积神经网络里面存在很多独有的层次，它们用于解决图像分类检索方面的问题。

引入的新层次主要包括以下几个：卷积层，池化层，Dropout 层，Normalization 层等等[22]。卷积层的作用是输入一个图片信息，输出转换后的图片信息，这个输出是图像某一方面特征的具体表示。池化层主要做的工作是对特征做一个总结。因为特征过多会加大训练的难度，如果能用更少的特征进行表示，同时不会降低模型的分类能力，就是更好的解决方案。Dropout 层和 Normalization 层都是为了避免过拟合，加速模型的收敛。

2.2.3 卷积神经网络扩展结构

在视频分类任务中，视频不同于单独的一张图片，是很多帧图片连起来的多媒体文件，所以视频文件中存在静态图片中不存在的帧与帧之间的连续关联性。为了利用好这个连续性，有研究人员提出扩展卷积神经网络中的经典层次[23]，比如卷积层，池化层。他们提出可以将这些层次的维度扩展一维用于利用帧与帧之间的连续关联性，即对图像的一块做卷积操作时，实际上结果是多个图片的同样区域的卷积结果的加权和。这样的卷积神经网络对视频分类的效果更好，特别是对帧与帧之间存在明显连续关系的视频效果更好。

2.3 卷积神经网络在视频分类检索领域的应用

2.3.1 视频分类检索应用概述

卷积神经网络在图像分类领域取得成功之后，有研究人员提出在视频分类中使用卷积神经网络。最简单的基于图片特征的视频分类算法是基于图片实现的，而采用卷积神经网络做图片分类是目前最好的算法，所以在视频分类算法中使用

卷积神经网络是必然的趋势。由于视频存在大量的帧，帧与帧之间存在关联性，直接使用卷积神经网络做视频分类已经不是更好的选择，扩展的卷积神经网络、将卷积神经网络和循环神经网络结合起来的分类算法[24]，都能够取得更好的效果。扩展的卷积神经网络能够取得更好的效果的原因是使用了连续帧之间的关系；而循环神经网络本身是用于处理时间序列任务的算法，视频本身具有时序性，所以使用循环神经网络能够更好的解决视频分类的问题。在将卷积神经网络哈希算法引入视频数据之后，可以实现相似视频检索。本文中给出两个视频分类检索算法的具体应用，一个是视频分类在视频摘要上的应用，一个是视频检索在推荐系统中的应用。

2.3.2 视频分类在视频摘要上的应用

视频摘要就是以自动或者半自动的方式，通过分析视频的结构和内容存在的时空冗余，从原始视频中提取有意义的片段，将它们以某种特点的方式重新组合成紧凑的、能够充分表现视频语义内容的浓缩视频。常用的视频摘要提取方式是静态视频摘要和动态视频摘要。

静态视频浓缩摘要通过描述原始视频中的每帧图像特征（如颜色、纹理、视觉显著性等），通过对相邻帧间的特征差异分析，抽取出原始视频的关键帧，对关键帧进行聚类，形成表达不同主题场景的视频片段，最后根据视频片段的信息进行组合，生成一段短的摘要视频。

动态视频浓缩摘要算法在得到视频帧之后，先对当前场景建立背景模型，然后快速根据视频照耀处理的特殊性，将原始视频份额外静态视频段、目标密集视频段、摘要基本段，同时为每个摘要基本段落生成其最佳背景模型。接着，基于背景建模，对运动目标进行监测、跟踪，提取其运动轨迹，通过运动轨迹表示该目标对象。然后对时空异步的多目标轨迹进行重新组合，去除视频的空间冗余，在重组的过程中应该考虑避免伪碰撞、保护原始相关性等原则，使重组的轨迹不丢失隐形信息。

以上两种算法都是计算机视觉相关研究领域提出的算法，具有专业性和功能

单一性。本文中使用的基于卷积神经网络的视频分类模型可以很简单的扩展到该应用场景，无需额外的工作量。算法的执行流程如下：

1. 对所有的视频提取一定数目的帧，进行基于单帧的卷积神经网络模型的训练，也就是图片分类模型的训练。一个视频的所有帧都是同样的标签。

2. 对一个给定的视频，对每一帧进行分类预测，得到该帧被分类成真实标签的概率。

3. 在得到一系列的概率之后，对概率从高到低进行排序，选择最高的那些帧作为摘要的帧。得到的帧组成的视频即为摘要视频。

算法本身的可靠性依赖于分类模型的分类准确性。模型分类一帧为某个标签的概率很大，说明这帧能够很好的表示该视频，作为浓缩的摘要帧也是合理的。具体的视频摘要例子如下：

首先在视频数据集上训练基于卷积神经网络的视频分类模型，这里无需引入 Hash Layer，因为该应用场景只需要分类模型。在得到视频分类模型之后，对视频的每一帧进行帧的分类，得到分类为正确标签的概率。在得到所有的帧对应的概率之后，选择概率最大的那些帧作为摘要视频中的那些帧，将这些帧合并得到的视频就是最终的摘要视频。原始视频中的帧如图 2.5，摘要视频的帧如图 2.6。

图 2.5 原始视频中的部分帧

图 2.6 使用算法剪辑后的视频中的部分帧

视频摘要获得算法中有一个参数可以设置，即选择分类概率最高的 k%的帧作为新的视频的帧。本文测试数据中的比例为 0.7，即选择预测概率为 top70%的帧拼接成剪辑后的摘要视频。视频的时间从 3s 变成了 2s，内容上最大的差异在于中间人具体的行走部分少了很多帧。不过这个摘要结果不影响对整个视频内容的判断，得到的摘要视频具有一定的概括性。

总的来说，该算法相较于其它算法的缺点在于缺少理论上的支持，完全没有考虑连续帧的特征差异，没有考虑视频的语义信息；优点在于算法非常简单，是基于卷积神经网络的视频分类模型的一个小的应用，无需额外的工作。

2.3.3 视频检索在推荐系统上的应用

推荐系统是现在大型系统中不可缺少的一个重要部分，主要用于基于历史记录或者实时操作进行用户可能感兴趣的东西的推荐。该领域主要使用的算法是协同过滤，协同过滤主要侧重于两个方面，一个是基于 Item 的协同过滤，即通过用户对不同的 Item 的评分来评测 item 之间的相似性，基于 item 之间的相似性做出推荐；另一个是基于 User 的协同过滤，即通过不同的用户对 item 的评分来评测用户之间的相似性，基于用户之间的相似性做出推荐。总的来说，都是基于评分的推荐操作。

本文中所做的相似视频检索是基于视频内容的相似检索，而内容相似和 tag 相似存在一定的关联关系，同一用户可能会对相似的视频都感兴趣。基于以上分

析，本文提出可以利用视频检索算法实现一个基于视频内容的推荐算法，每次推荐相似的视频给用户。此外可以将协同过滤和基于内容的相似视频检索结合起来，推荐那些相似且协同过滤认为可能感兴趣的视频，这样就可以将评分和视频内容结合起来，推荐系统能够起到更好的效果。推荐系统的宏观模型结构如图 2.7 所示。

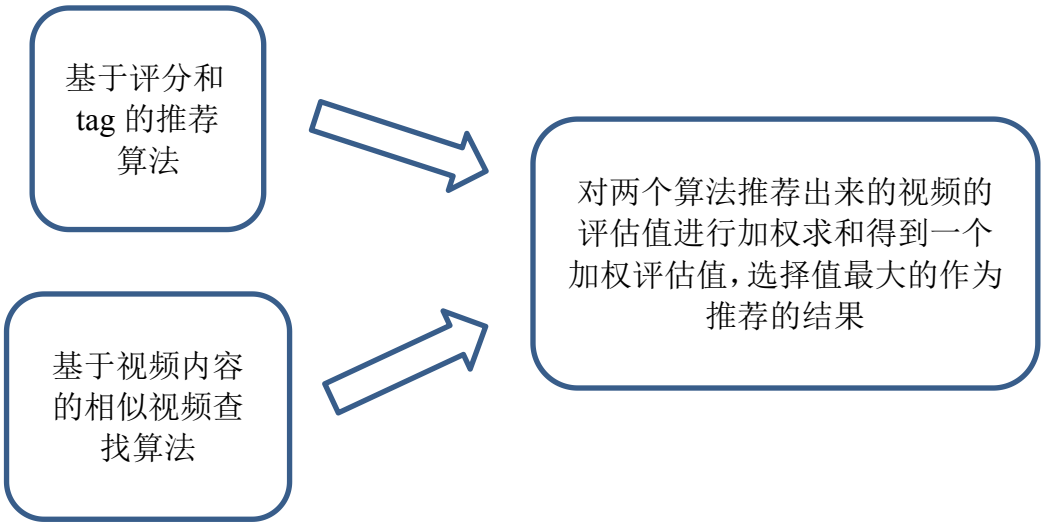

图 2.7 整合的视频推荐策略

具体的两个推荐结果的融合方式如下：

对一个视频的推荐值= $w1\times$(基于tag的推荐算法得到的值)+ $w2\times$（视频和当前视频的相似概率）

2.4 本章小结

本章主要介绍了常见的图像分类检索算法，视频分类检索算法。图像的分类任务主要通过经典特征和卷积神经网络进行处理；相似图片的检索主要通过哈希得到指纹，进而比较汉明距离来进行处理；视频的分类任务可以通过视频帧的特征进行处理，目前主流的方法是通过卷积神经网络和循环神经网络结合起来进行处理；视频的检索可以基于关键帧进行处理，也可以在得到指纹后进行相似性的度量，再进行相似视频的检索。此外简单说明了卷积神经网络的模型结构，网络层次以及卷积神经网络用于视频分类的方式，即基于单帧的视频分类算法以及扩

