



中南大學

CENTRAL SOUTH UNIVERSITY

本科毕业设计(论文)

GRADUATION DESIGN (THESIS)

题目: 基于 CNN 的室内导航系
统设计与开发

学生姓名: 赵俊杰

指导教师: 任炬教授

学院: 计算机学院

专业班级: 计科 1501

本科生院制

2019 年 6 月

基于 CNN 的室内导航系统设计与开发

摘要

随着近年来软硬件的蓬勃发展，智能手机等集成有 GPS 组件的设备可以为我们在室外开放环境中的出行活动提供便捷可靠的定位和导航服务。但是，由于 GPS 信号在城市大型建筑内部的严重衰减，精细的室内地图构建难度大，以及高昂的布署成本等原因，室内环境的定位和导航问题目前仍缺少优秀的解决方案。

为了解决在室内导航中如何提供精确的定位、如何构建精细的室内地图、以及如何有效降低系统布署成本等难点和挑战，本文提出 DeepNavi，一种基于 CNN 的室内导航系统。首先，在 DeepNavi 中我们训练和使用在图片分类方面具有先天优势的卷积神经网络，通过神经网络对用户所处实时位置的图片进行识别来为用户提供精确的定位。其次，在 DeepNavi 系统中，我们通过一段路径的图像信息为该段路径构建一份详细而精确的室内地图，并且同一区域内多段路径的地图可以组合为该区域的完整地图提供对该区域的导航。此外，DeepNavi 系统的布署成本很低，而且布署方式相当简单，只需要录制一段目标路径的视频，系统便会自动完成在该路径上的布署。

DeepNavi 的优势在于确保精确的定位和构建精细的室内图的同时只需要花费相当小的成本。系统的布署和运行不需要额外的基础设施支持（如 Wi-Fi AP），完全依赖布署区域的图像信息，可以完美覆盖绝大多数室内场景。而且系统简洁高效，灵活性高，不必对完整一层甚至整个建筑物进行布署，对单独的一小段路径完成布署即可在该段路径使用，因此也能很好地支持动态性高的室内场景。

为了测试 DeepNavi 系统在定位精度、部署成本、纠偏能力以及稳定性等方面的表现，我们在运行 Android 操作系统的智能手机上实现了一个原型系统，并在一栋 7 层的教学楼内部进行了布署和测试。多组测试数据表明，DeepNavi 可以提供 1 米误差范围内的精确定位，系统布署过程耗时短，操作简单，并且系统在纠偏和服务稳定性等方面均表现出色。

关键词：室内导航 定位 地图 布署成本 CNN

Design and Development of CNN-based Indoor Navigation System

ABSTRACT

With the vigorous development of hardware and software in recent years, devices integrated with GPS components such as smart phones can provide convenient and reliable positioning and navigation services for our outdoor travel activities in open environments. However, due to the serious attenuation of GPS signals in large urban buildings, the difficulty of building accurate indoor maps, and the high cost of deployment, there is still a lack of excellent solutions for indoor environment positioning and navigation.

In order to solve the difficulties and challenges in indoor navigation, such as how to provide accurate positioning, how to build accurate indoor maps, and how to effectively reduce the cost of system deployment, this paper proposes DeepNavi, an indoor navigation system based on CNN. Firstly, in DeepNavi, we train and use convolutional neural network, which has innate advantages in image classification, to recognize the real-time position of the user's image through the neural network to provide accurate positioning for the user. Secondly, in DeepNavi system, we construct a detailed and accurate indoor map for the path through the image information of a path, and the map of multi-path in the same region can be combined to provide navigation for the whole map of the region. In addition, the deployment cost of DeepNavi system is very low, and the deployment method is quite simple. It only needs to record a video of the target path, and the system will automatically deploy on the path.

DeepNavi has the advantage of ensuring accurate positioning and building fine interior maps at a relatively low cost. The deployment and operation of the system does not require additional infrastructure support (such as Wi-Fi AP), and relies entirely on the image information of the deployment area, which can cover most indoor scenes perfectly. Moreover, the system is concise, efficient and flexible. It does not need to deploy the whole floor or

even the whole building. A single path can be deployed in this section, so it can also support the indoor scene with high dynamic performance.

In order to test the performance of DeepNavi system in positioning accuracy, deployment cost, corrective ability and stability, we implemented a prototype system on a smartphone running Android operating system, and deployed and tested it in a 7-storey teaching building. Several sets of test data show that DeepNavi can provide precise positioning within a 1-meter error range. The system deployment process is time-consuming and easy to operate, and the system performs well in rectification and service stability.