展的卷积神经网络分类算法。在第三部分给出了基于卷积神经网络的视频分类检索算法的实际应用场景，包括视频分类在视频摘要上的应用以及视频检索在推荐系统架构上的应用。本章简单提到了本文中所涉及到的工作。

第 3 章 问题描述

本章主要介绍论文中需要解决的问题，即视频分类分类检索，以及本文中所做的一些创新。首先介绍论文中会出现的常用术语和符号，主要涉及图像处理相关的术语，机器学习相关的术语，以及近邻检索相关的术语。此外具体定义了需要解决的问题，即如何做视频分类，如何生成视频的指纹，以及如何对相似视频进行检索。最后分析了问题解决过程中存在的难点，本文计划如何解决这些难点，以及预期得到什么样的结果。

3.1 常用术语与符号

这里简单介绍一下论文中会涉及到的一些常用术语以及对应的简单描述，避免论文中使用的一些名词造成误解或者导致难以理解。计算机视觉相关的名词如表格 3.1，深度学习算法相关的名词如表格 3.2，机器学习算法相关名词如表格 3.3，近邻检索领域的名词如表格 3.4。

表格 3.1 计算机视觉名词

名称	描述
SIFT	尺度不变换特征，用于物体辨识、机器人地图感知与导航等领域
HOG	方向梯度直方图，用来计算局部图像梯度的方向信息的统计值
GIST	通过图像光谱信息反映其整体布局

表格 3.2 深度学习算法相关名词

名称	描述
CNN	卷积神经网络
Convolution Layer	卷积层，如果加上了帧的连续处理就是扩展了维度的卷积层
Pooling Layer	实现池化操作的层次
Hash Layer	CNN 中加入的一个全连接层，用于得到图像或者视频对应的哈希向量，进而得到图像或者视频对应的 Binary Hashcode
LSTM	长短期人工记忆神经网络，适合处理和预测时间序列中间隔和延迟较长的事件

表格 3.3 机器学习算法相关名词

svm	支持向量机，用于分类的机器学习算法，目的在于最大化决策边界。能够应对线性可分和线性不可分的情况，是在深度学习出现之前非常强有力的一个模型
ensemble	模型融合操作，融合策略有很多，最常用的应该是加权和的方式来进行模型结果的融合
knn	通过 k 个近邻数据点的类别来预测某个数据点的类别，一般取 k 个数据点类别的众数进行预测

表格 3.4 近邻检索领域的名词

名称	描述
汉明距离	对于两个 0-1 向量，汉明距离即为两个向量中不同的二进制位的个数
Trie	用于支持前缀查找的前缀树，因为存储的是 Binarycode，所以这里对应的是一棵二叉树。在本文中会被用于查找一个可能的 Hashcode 是否存在，如果前缀不存在，提前终止查找
KD-Tree	对整个样本数据空间进行划分的查找树，是一棵二叉树，每次在一个维度上按照一个值进行划分
ANN	近似近邻查找，查找近似的近邻，不去查找精确值的算法。用于快速查找近邻
LSH	简单来说，就是满足以下性质的一类哈希策略。通过将原始数据空间中的两个数据点通过相同的映射或投影变换后，这两个数据点在新的数据空间中依然相邻的概率很大。因为保留了相邻的关系，所以可以被用于近邻的查找

3.2 问题定义

视频分类与相似视频检索指的是对一个新的视频的类别进行分类，以及给一个新的视频，检索出相似的视频这两个任务。之前对两个问题都有一些独立的解决方法，对于视频分类可以采用关键帧图像特征提取来做，也就是用图像分类的算法来做；对于视频检索可以提取视频的指纹，然后比较相似度进行相似视频的查找。本文希望做的事情是将这两个任务在一个模型中进行解决，即得到一个通用的模型，在做视频分类的同时得到视频对应的哈希码。分类和检索的效果都可以用准确率来衡量，分类准确率即在测试集上分类的准确率，检索准确率即针对一个视频检索出一定的视频，其中的和当前视频的标签相同的视频的个数占总视

频个数的比例。

本文中所做的工作大致如下，在普通的视频分类卷积神经网络模型上加入一个全连接层，作为哈希码的生成网络层次。然后按照普通的模型训练策略进行模型的训练。这样这个模型就同时具备了分类的能力和生成哈希码的能力。在进行检索操作时，首先生成视频对应的 Binary Hashcode，然后找出汉明距离最近的那些视频作为相似视频检索的结果。这里的检索操作相较于依次判断所有视频和当前视频的汉明距离，本文采用了一种更为高效的策略来查找相似视频，即将所有的视频的 Binary Hashcode 用 Trie 组织起来，提供更快的哈希码是否存在的判断，同时维护哈希码到视频文件的映射。

3.3 实现难点

本文的难点有以下几部分：首先模型的训练是一个很大的挑战，因为视频文件相较于图像文件更为复杂，进行分类的模型也就更为复杂，模型的复杂会导致模型中参数的剧增，从而加大模型的训练难度，合理设置网络结构层次是很重要的一个环节；其次 Hash layer 的设置，包括神经元个数的选择，二分阈值的选择都是影响最终结果的关键点；最后，如何提高检索的效率是非常重要的一点，在 Deep learning of Binary Hash Codes for Fast Image Retrieval 这篇论文中，采用的检索方法是最为朴素的依次比较，这样的检索方法是比较低效的一种做法，同时也没有利用得到的结果是 0-1 向量这一特殊的特征。本文提出了一种新的检索算法来提高检索的效率。

总的来说，难点在于以下几部分：

1. 基于卷积神经网络的视频分类检索模型的成功训练
2. 哈希层的设置，具体包括神经元个数的选择以及二分阈值的选择
3. 提高相似哈希码的检索效率，采用 Trie 来组织所有视频的哈希码。一方面可以把哈希码相同的视频存储到一个叶子结点之中，另一方面可以利用前缀查找优化对于一个哈希码是否存在的判断。

3.4 预期目标

本论文的预期目标是基于卷积神经网络实现一个集视频分类检索于一体的通用模型，能够同时保证较好的分类准确率和相似视频检索准确率。此外改进相似视频检索策略，提高相似视频检索的效率，保证视频检索的准确率不会因为速度的提高而下降。

3.5 本章小结

本章主要明确了整篇论文需要做的工作，需要完成的目标。论文的主要工作在于需要实现一个基于卷积神经网络的视频分类模型，需要在模型中引入哈希的层次，需要改进已有的近邻检索策略。本文的预期目标是能够成功训练基于卷积神经网络的视频分类模型，并能够对每个视频得到对应的哈希码，改进的近邻检索策略能够在保证准确率的同时提高检索速度。在下一章中，论文主要描述进行视频分类时使用的算法，包括基于卷积神经网络的视频分类算法；进行近邻检索时使用的算法，包括依次顺序比较算法，基于 KD-Tree 的近邻查找算法，基于 ANN 的近邻查找算法以及本文提出的基于 Trie 树的近邻查找算法。

第 4 章 基于卷积神经网络的视频分类检索

本章内容是论文的重点，重点描述如何实现本文中所提到的所有的算法，包括具体的实现策略以及算法原理。在第三章中本文明确需要解决的问题在这章都会给出具体的解决方案。

4.1 算法概述

本文中使用的算法主要基于卷积神经网络。输入的是视频的帧，即图片。中间会经过很多的卷积操作，池化操作，以及 Dropout 操作。最后还有一个小的全连接层，激活函数不采用一般深度学习中习惯使用的 RELU[25]，而是使用 Sigmoid 激活函数，这是为了该层的结果能够是一个(0 , 1)的浮点数，可以通过一个阈值来将浮点数向量转成 0-1 向量，这样就可以得到视频对应的 0-1 向量。

模型中间涉及到的卷积操作，池化操作，Dropout 操作的层次结构设计理念源于经典的卷积神经网络网络模型 VGG-Net[26]。该网络结构是 ImageNet 比赛中取得了很好的成绩的一个经典卷积神经网络网络模型。主要特点是，前面使用了大量的卷积操作提取图像丰富的特征，并辅以一些最大池化层进行特征的聚合，同时也是为了避免特征过多导致难以训练。在特征提取结束后平摊所有的特征，将其与全连接层相连，总结特征，最后连接到 Softmax 层进行图像的分类操作。模型的计算量因为深层的特征提取相对较大，但是能够很好地适应不同的数据集，具有较好的通用性。

在得到视频对应的 0-1 向量之后，需要应用检索策略来针对一个视频快速检索出这个视频的相似视频。最基本的做法是遍历所有的视频，比较视频哈希码之间的汉明距离，找出最小的汉明距离的那些视频，这些视频即为最相似的那些视频。本文提出了一个更好的检索算法来代替这种遍历计算汉明距离的算法，极大地提高了检索的效率，让相似视频检索变得更为可用。

4.2 卷积神经网络结构

本文中采用的网络结构源于经典的 VGG-Net 网络结构。VGG-Net 本身源于一个更经典的网络结构 AlexNet[27]，网络结构如图 4.1 所示。这是在 ImageNet 比赛中创下惊人成绩的第一个知名卷积神经网络结构，首次引入了包括数据增强，Dropout，Relu 等概念。下面简单介绍一下这些引入的新概念。

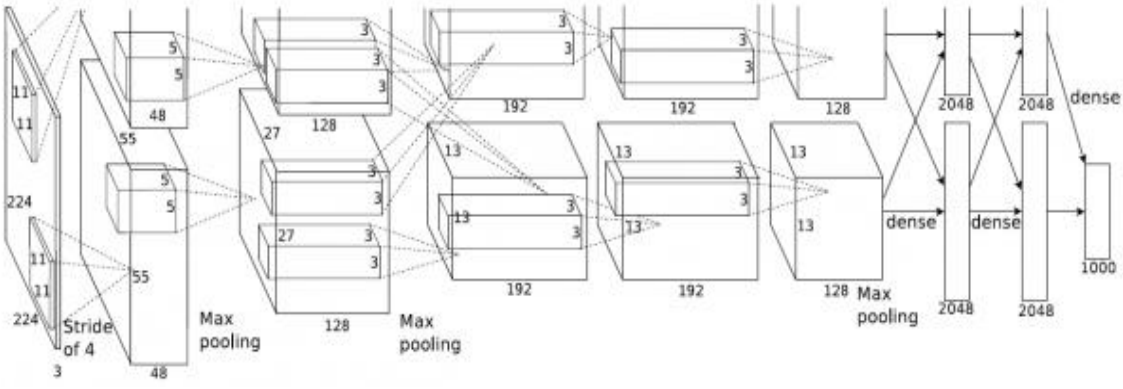

图 4.1 AlexNet 网络结构

数据增强是为了在已有的数据上创造出更多的数据。因为深度学习需要大量的数据才能够得到很好的效果，否则很可能导致过拟合之类的问题。在已有的数据有限的情况下，通过数据增强策略能够得到更多的数据用于训练，这样能够提高模型分类的效果。常用的数据增强策略包括水平翻转，随机裁剪、平移变换，颜色、光照变换。

Dropout[28]是在该模型中首次引入的防止过拟合的策略，简单来说就是让一些神经元之间的连接无效化。Dropout 的提出很好的降低了模型过拟合的问题。

RELU[29]则是一个新提出的激活函数，即类似于原来的 Sigmoid 函数的功能，同样是用于实现将输入的加权和做一个函数变换，得到的值作为神经元的输出值。不过相较于 Sigmoid，Tanh 类似的激活函数，RELU 具有以下几个优点：

1. ReLU 本质上是分段线性模型，前向计算非常简单，无需指数操作类似的高代价操作

2. 偏导的计算也很简单，反向传递梯度的时候无需指数对数之类的操作

3. 最关键的一点在于 ReLU 不容易发生梯度发散的问题，比如之前的很多激活函数在两端的时候导数容易趋于 0，多次连续求导之后梯度更加趋近于 0

4. 另外，ReLU 关闭了一边，会使很多的的隐层输出为 0，即网络变得稀疏，起到了一个类似于正则化的作用，可以缓解过拟合

总的来说，ReLU 的优点就是计算简单，避免梯度发散导致的模型无法训练，是更加适用于深度学习应用场景的激活函数。

在 VGG-Net 中，这些特征都得到了使用，因为它本身就是继承自 AlexNet 的。思想上类似于 AletNet，也是先做卷积操作提取特征，中间用池化操作聚合特征，然后使用全连接层进行总结，最后用 Softmax 层得到每个分类对应的概率。不过它的一个很明显的特点就是模型更深了，做了更多的卷积操作，同时导致计算量和神经元的个数也增加了很多。VGG-NET 的常见结构如图 4.2 所示。这种思想的成功使得后面的网络模型变得越来越深，最近的 ImageNet 中获得冠军的微软团队提出的 Residual Neural Network 的网络层次已经达到了 152 层之多。在数据量足够的情况下，residual neural network 的拟合能力非常强大。

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					

图 4.2 VGG-Net 的常用结构

4.3 哈希层

哈希层的思想源于论文 Deep learning of Binary Hash Codes for Fast Image Retrieval，论文中为了将图像的分类和 hash 结合到一起，在卷积神经网络上多加入了一层，这一层放在 Softmax 之前，作为 Hash Layer。这一层就是一个普通的全连接层，不过激活函数采用 Sigmoid。因为 Sigmoid 得到的值是在(0 , 1)范围之内的，通过一个阈值即可将向量转换成 0-1 向量。如果采用 ReLU，值的取值范围会很大，没有很好的策略可以将这个向量转换成 0-1 向量。具体的网络结构如图 4.3 所示。

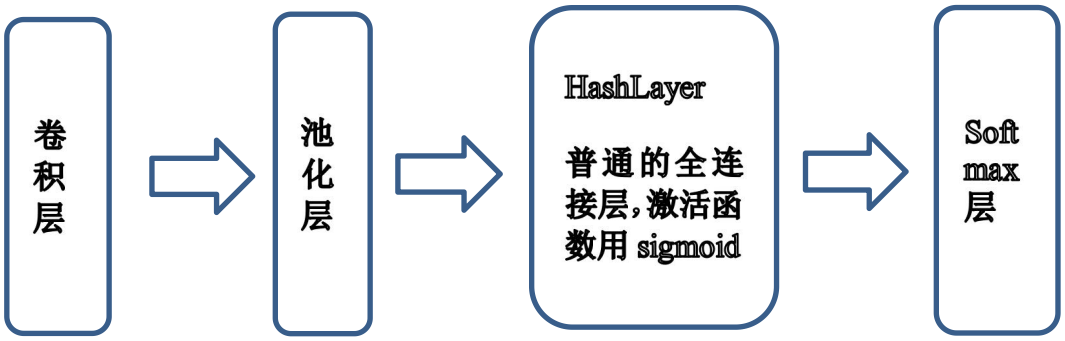

图 4.3 加入 Hash Layer 的方式

此外，Hash Layer 得到的哈希码不仅可以作为视频的一个标识，哈希码之间的距离同样可以作为视频相似性的一个衡量标准。这样就成功的将相似视频检索的问题转化成了 0-1 向量之间的相似向量查找。

Hash Layer 的可用性在之前的论文中已经得到了验证，本文在这里将 Hash Layer 引入视频数据集，验证它在视频数据集中的可用性。

4.4 优化的相似检索策略

4.4.1 常用的相似视频检索策略

4.4.1.1 顺序检索

在得到了所有视频对应的哈希码之后，最简单的一种检索算法就是遍历所有的视频，计算视频对应的哈希码的汉明距离，找出里面最接近的哈希码，即最相

似的那些视频。这样做的好处是实现起来很简单，不过该算法的时间代价较高。因为汉明距离的计算速度较快，时间主要消耗在遍历所有的视频，总的时间开销就是线性时间。在视频数据量比较大的情况下，时间增长的很快。