Key words: Indoor Navigation Location Map Deployment Cost CNN

目录

第 1 章	绪论	1
1.1	研究背景和研究意义	1
1.2	研究目标	1
1.3	研究现状及其缺点	1
1.3.1	定位和导航方式的研究	2
1.3.2	地图构建方式的研究	2
1.3.3	导航系统架构的研究	3
1.4	研究挑战和解决措施	3
1.4.1	高精度的定位方式	3
1.4.2	数据采集和地图构建方式	3
1.4.3	针对室内环境高动态性的解决方法	3
1.4.4	快速响应、稳定的系统架构	4
1.5	动机	5
1.6	内容安排	5
第 2 章	系统概况	6
2.1	需求分析	6
2.2	关键技术设计	6
2.2.1	室内地图的构建	6
2.2.2	DeepNavi 的定位原理	7
2.3	布署方法和使用方式	8
2.4	系统架构概述	9
2.5	本章小结	10
第 3 章	系统的构建	11
3.1	构建工作概述	11
3.2	数据采集和标注	11
3.3	地图的构建	13
3.4	模型和 label	14

3.5	图片中行人的擦除	15
3.6	本章小结	16
第 4 章	定位和导航	17
4.1	DeepNavi 定位和导航概述	17
4.2	实时图片的获取和分类	18
4.3	定位算法的设计	19
4.4	构造导航信息	21
4.5	本章小结	23
第 5 章	测试和评估	24
5.1	布署和实验	24
5.1.1	在不同性能手机上的实验结果	24
5.1.2	不同路段的实验结果和对比	25
5.2	分析实验结果	26
5.3	系统优化	28
5.3	本章小结	30
第 6 章	总结和展望	31
6.1	总结	31
6.2	展望	31
致谢	32
参考文献	33

第 1 章 绪论

1.1 研究背景和研究意义

近年来, LBS 服务从研究到商用都经历着快速而蓬勃的发展。市面上已经有多款成熟的商用 LBS 手机软件可供选择, 如百度地图。针对室外开放环境, 他们依靠便捷的 GPS 信号或网络基站对用户提供的最高可以达到米级别精度的定位或导航服务, 从而可以实现将用户准确地从 A 行政区的一个家属院带到 B 行政区的一个大型商场这种导航服务;

与成熟发展和商用的室外定位和导航服务形成鲜明对比的是, 对于室内环境的定位和导航问题虽然已有不少研究工作或解决方案, 但多数方案在布署成本, 服务精度, 系统的稳定性等方面的表现都存在比较多的问题。可以说, 对于室内导航问题, 目前仍缺少优秀的解决方案。

然而与此同时, 我们在日常生活中对与室内定位和导航服务的需求却不断增加。据调查, 进入 21 世纪之后, 大众接近 80% 的时间处于室内, 且其中超过 60% 的信息与位置有关^[1]。因此, 作为一个与我们的日常出行活动息息相关的重要问题, 室内定位和导航亟待一套成熟可靠的解决方案。

与开放环境的定位和导航系统的相比, 室内导航系统的研究和布署完全不同。由于 GPS 信号在城市大型建筑内部的严重衰减, 精细的室内地图构建难度大, 以及高昂的布署成本等原因, 我们需要研究并设计一套全新的解决方案。该方案从数据的采集, 到地图的构建, 再到定位和导航方式都需要针对室内环境量身定做, 以达到精度和成本等方面的要求。

1.2 研究目标

越来越多的大型商超、机场等公共室内区域的建设和开放, 带来了越来越多的室内定位和导航需求, 与之对立的是优秀的解决方案的缺乏。

本文以解决室内导航难题为研究目标, 以高精度的定位方式、精确的室内地图、准确的航向推算, 以及低成本和方便操作的系统布署方法为研究重点, 探索并提供一套简洁、高效、并且可靠的完整解决方案。

1.3 研究现状及其缺点

为了解决室内导航的难题, 国内外也已经有很多研究做了相当多的相关工作。总

体来看, 这些方案多将研究重点集中在对目标的定位和导航方式、数据的采集和室内地图的构建、以及导航系统架构的研究等方面, 并取得了一些不错的效果。但这些方案都不可避免的存在一些精度不够、系统对不同环境的适应性低、过高的布署成本、以及系统稳定性低、和计算延迟过高等方面的缺点, 如下所述。

1.3.1 定位和导航方式的研究

定位和导航方式的研究在各个方案中总是研究重点, 多种多样的解决思路也被提出, 比如利用成熟发展并被广泛部署的 Wi-Fi 网络产生的 Wi-Fi 指纹, 或更加普遍地存在于周围环境中的地磁指纹^[2-13]等可以记录目标位置信息的数据, 配合手机陀螺仪记录的惯性单元等辅助数据为用户进行定位。但这些方案对各种传感数据的依赖也带来了抗干扰能力差和对不同环境的适应度低等缺陷。

如 Travi-Navi^[2] 系统, 一种使用视觉图像引导目标的室内导航系统, 布署时通过智能手机记录 Wi-Fi AP、步幅、地磁以及路径的高清图片等信息来产生一条参考路径, 在用户行进时将用户手机实时采集的 Wi-Fi 指纹和地磁指纹与参考路径中的相关信息进行比对, 并估算其步幅从而为用户提供在该段路径上的定位, 以及使用图片信息进行航向引导。

但多数方案中依据各种传感器数据进行位置记录的方案总是存在精度以及系统对不同环境的适应性较差等方面的问题。以 Travi-Navi 为例, 我们知道 Wi-Fi AP 是随时间变化的, 而对用户的步幅只能进行估算, 与个体有较强的相关性, 这些问题都在很大程度上降低了定位和导航的精度。此外地磁在大型建筑内部还会出现较大扰动, 这很大地降低了系统的抗干扰性。此外, 系统所依赖的基础设施 Wi-Fi 网络会大幅度限制系统的使用场景。

1.3.2 地图构建方式的研究

精确的室内地图对于一款优秀的室内导航系统同样相当重要, 怎样去高效地采集数据并进行地图的构建是许多方案的研究重点, 如多数方案中采用的“众包”的方式^[14-19]。这些方案在一定程度上解决了数据采集和地图构建的问题, 但他们在数据质量、成本、以及持续性方面的表现都很难让人满意。

劳动密集型的众包方式在数据的采集、地图的构建以及系统的布署过程中相当常见, 如 Poster^[8], 一种基于众包和多传感器的室内-室外地铁站导航系统。通过一些激励方式让地铁站等室内区域的用户采集和上传路径信息, 系统构建地图后为用户提供定位和导航。然而, 众包机制存在一些比较明显的问题, 如怎样建立持续有效的激励机

制？以及众包必然会带来的不同来源的数据之间的质量差别。这些问题对于系统的持续推广，系统精度，以及系统的鲁棒性等方面会产生很大的负面影响。

1.3.3 导航系统架构的研究

对导航系统另外一个相当重要的研究点在于系统的稳定性和响应速度，怎样在确保精度的情况下提升系统的稳定性和响应速度值得探讨。已有的研究中对系统的架构有三种选型：服务器负责运算，客户端负责交互的 C/S 架构；基于云服务的用户之间进行路径共享的分布式架构；以及客户端负责全部运算和交互的架构^[20-25]。

服务器负责运算，客户端进行交互的架构带来了明显的延迟问题（包括数据传输的延迟和计算的延迟），并且降低了系统的稳定性。而将运算和交互全部交由客户端来处理的架构同样面临很严重的问题，如目前智能手机的计算能力依然相当有限，无法很好的进行快速有效的计算。这些问题都限制了系统的推广和部署。

除了以上方案，还有一些研究着重探索一些特殊的定位和导航方式，如使用声波等物理信息，或从算法的角度进行优化和尝试^[25-27]。

1.4 研究挑战和解决措施

室内导航服务研究的关键和难点在于高精度，包括对目标的精确定位，和布署区域高精度地图的构建。精确的定位确保我们可以随时获知目标在建筑内部的位置，而高精度的地图协助定位结果对目标进行航向推断。此外，如何在确保高精度的同时提高系统的响应速度和稳定性、增强系统的抗干扰能力和对不同环境的适应能力、以及如何有效降低系统布署成本等都是本文的研究所面临的挑战。

通过对已有研究和它们的缺陷的探讨，以及对研究面临的难点和挑战的分析，在 DeepNavi 中我们开创性地提出并设计完全依赖于图像信息的定位和地图构建方式。我们通过使用 CNN 对用户实时位置图像的分类来为用户做出精确定位，使用目标路径的图像信息构建一份精确的室内地图，和使用这些图像信息训练神经网络。完全依赖于图像信息的做法也极大的提高了系统对于不同环境的适应能力。而布署系统所需的图像信息的采集成本很低，并且采集方式相当简单。