4.4.1.2 基于 KD-Tree 的检索策略

在这里视频检索问题实际上已经被转化成哈希码检索问题，所以相似视频检索问题已经变成了高维空间上的近邻检索问题。在解决近邻查找的问题时，一种经典的算法是使用 KD-Tree[30]来加速近邻的查找。

KD 树是一种对 k 维空间中的实例点进行存储以便对其进行快速搜索的树形结构，它是一棵二叉树，表示对 k 维空间的一个划分。构造 KD-Tree 相当于不断地用垂直于坐标轴的超平面将 k 维空间进行切分，构成一系列的 k 维超矩阵区域。KD-Tree 的空间划分如图 4.4 所示。在每个树结点中会存储以下信息：

1. 数据点，也就是数据集中的某个点
2. 该节点表示的空间范围
3. 该节点是按照哪个维度进行划分的
4. 左子树，右子树，parent 结点

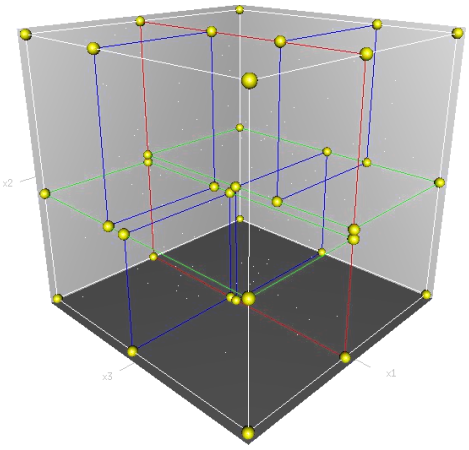

图 4.4 KD-Tree 在三维空间划分的示意图

KD 树的建立很简单，每次按照一个点进行一个维度的划分。将数据集分成两份，然后在分出来的两份上进行递归建树。

具体的查找算法的实现，其实也是利用了搜索中最常用的加速技巧，即剪枝，

这样可以避免无效的搜索。里面采用的剪枝策略为：当找到了一个近似的近邻点时，可以利用这个点来得到一个可能的近邻距离，如果一个结点的可能的最近距离都比这个大，那么这个结点的搜索就没有意义，无法得到更近的近邻，也就成功进行了剪枝，避免了无效的搜索。如果有可能得到更好的结果，那么还是要进行搜索。

利用 KD 树进行近邻搜索也不一定就能提高检索的效率，当维数比较大时，直接使用 KD 树进行搜索的话性能会下降的很厉害，剪枝会很少起到效果。所以这也不是通用的一个方案。

4.4.1.3 基于 ANN 的近似近邻查找

在实际应用场景下，可能不需要去求严格的 k 个最近邻的点，如果需求是检索出相似的视频，不一定要找出最相似的那些视频。所以近似近邻查找在这个应用场景中是很适合的，为了加速相似视频的查找牺牲一些准确率。具体的近似近邻查找算法的实现主要有两种策略：基于 KD-Tree 的近似近邻查找以及使用 LSH（局部敏感 hash）做近似近邻查找。

基于 KD-Tree 的近似近邻查找是由 Jeffrey S. Beis 和 David G. Lowe 提出来的，名为 Kd-tree with BBF（Best Bin First）的改进算法[31]。因为 KD-Tree 在做搜索的时候会做大量的回溯，开销会比较大，这个改进算法做的工作就是有效的减少回溯的次数。算法中设置了优先级队列和运行超时限定，优先级队列用于选择可能性更好的分支进行搜索，运行超时限定避免搜索进行的太深，搜索过深可能可以得到更多的近邻，但是付出的时间代价可能过高。通过这些限定和策略，算法可以更快的得到一些近似的近邻。

基于 LSH 的近似近邻查找是一种不一样的策略，它依赖于一类特殊的哈希函数，即局部敏感哈希[32]。与一般哈希函数不同的是它们具有位置敏感性，也就是散列前的相似点经过哈希之后，也能够在一定程度上相似，并且具有一定的概率保证。对于一个查询点 q，以及给定的距离阈值 r，搜索桶 g1(q)....gL(q)，取出其中的所有点 v1，....vn 作为候选近似最近邻点。对于任意的 vj，如果 D(q，vj) <= r，那么返回 vj，其中 D 为相似性度量函数。在创建 LSH 索引时，选取的

hash 函数是 k 个 LSH 函数的串联函数，这样就相对拉大了距离近的点冲突的概率 pn 与距离远的点冲突的概率 pf 之间的差值，但这同时也使这两个值一起减小了，于是需要同时使用 L 张哈希表来加大 pn 同时减少 pf。通过这样的构造过程，在查询时，与查询点近的点就有很大的概率被取出来作为候选近似最近邻点并进行最后的距离计算，而与查询点 q 距离远的点被当做候选近似最近邻点的概率则很小，从而能够在很短的时间内完成查询。LSH 的算法示意图如图 4.5 所示。

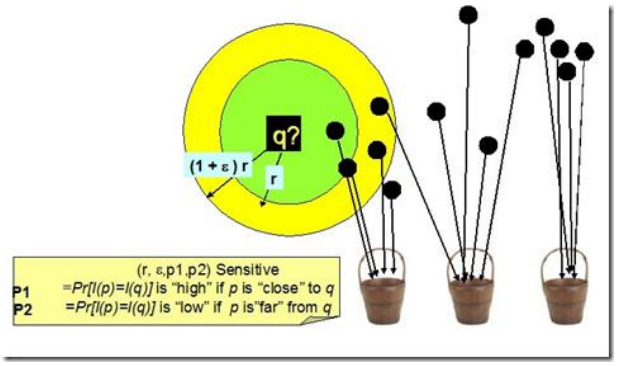

图 4.5 LSH 示意图

ANN 是更好的适用于相似视频查找的近邻查找算法，因为它速度更快，且不会牺牲太多的准确率，是更匹配本文应用场景的算法。不过这几个算法是相对通用的解决方案，没有利用到这个问题里面 0-1 向量的性质，接下来会提到论文中提出了怎样的一种方案来利用好 0-1 向量以及汉明距离的性质。

4.4.2 针对 Binarycode 的检索策略

论文中希望对每个视频得到对应的 Binary Hashcode。引入 0-1 向量的原因有几点：一个是计算距离更为简单，不会涉及到复杂的浮点数运算，只用二进制运算即可计算；另一个是更好表示，如果 0-1 向量的位数不是很长，一个 int32 或者 int64 就可以进行表示，相较于一个实数向量对应的实数数组，内存存储开销大幅降低。基于 0-1 向量的这些性质，本文提出一个新的基于 0-1 向量的相似视频检索策略，检索速度和存储开销都有所改进。

算法基本思想是，利用 Trie 树组织所有的 0-1 向量，在 Trie 树的叶子结点存储哈希码对应的视频，相同的哈希码的视频信息会存在一个 Trie 叶子结点中。

然后每次计算一个视频的相似视频时，迭代加深 Dfs 判断所有可能的近邻哈希码，每次以哈希码之间差异的位数作为迭代的目标，即相差 0 位，相差 1 位，相差 2 位.....。每次去判断所有可能的哈希码是否存在对应的视频，如果存在就找到了一些近邻视频。如果找到了足够多的相似视频，就停止查找操作，结束相似视频的查找。伪代码如下，算法流程图如图 4.6 所示。

```
limit = 3          // the maximum hamming_distance will be checked
check_hamming_dist = 0   // the hamming_distance checked every time
total_videos_retrieval = 0  // the total number of videos
results = []
while total_videos_retrieval < k && check_hamming_dist < limit:
    hashcodes = get_possible_hashcodes(check_hamming_dist)
    for hashcode in hashcodes:
        videos = get_videos(hashcode)
        results add videos
        total_videos_retrieval += len(videos)
    // update the possible hamming distance
    check_hamming_dist += 1
return results
```