1.4.1 高精度的定位方式

与 Travi-Navi 等基于 Wi-Fi、地磁等传感数据对目标进行定位和导航的方案不同，本文提出的基于 CNN 的室内导航系统 DeepNavi 不依赖于其他基础设施和传感数据。DeepNavi 对目标的定位完全依靠神经网络对图像信息的分类，因此对部署环境几乎无特殊设施要求。而从神经网络对用户实时位置图像的分类到用户位置的精确定位，本

文提出了一些高效的相关算法来进行计算，并且后期的测试结果显示 DeepNavi 的定位功能可以达到相当高的精度。

1.4.2 数据采集和高精地图构建

在 DeepNavi 系统的地图构建部分，同样完全基于目标路径的图像信息，而且对目标路径关键点的划分与领航员或用户的行为无关，而采用固定的时间长度作为单位，这杜绝了使用步幅等数据的方式带来的个体上的差异，进而导致的地图精度的损失。因此 DeepNavi 数据采集的关键也在于如何获取高质量的路径图像。

在本文提出的基于 CNN 的室内导航系统 DeepNavi 中，我们使用“自激励”的方式来完成数据的采集工作。所谓“自激励”是指数据的采集由目标 POI 的所有者完成。比如大型商场内部的一家奶茶店的店主，为了向更多对他的店铺感兴趣却在商场内迷路的潜在客户销售自己的产品，该店主会主动使用我们的系统提供的简单易用的领航员模块采集从商场入口到目标 POI 之间的一段路径来进行系统在这段区域的部署。可想而知的是，店主必然是相当熟悉从入口到目标 POI 之间的这段路径的，并且因为其中的相关利益，在数据采集的过程中其会保持较高的精度。从而，我们也能确保系统构建的地图的精度。

1.4.3 针对室内环境高动态性的解决方法

室内环境的动态性是几乎所有对室内导航的研究都需要考虑的一个难题，DeepNavi 支持对建筑内小部分区域甚至单独一段路径独立布署和使用，即插即用，灵活性高，相当好的满足了临时场景内的导航需求。

与一些方案需要构建目标区域内一整层或多层，甚至整个区域的地图文件不同的是，本文提出的基于 CNN 的室内导航系统可以实现多个路段相互独立。不需要完成对整个区域的数据采集和部署，即插即用，尽管一处建筑内会只构建一个模型和地图。一个领航员完成入口和与自己利益相关的那个目标 POI 之间这段路径的数据采集之后即可部署这段路径供相关用户使用，这也与实际情况相符，即大型商超中多为店主身份的领航员只会主动部署与自己的利益有关的那段路径。

而且，在系统中，更多的路径的采集和部署对系统在该区域的精度提升更加有利，这体现在更多的数据可以对该区域的模型进行更好的训练，以及更大的地图和更多的位置点为寻找最优路径提供了可能性。

1.4.4 快速响应、稳定的系统架构

在对已有研究的缺点的探讨中我们发现，一个优秀的室内导航系统需要在保证精

度的同时实现快速的响应和稳定的服务。这要求我们采用一个合理而优秀的系统架构。DeepNavi 采用 C/S 架构，于传统 C/S 架构不同的是，我们将数据的计算集成在客户端，服务器只负责对模型和地图的下发，这样同时保证了服务的快速响应，也确保了系统的稳定性。

1.5 动机

一个值得思考的问题是，即然已经有相当多关于室内导航系统的研究和解决方案，我们为什么还要设计并提出 DeepNavi？以下几点是我们的考量：

- 依赖 Wi-Fi AP、网络基站等基础设施，和地磁指纹等复杂传感器数据的系统并不能覆盖所有应用场景，而 DeepNavi 不需要任何额外的基础设施支持，这极大地扩展了系统的适应范围，有效的提高了系统的鲁棒性。
- DeepNavi “自激励”的主动部署方式避免了多数系统采用的众包方式难以长期激励推动部署的问题。并且因利益相关，领航员在进行数据采集时总能保证数据的高质量，从而确保了 DeepNavi 服务的高精度。
- 即插即用，不用完全采集和部署整个区域也能出色工作，很好地适应了临时场景。而且系统所提供服务的精度还会随着区域内部署路径的增多而不断提高。值得一提的是，自激励的主动部署方式和对单条路径布署的支持无形中满足了对室内环境的数据采集时无法绕过的隐私问题。