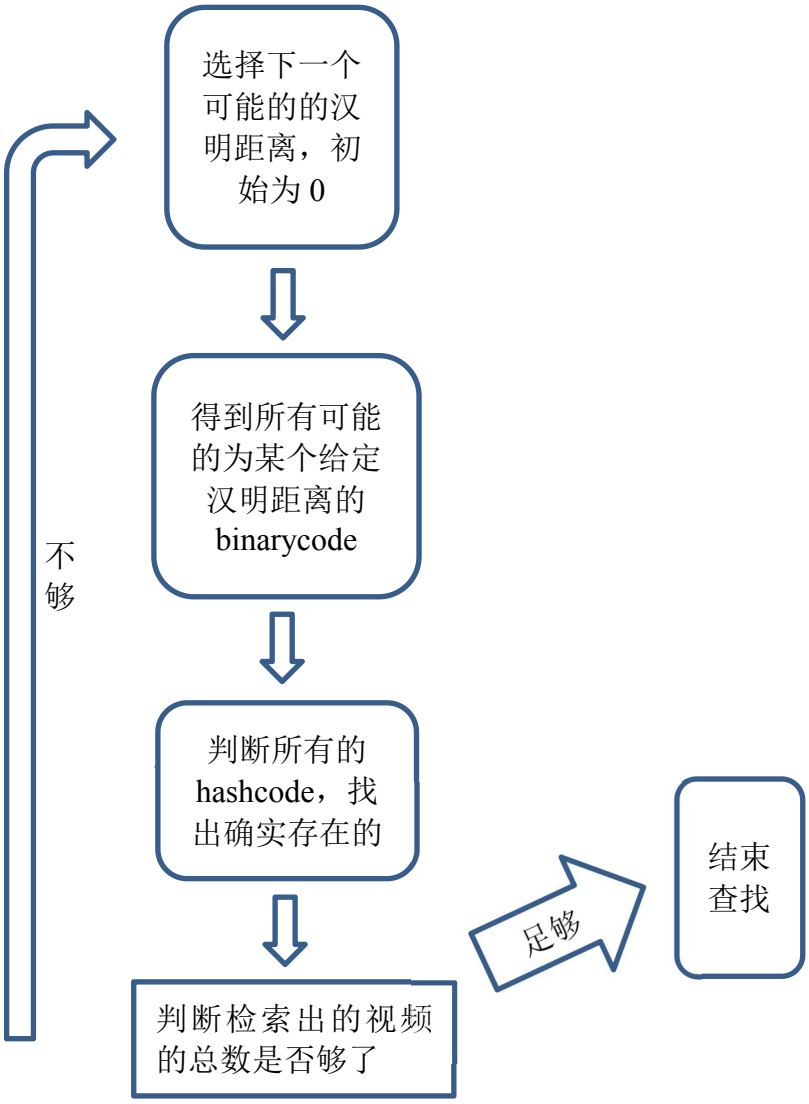

图 4.6 算法流程图

Trie 树是用于前缀检索的一种非常有效的数据结构。因为本文采用 0-1 向量作为视频数据的哈希码，对应的 Trie 是一棵二叉树，左右孩子分别表示某一位取 0 或者取 1。在叶子结点会存储对应的视频信息，即到叶子结点的哈希码对应的视频是哪些。如果要判断一个哈希码是否存在，最坏需要 O(len(hashcode)) 的比较时间。实际应用中可能因为前缀无法匹配上，马上就能中止查找，速度会更快。这也是哈希表类似的解决方案没有的好处，因为这些数据结构无法提供类

似于前缀比较的功能。总的来说，Trie在本文的视频检索算法里面起到的一个作用是将相同的哈希码对应的视频聚集起来，在一个叶子结点里面进行存储；另一个重要的用处是用于实时判断当前正在判断的哈希码前缀是否存在，如果不存在就可以提前中止该哈希码的判断。

迭代加深dfs是结合了dfs和bfs优势的一种算法，简单来说就是每次限定搜索的深度进行dfs。单纯的dfs算法的缺点是容易搜得太深，bfs算法的缺点则是耗费大量内存。迭代加深比较好的结合了这两种算法，一方面深度逐渐增加；一方面不会用队列维护一层所有的结点，节省了内存开销。此外迭代加深本身也很适合本文的应用场景，因为相似视频的定义就是汉明距离相近，距离从0开始不断判断的过程和本文中的应用场景相吻合。搜索过程中深度不断改变的也就是汉明距离的具体值，为了每次去判断不同的汉明距离对应的所有可能哈希码，如果存在就找到了一些相似视频，算法会不断执行直到找到足够的视频或者判断的深度达到了上限。

本文在此分析一下算法的时间开销。哈希码判断本身的时间复杂度是很低的，也就是O(len(hashcode))，取决于哈希码的长度，实际平均下来运行速度会更快。主要的时间开销在于找出不同的汉明距离对应的可能的hashcodes，如果汉明距离为0，可能的哈希码只有一个；如果汉明距离为1，可能的哈希码就存在len(hashcode)个；如果距离为2，可能的哈希码就存在len(hashcode)^2个.....从上述分析可知，如果距离很大的话，搜索空间会很大。虽然采用了Trie树用于剪枝，如果数据量很大剪枝的效果也不会很好，因为数据之间的前缀关系可能会很多。不过这对算法的有效性不会产生影响，因为在该应用场景下需要做的事情是相似视频检索，如果汉明距离过大，实际上就不能被定义为相似视频，也就没有搜索的必要。所以算法只需要去搜索那些小的汉明距离对应的哈希码，比如3以内，最大的开销就是len(hashcode)^3，如果哈希码采用32位，开销就是2^15，在数据量很大的情况下这是非常小的一个数字。同时因为剪枝的存在时间开销会更小。更好的一点在于这个时间开销和视频的数目无关，只和哈希码的位数有关，时间开销不会随着视频数目的增长而增大。总的来说，这个方法能够

做到准确相似视频查找，且开销不会随着视频数目的增加而增大。

算法的优点有以下几点：

1. 将相似的视频聚合到一起，可以一次取出一个哈希码对应的所有视频，降低了时间开销。

2. 使用 Trie 在判断的过程中进行剪枝，加速哈希码是否存在的判断。

3. 利用了相似视频的概念，确定搜索的汉明距离的上限，同时确定了迭代加深 dfs 的深度上限，避免无效的搜索。

算法的缺点有以下几点：

1. 如果数据量不够大，可能生成的哈希码差距都很大，这时如果还是希望进行相似视频的检索，开销会很大，因为复杂度是 O(len(hashcode)) ^ n 的，n 为对应的汉明距离。时间的增长可能很快会超过线性查询。

2. Trie 树的存储相较于哈希表之类的数据结构冗余度是比较高的，因为存储了前缀的信息，所以内存开销是比较大的，在内存比较紧张的环境下不是很适用。

总的来说，这个新的基于 0-1 向量的视频检索策略不是通用的近邻检索策略，是适用于特定应用场景的一种解决方案。如果不是 0-1 向量算法无法使用 Trie 进行哈希码的存储；如果不是通过汉明距离进行哈希码之间距离的度量，也无法通过搜索来判断可能的哈希码。但是在这个应用场景下，这个方法相较于其它通用算法更加有效。

4.4.3 整合的相似视频检索策略

因为不同的解决方案都有各自的优势，如果能够结合起来进行相似视频检索的话能够起到更好的效果。

本文中提出的基于哈希码的相似视频检索算法在哈希码是 0-1 向量的应用场景下会有不错的表现，但是在汉明距离相差比较大，且还是需要进行相似视频查找的情况下工作的会很差。如果能够将其它的近邻查找算法和本文中的算法结合起来，会起到更好的效果。

整合算法的大致框架如下：首先采用本文中的近邻搜索算法进行相似视频的检索，找出汉明距离不超过一定值的所有的相似视频。因为算法本身会很快，这个流程会很快执行完成。如果得到的相似视频很少，说明当前处于本文中提到的场景，即汉明距离相差较大且需要找相似视频。这时算法选择放弃该搜索策略，选择其它的近邻查找算法进行查找。最后将两个算法的结果做一下并集操作，得到最后的结果。总的来说，就是利用本文中的算法先做一个快速的查找，如果效果不好就换回其它的近邻检索算法。这样整个算法就能够适应所有的场景，且平均下来的效果更好。整合的相似视频检索算法框架如图 4.7。

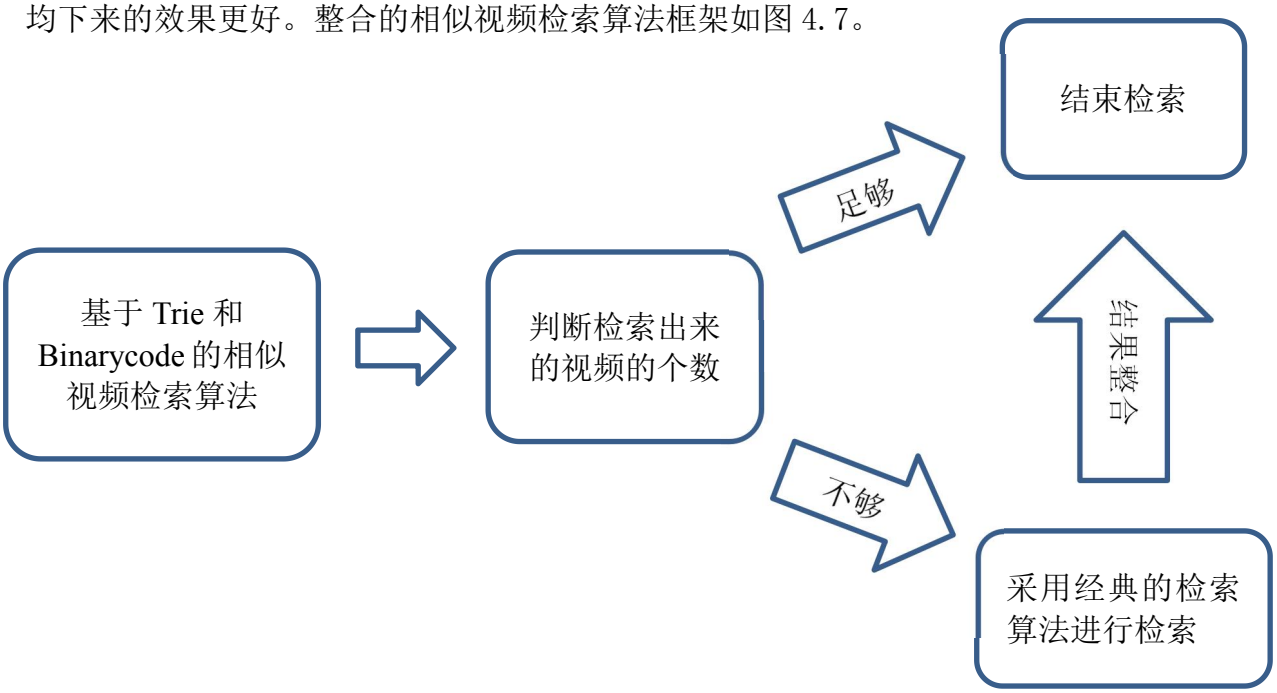

图 4.7 整合的相似视频检索算法框架

4.5 本章小结

本章详细阐述了论文中需要实现的算法以及相关研究领域使用的算法。引入了具体使用的卷积神经网络模型结构 VGG-NET，解释了模型的具体结构和设计思想。定义了哈希层的概念，解释了采用 Sigmoid 作为激活函数的原因，说明了得到 0-1 向量的方式。最后介绍了常用的近邻检索策略，包括顺序检索的策略，基于 KD-Tree 的近邻检索策略，基于 ANN 的近邻检索策略，以及本文中提出的适用于 0-1 向量应用场景的近邻检索算法。

第 5 章 本章对比实验结果

本章是本论文的实验部分，包括实验环境的介绍，实验内容的介绍以及实验结果的介绍，最后对实验的数据结果做一个展示。

5.1 实验配置

5.1.1 运行环境

项目	内容
中央处理器	Intel Xeon E5-2660
内存	64GB
GPU 显卡	NVIDIA GeForce GTX TITAN X
操作系统	Linux Ubuntu 14.04 LTS server
Cuda	Cuda8.0 with cudnn
数据处理	Python2.7,OpenCV,Numpy,Keras, Theano

5.1.2 实验数据描述

本文主要在两个公开数据集上进行实验。它们已经被研究人员大量用于算法研究，具有很高的可靠性。

一个是名为 WEIZMANN 的行为视频数据集[33]，包含 90 个 180 * 144 分辨率的视频，内容都是一个人的某种行为，包括 walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, skip 这 10 类操作。视频长度都在 5s 之内，都是比较短的视频，视频的帧的数目不会超过 100 帧。所有的视频已经标注好 label，且所有的视频都是单一 label 的视频，也就是实验的目标都是对视频做单分类，不会存在多分类的情况。该数据集本身是用于行为分析相关的研究，用于测试识别人行为的算法，本文在这里将其用于视频分类任务。

每一类对应的 label 如表格 5.1：

表格 5.1 数据的 label 和和编号对应表

Pjump	0
Side	1
Jack	2
Wave2	3
Bend	4
Skip	5
Jump	6
Walk	7
Wave1	8
Run	9

一个是名为 UCF50 的运动视频数据集[34]，其中包含 6000 多个 320 * 240 的视频数据，总共的类别个数有 50 个，都是 diving , swing , YoYo 之类的体育运动。视频的长度都在 10s 左右，每个视频的帧的个数不会超过 1000 帧，是比较短的视频。所有的视频都是单一的类别，即都是一种体育运动，不存在多分类的情况。相较于第一个数据集，里面包含的行为更加复杂，是一个完整的体育运动。

5.1.3 实验对比算法

实验主要分为两块进行对比，一个是针对视频分类的结果进行对比，比较基于卷积神经网络的视频分类算法和基于图像特征的视频分类算法的准确率；一个是针对视频检索的准确率和速度进行比较，比较基于欧氏距离的朴素检索算法、基于汉明距离的朴素检索算法以及基于 Trie 树的近邻检索算法的检索效率。具体说明如下：

视频分类方面将本文中使用的基于卷积神经网络的视频分类算法和论文中的分类算法进行比较，比较分类的准确率。

视频检索方面将本文中基于 Trie 树的近邻检索策略和朴素的依次比较检索策

略进行比较，比较检索的时间。同时测试检索的准确率。

5.1.4 评判指标

评判指标主要分为准确率和时间两块。准确率在分类和检索中分别表示模型在测试集上分类的准确率以及模型检索出的相似视频中 label 确实相同的视频个数占总的检索出的相似视频的个数的比例，时间主要是比较得到 hashcode 之后在 hashcode 集合中检索出相似的 hashcode 所花的时间，不同的算法检索出相似 hashcode 所花的时间不同。具体的指标定义如下：

视频分类的准确率的定义为分类正确的视频的个数占总的测试集视频个数的比例，公式即为 $\frac{number\_of\_correct\_classified\_videos}{number\_of\_videos}$。视频检索的准确率即为检索出的视频中 label 正确的个数占总的检索出的相似视频个数的比例，公式即为 $\frac{number\_of\_correct\_videos\_retrieval}{number\_of\_videos\_retrieval}$。

时间的评估不存在公式，具体的定义就是从得到视频对应的 hashcode 开始，到计算出所有相似的 hashcode 对应的视频为止，整个的时间即为检索需要的时间。本文中会计算三种方法需要的时间，一个是使用欧氏距离的朴素检索算法需要的时间，具体方式为依次计算 hashcode 之间的欧氏距离，然后找最接近的那些 hashcodes；一个是基于汉明距离的朴素检索算法需要的时间，具体方式即为依次计算 hashcode 之间的汉明距离，然后找最接近的那些 hashcodes；最后一个是基于本文提出的优化算法所需要的检索时间，即采用汉明距离同时用 Trie 进行优化的算法。

5.2 实验过程和步骤

本文中用来做对比的实验算法都采用 python 代码进行实现，包括基于卷积神经网络的视频分类算法，基于 Binarycode 的相似视频检索算法，检索算法包括基于欧氏距离的朴素比较算法，基于汉明距离的朴素比较算法，基于 Trie 树的近邻

检索算法。哈希层神经元的个数在所有的实验里固定为 32，即一个整数的大小，这是为了利用位运算更快的计算汉明距离，从而减少检索 Hashcode 的时间。分类算法和检索算法的算法流程如下：

基于卷积神经网络的视频分类算法：

1. 训练卷积神经网络模型，输入是视频的帧，训练目标是帧对应的 Label。
2. 对测试数据集中视频的帧进行分类，得到不同的帧对应的不同分类结果。
3. 分类结果的众数即为对这个视频的分类结果。

基于 Hashcode 的相似视频检索：

1. 通过卷积神经网络得到视频对应的 Hashcode。
2. 在已有的 Hashcode 集合上进行相似 Hashcode 的查找。
3. 如果是朴素依次比较的算法，需要遍历所有的 Hashcode，找出里面距离最近的 k 个，具体的查找算法可以用堆维护最近的 k 个 Hashcode，时间复杂度为 O(nlogk)；如果是本文中给出的改进算法，需要先把所有的哈希码存入 Trie 树，然后从 0 开始判断可能的汉明距离，找出汉明距离对应的所有 Hashcode，判断这些 Hashcode 是否存在，如果存在找出对应的视频。