1.6 内容安排

本文的其余部分组织如下。我们首先在第 2 章阐述了系统的总体设计概述。接着，在 3 章中我们详细讨论了地图的构建和模型的训练，此外还对包括数据的采集和标注方式，以及在图片的前期处理中非常重要的对照片中行人的擦除等内容的介绍。然后我们在第 4 章详细介绍了如何通过领航员模块中构建的地图和 CNN 进行定位和导航工作。包括对其中一些关键算法的解释。再然后我们将在第 5 章中对 DeepNavi 系统的实际表现进行深入的讨论，包括定位精度以及在实验过程中系统表现出的一些问题的思考。接下来我们在第 6 章中对本文进行了总结和展望。最后我们在第 6 章中总结本文的研究并在致谢后结束本文。

第 2 章 系统概述

本文提出并设计实现 DeepNavi，一种“基于卷积神经网络（CNN）的室内导航系统”。DeepNavi 不依赖于基础设施和传感数据，完全使用路径图像信息进行高精度室内地图构建和神经网络训练，神经网络通过对用实时位置图片的分类进行精确定位，系统根据定位结果和地图为用户做出准确的航向推断。DeepNavi 布署成本低，布署方式简单高效，服务稳定。同时系统在抗干扰能力和对不同环境的适应能力等方面均表现出色。

本章将从对室内导航系统的需求分析、针对相应需求的关键技术的设计、系统的总体架构概述、以及系统的部署和使用方法几个方面对 DeepNavi 的系统概况进行介绍。

2.1 需求分析

为了在建筑物内部给用户提供精确的室内导航服务，我们首先需要解决精确的室内地图构建、高精度的定位、准确的航向推算等几个最重要的问题。此外，还要求我们在系统的布署成本和布署时间、以及系统的稳定性等方面给出优秀的解决方案。

2.2 关键技术设计

2.2.1 室内地图的构建

室内地图的构建是为了精确的定位和导航工作做准备。我们对 DeepNavi 的精度设计为 1 米范围以内，因此需要将目标区域划分不大于 1 米的子区域。在定位和导航阶段将以这些子区域为运算单位对用户的位置的进行定位，以及根据定位的结果，对用户需要进行转向（包括台阶等）的位置进行提示。因此，在地图中我们也将主要记录这些子区域，以及子区域之间的连接关系。

这样的子区域在 DeepNavi 中被称为位置点。以一条 10 米长的走廊为例，我们将这段走廊以 1 米的长度进行划分，这段走廊将被划分为 10 个长 1 米的子区域，每个子区域即为一个位置点。

在 DeepNavi 中，位置点划分的具体做法为：对系统布署时领航员采集的目标路径的一段视频数据（数据的采集和标注的内容详见 3-2 节）以 1 秒为单位进行切分。因为这段视频数据记录着目标路径从起点到终点的图像信息，并且综合考虑成年人的平均步长和步行速度^{[28][29]}，切分后的每段时长 1 秒的子段视频记录着一段长为 1 米左右

的路径的图像信息。因此我们以 1 秒为单位对这段视频进行切分。从而，目标路径将被划分为长度为 1 米左右的子区域，这个区域将在地图中对应一个位置点。之所以要使用图像信息来对目标路径进行位置点的划分，而不是像其他研究中的做法中对目标路径直接划分，是因为我们在定位的工作中需要使用到每一个位置点的图像信息进行模型的训练。对视频进行划分的做法可以产生一个“目标路径 – 地图 – 模型输出类别”之间的映射。

两个位置点之间将存在直接相邻、和经相邻位置点可达的非直接相邻的关系，地图将被用以记录这些位置点，和位置点之间的关系。为了表达方便，我们对每一个位置点进行编号，记为一个 `placeId`，每一个位置点将对应一个唯一的 `placeId`。而地图中对每一个位置点的描述也将用 `placeId` 来表达。为了实现系统时编码操作的方便，地图是一个 `txt` 格式的文本文件。地图对这些位置点以及位置点之间的关系的记录方法为：以一行来描述一个位置点，在行首声明这个 `placeId`，接着在这一行的后面记录与这个位置点直接相邻的位置点的 `placeId`。这些直接相邻包括两种：按照从起点到终点为方向，在这个位置点之前的相邻位置点，和在这个位置点之后的相邻位置点。一个位置点可以包含一个到多个这两种相邻位置点。在地图中，都会给出相应的记录方法（地图的详细格式见 3.3 小节）。