5.3 实验结果与分析

WEIZMANN 数据集包含 90 个分辨率为 144 * 180 的短视频，长度都在几秒之内，帧数不会超过 300。总共包含 10 个类别，都是人的某种行为。本文在数据集上进行以下几个实验：

1. 进行基于卷积神经网络的视频分类算法的实现，得到算法的分类准确率，和之前论文中的结果进行比较
2. 测试卷积神经网络得到的 Binarycode 的相似视频检索准确率
3. 比较不同的相似视频检索算法所需要的时间，包括采用欧氏距离的依次比较算法，采用汉明距离的依次比较算法以及基于 Trie 树的近邻检索算法

分类算法采用的网络模型如图 5.1，基本按照 VGG-NET 的结构进行设计。训

练的时候学习速率采用了较小的 0.001，训练速率较小是为了得到更好的模型。其它的参数和默认的参数基本一致。训练的输入是视频的一些帧，训练目标是帧对应的视频的 Label。本文随机将全部的 90 个视频按照 2 : 1 的比例划分成训练集和测试集，训练集总共有 60 个视频，测试集总共有 30 个视频。在训练集上视频的帧进行训练，在测试集上进行模型结果的测试。测试结果如表格 5.2。

```
# train model
model = Seq()

model.add(Conv2D(64 , 3 , 3 , input_shape = (CHANNEL_NUM , WIDTH , HEIG
HT)))
model.add(Activation('relu'))
model.add(Max2D((2 , 2)))

model.add(Conv2D(64 , 3 , 3))
model.add(Activation('relu'))
model.add(Max2D((2 , 2)))

model.add(Conv2D(128 , 3 , 3))
model.add(Activation('relu'))
model.add(Conv2D(128 , 3 , 3))
model.add(Activation('relu'))
model.add(Max2D((2 , 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(1024))
```

```
model.add(Activation('relu'))
model.add(Dropout(0.5))

# hash layer
model.add(Dense(32))
model.add(Activation('sigmoid'))

model.add(Dense(CLASS_NUM))
model.add(Activation('softmax'))

# SGD
sgd = SGD(lr = 0.001 , momentum = 0.9 , nesterov = True , decay = 1e-6)
model.compile(loss = 'categorical_crossentropy' , optimizer = sgd , met
rics = ['accuracy'])
```

图 5.1 分类模型结构

表格 5.2 分类结果

论文中的分类准确率	0.967%
本文基于卷积神经网络的分类模型的准确率	0.967%

从以上数据可以得知，基于卷积神经网络的模型和论文中的结果相近。基于卷积神经网络的模型没有很大的优势有几点原因，一个是因为该数据集本身是用

于测试行为识别算法，不是用于视频分类；第二个是因为数据量较小，模型拟合能力无法体现。这里本文只是证明基于卷积神经网络的视频分类模型的有效性，具体的算法的优势在大数据集上会有一定的体现。

视频检索方面的实验主要测试两个方面，一个是检索出来的视频的准确率，即检索出来的 k 个相似视频中类别正确的比例，这里本文用测试集里面所有的视频作为输入，在训练集中做检索，计算所有测试集视频对应的所有 k 值检索准确率，然后取平均值作为整个测试集检索准确率的评估；另一个是检索的速度，即检索相似视频需要多少时间。因为这个数据集中的视频数目很小，所以这里本文把检索的范围放宽到帧，即做图像层面的检索，训练集总共存在 4094 帧，测试集存在 1607 帧。检索的准确率如图 5.2，检索时间表如表 5.3。从图中可以看出检索的准确率和视频分类的准确率类似。检索时间方面，需要进行说明的是试验中没有包括预处理哈希码的时间，即插入 Trie 树的时间。整个检索花费的时间是从得到输入视频对应的哈希码开始，到检索出所有的相似哈希码的时间，不包括其它处理过程的时间。本文中将检索的 TopK 设为 20，即找出最相似的 20 帧。另外因为基于 Trie 的相似哈希码查找算法是按照汉明距离迭代加深处理的，如果找到的帧的数目达到了需求就停止算法，所以可能找出的相似帧的数目会超过 k 的需求，不过这不会影响查找的速度。从结果上可以看出，基于 Trie 的相似哈希码查找算法的查找速度是朴素依次比较算法的 1%。

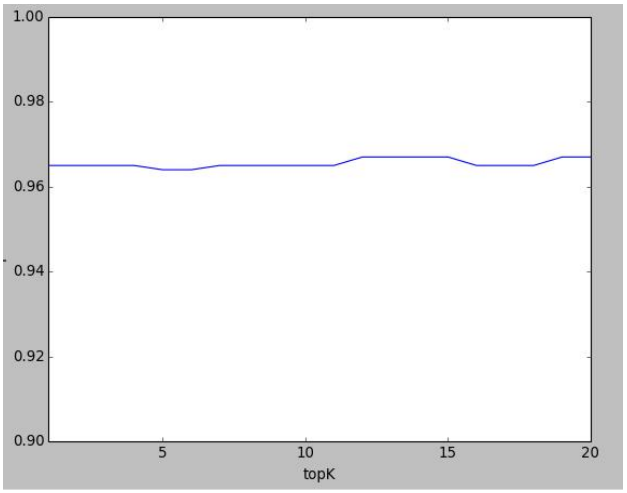

图 5.2 测试集视频帧检索准确率结果

表格 5.3 检索时间结果

基于欧氏距离的朴素比较算法的检索时间	4.45308685303e-05s
基于汉明距离的朴素比较算法检索时间	2.17740535736e-05s
基于 Trie 树优化的检索时间	6.103515625e-08s

UCF50 是比较大的一个数据集，里面包含 50 类数据，总共有 6000 多个分辨率为 320 * 240 的短视频，每个视频的长度在几秒到 10 几秒不等。本文在这个数据集上做的工作和本文在 WEIZMANN 数据集上做的工作类似，包括测试视频分类的准确率，视频检索的准确率以及检索的时间效率。训练集和测试集的划分比例为 7 : 3，即 4200 多个训练集视频和 1800 多个测试集视频。因为这个数据集中的样本是上一个数据集的几十倍，在该数据集做检索相关的实验就是以视频为单位进行检索，视频分类的实验过程和上一个数据集相同。相似视频检索中的 TopK 依然取为 20。分类准确率和检索时间如表 5.4，表 5.5，检索准确率如图 5.3 所示。

表格 5.4 分类结果

论文中视频分类的准确率	76.9%
本文中视频分类的准确率	80.6%

表格 5.5 检索时间结果表

基于欧氏距离的朴素比较算法的检索时间	4.87370491028e-05s
基于汉明距离的朴素比较算法检索时间	2.63271331787e-05s
基于 Trie 树优化的检索时间	1.88112258911e-07s

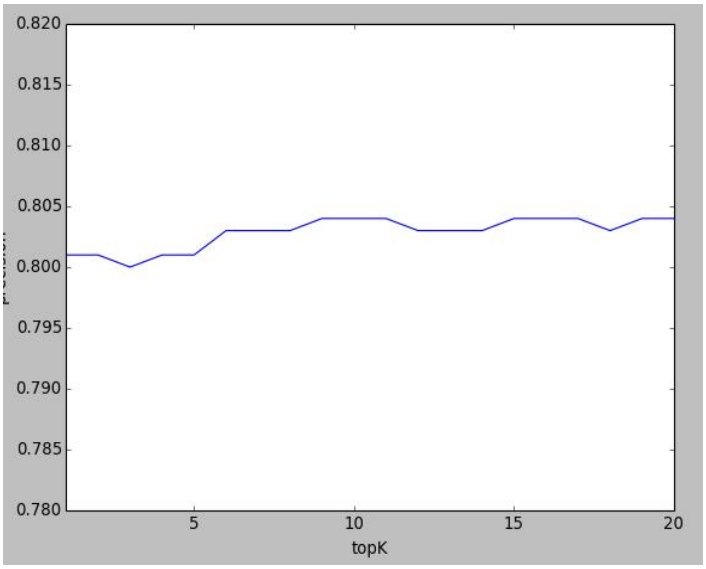

图 5.3 检索准确率随 k 值的变化曲线

5.4 本章小结

本章是论文的实验部分，简单介绍了对比实验中涉及到的算法，包括算法的执行流程以及算法的原理。本文中采用的数据集都是行为相关的标准数据集，所有的实验都在这些数据集上进行，结果展示都是基于这些数据集。最后的实验结果表明，论文中基于卷积神经网络的视频分类算法比基于特征的视频分类算法的准确率要高出 1%~3%，论文中提出的基于 Trie 树的相似视频检索算法比朴素比较算法的检索速度要快，时间开销是朴素比较算法的 1%。

第 6 章 总结和展望

6.1 本文工作总结

本文的两个侧重点分别是视频的分类和相似视频的检索。视频分类的传统做法是提取单帧的特征作为类似于支持向量机的分类器的输入，视频的标签作为分类训练的目标，进行分类器的训练。在预测一个视频的类别的时候，对关键帧进行分类，从而判断视频的类别。这种方法的缺点和利用特征进行图像分类一样，因为利用特征以及词袋模型进行分类的准确率要低于基于卷积神经网络的分类算法。后续研究人员提出的扩展卷积层以及循环神经网络的使用进一步改进了分类准确率。相似视频检索的传统做法类似于图像检索，先从视频中提取出特征向量，然后在特征向量空间里进行近邻查找。这里本文使用 0-1 向量作为视频的哈希码，因为汉明距离的计算代价是要小于浮点数计算的，所以相似度计算要更加的容易。此外本文中提出的相似视频哈希码查找算法也是依赖于 0-1 向量表示的，所以 0-1 向量表示具有很大的优势。

总的来说，本文的工作量如下：

1. 研究 VGG Net 模型在视频分类任务中的使用，包括模型结构的设计以及模型参数的调整。
2. 研究了哈希层的引入是否有效，包括是否会对模型的分类效果产生影响，以及得到的 0-1 向量是否有效。
3. 提出了新的基于 0-1 向量的相似视频检索策略，验证了检索策略的准确性和时间上的优越性。
4. 将本文中所提出的所有思想在两个公开数据集上做了有效性的验证，对算法的结果做直观的展示。

6.2 未来研究工作

在本文的工作中，包括了分类和检索两个部分，这两个部分都存在待改进或者待提高的地方，未来的研究工作会主要集中在这两部分，具体的设想如下：

1. 这里的视频分类问题是单分类的问题，而实际应用中往往一个视频可以对应很多的标签，如何解决多分类的问题是一个很大的挑战。

2. 因为视频的分辨率要比常用的图片数据集高很多，且一个视频可能会对应很多的帧，视频可用的数据量要远大于图像。如何在有限的资源下利用好大量的数据是下一步需要研究的方向，包括如何提取关键帧以及重新调整视频帧大小是否可行。

3. 本文第五章提出了算法的可能应用场景，没有进行完整的实现，无法确定算法在这些应用场景下是否确实可用。对这些应用场景的完整实现很有必要。

4. 现在最为主流的视频分类方式是结合卷积神经网络和循环神经网络的视频分类算法，利用卷积神经网络分类帧，利用循环神经网络捕捉时间序列上的关系。尝试这方面的算法是未来的一个研究方向。

6.3 本章小结

本章概括了全文的工作，对未来的工作做展望。全文工作主要包括两个部分，一部分是实现了基于卷积神经网络的视频分类算法，成功在其中引入了哈希层实现对视频的哈希，得到视频对应的 0-1 哈希码；另一部分是提出了基于 Trie 树的相似视频检索算法，提高了检索的速度，同时保证检索的准确率。目前深度学习的发展还是非常的迅猛，一方面是因为算法本身的不断改进不断提出；另一方面是因为硬件的发展以及系统架构的发展，现在能够承载更大的数据量以及处理更多的数据。相信在未来深度学习依然会是图像、视频领域的主流解决方案，且会向更多的新领域发展。

参考文献

[1] GE Hinton, S Osindero A Fast Learning Algorithm for Deep Belief Nets Neural computation, 2006

[2] Y LeCun, Y Bengio, G Hinton Deep learning - Nature, 2015

[3] Y Boureau, F Bach, Y LeCun, J Ponce Learning mid-level features for recognition Computer Vision and Pattern Recognition 2010. CVPR, 2010

[4] Y LeCun, K Kavukcuoglu, C Farabet Convolutional networks and applications in vision. ISCAS, 2012

[5] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, Chu-Song Chen Deep learning of Binary Hash Codes for Fast Image Retrieval CVPR, 2015

[6] C Farabet, C Couprie, L Najman, Y LeCun Learning Hierarchical Features for Scene Labeling IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013

[7] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In Proc. ECCV, pages 584 – 599. Springer, 2014

[8] J. Deng, A. C. Berg, and F.-F. Li. Hierarchical semantic indexing for large scale image retrieval. In Proc. CVPR, 2011

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proc. CVPR, 2014

[10] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In Proc. CVPR, pages 1 – 8, 2008

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In Proc. NIPS, 2012

[12] A. Krizhevsky. Learning multiple layers of features from tiny images. Computer Science Department, University of Toronto, Tech. Report, 2009

[13] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In Proc. CVPR,2012

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91 – 110, 2004

[15] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul

Sukthankar, Li Fei Fei Large-scale Video Classification with Convolutional Neural Networks CVPR, 2014

[16] R. Salakhutdinov and G. Hinton. Semantic hashing. International Journal of Approximate Reasoning, 500(3):500, 2007

[17] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In Proc. ACM MM, pages 157–166, 2014

[18] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In Proc. AAAI, 2014

[19] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In Proc. CVPR, 2010

[20] A. Krizhevsky and G. E. Hinton. Using very deep autoencoders for content-based image retrieval. In ESANN, 2011

[21] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. PAMI, 2002

[22] S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. Workshop on Models versus Exemplars in Computer Vision, 2001

[23] L. Gorelick, M. Galun, E. Sharon, A. Brandt, and R. Basri. Shape representation and recognition using the poisson equation. CVPR, 2:61–67, 2004

[24] I. Laptev and T. Lindeberg. Space-time interest points. ICCV, 2003

[25] M.D.Zeiler and R.Fergus. Visualizing and understanding convolutional networks. In Proc. ECCV, pages 818–833. Springer, 2014

[26] David G. Lowe Distinctive Image Features from Scale-Invariant Keypoints IJCV, 2004

[27] Matthew Brown and David Lowe Invariant Features from Interest Point Groups, 2002

[28] Going Deeper with Convolutions, Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, 2014

[29] Learning Convolutional Feature Hierachies for Visual Recognition, Koray

Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu and Yann LeCun, Advances in Neural Information Processing Systems (NIPS 2010), 2010

[30] Graves, Alex, et al. “A novel connectionist system for unconstrained handwriting recognition.” Pattern Analysis and Machine Intelligence, IEEE Transactions, 2009

[31] V.N. GudiVada V.V. Raghavan Content based image retrieval systems IEEE, 1995

[32] Implementation and benchmarking of perceptual image hash functions C Zauner phash.org, 2010

[33] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani and Ronen Basri Actions as Space-Time Shapes ICCV, 2005

[34] Kishore K. Reddy, and Mubarak Shah, Recognizing 50 Human Action Categories of Web Videos, Machine Vision and Applications Journal (MVAP), 2012

攻读硕士学位期间主要的研究成果

致谢

首先感谢开源框架 theano，keras 的作者，如果没有这两个非常强大且易用的 Python 框架，实验代码的编写不会那么的容易。另外因为整个实验针对的是视频数据，需要对视频处理，其中用到了 OpenCV，同时也非常感谢 OpenCV 的作者，让视频图像处理变得异常简单。

论文的选题和顺利完成离不开导师组里老师们的帮助，无论是导师寿黎但老师，还是陈刚老师，陈珂老师，伍赛老师，都对我的论文提出了很多的意见，论文最终能够得以完成和老师们的帮助密不可分。两年半下来，无论是技术方面，还是为人处世方面，老师们都给了我很多的指导，能够走到今天，离不开老师们的指引。在此感谢老师们的帮助。

论文的最终完成同样离不开实验室同学们的帮助，如果没有大家的帮助和支持，碰到的很多困难可能就不会那么顺利的解决，论文的完成也不会那么顺利。同时也感谢日常生活时大家的陪伴，实验室工作以及找工作期间大家的帮助和支持，如果没有这些，我不能走到今天这一步。

最后感谢浙江大学这所百年名校，在这里两年半的学习经历让我开阔了眼界，提高了自己的人文素质和科研技术水平，成为了一个更为完善的人。就算在未来离开学校，走向社会工作以后，我依然会以自己曾经作为浙江大学的一员而感到骄傲。与此同时我会努力成为一个更好的人，无愧浙江大学对我的教育和栽培。

署名

当前日期