我们之所以到定义并在地图中记录这些位置点之间的相邻关系，是为了在导航工作中进行航向的查询和推测，以及路径上需要关注的位置点的获取。

2.2.2 DeepNavi 的定位原理

精确的定位是成功导航的前提，因此，高精度的定位在室内导航系统中极其重要。为了探究和设计一种高精度的定位方式，我们详细了解了多个方案。除了通过使用 Wi-Fi 指纹等可记录用户位置变化的数据来定位用户的方案之外，基于图像或视觉引导的方案、和使用机器学习的技术来提高系统表现的一些尝试^[21]给了我们启发。基于图像或视觉的引导相较于文字提示，对用户而言更加直观和容易被理解。而多个使用到机器学习的方案中，神经网络等技术的加入明显提高了系统在精度等方面的表现。

如果说的通俗一些的话，定位其实就是对用户当前位置的识别。考虑到卷积神经网络在图像识别方面的先天优势^[29-33]，并结合我们期望使用视觉引导这种基于图像的方式来进行导航提示，我们在 DeepNavi 中使用卷积神经网络来作为定位的工作的核心。

为了对用户的位置进行识别，我们首先需要设计训练一个神经网络。首先来看神

经网络的输入和输出。显然，这个神经网络的输入将会是对用户进行导航时用户所处位置的实时图片，而输出将会是这张输入图片与目标路径上的每一个位置点的匹配度，即置信度。这不难理解，为了分辨用户当前的位置，我们需要将用户当前的位置和目标路径上每一个位置点进行匹配，以找出最符合的那个位置点，即这个位置点就是用户当前所在的位置。这个匹配的过程在神经网络的工作中即对输入图片的分类。

这张图片将会由 DeepNavi 的客户端进行获取。在导航部分的工作中，对实时图片进行格式化、裁剪等工作后，将被输入给神经网络。在一个被标记为 N 个位置点的区域中，神经网络的输出就是输入图片在这 N 个位置点的置信度。因此，这个神经网络将会含有一个 1 输入节点和 N 个输出层节点。

为了训练这个包含 N 个输出层节点的神经网络，我们需要对应的训练数据集。我们在 2.2.1 中详细阐述了通过对目标路径视频图像的划分来将目标路径划分为位置点，并由此形成一个从“目标路径 – 地图 – 模型输出类别”的映射的工作。在此，我们进一步将这个映射描述为“位置点 – 神经网络输出节点”之间的映射。显然，每一个位置点在神经网络中将会对应着一个输出节点，即类别。因此，我们需要为每一个类别创建一份训练数据集，总共 N 个类别。通过分析这种映射关系，我们还可以发现：在 2.2.1 中，我们对目标路径的这一段图像以 1 秒进行切分后，产生的 N 个时长为 1 秒的子视频段正好记录着这个位置点的图像信息。因此这个长 1 秒的视频分段便成了最好的训练数据来源。于是，我们的训练集将从这些时长 1 秒的视频段中产生。

因为神经网络的训练集应该是多张离散的图片，所以我们需要对这段 1 秒的视频片段进一步处理，以产生多张离散的图片。这个片段可记录的路径长度为 1 米左右，此外领航员采集的视频帧率为 60FPS，因此我们在这个片段中以 1/30 秒为间隔从视频中抽出共 30 帧图片，这 30 帧图片将会作为这个类别（对应的位置点）的训练集。从而，我们将使用目标路径的这段视频构造出一个 $N * 30$ 的训练集。

当然，神经网络的分类结果并不可以直接作为定位的最终结果，因为最高的置信度和正确的位置之间存在偏差（在 4.2 节中有对数据的详细分析）。因此，我们还会设计一个类似“聚类”的算法来对原始的分类结果进行运算，从而计算出最终的定位结果。

2.3 部署方法和使用方式

DeepNavi 部署方式相当简单。系统部署者只需要采集一段目标区域或目标路径的视频信息，系统便会主动完成高精度室内地图构建和神经网络的训练，从而完成部署。

我们提供一个运行于 Android 智能手机上导航应用，有导航需求的用户进入已部署 DeepNavi 的区域后应用会下载该区域的地图文件和模型文集，行进中导航应用会对用户实时位置进行定位和推算用户航向，并在应用中对用户给出定位和导航结果。在第 4 章中给出了客户端的使用截图。

2.4 系统架构概述

在 DeepNavi 中有“用户”和“领航员”两种角色。用户顾名思义为有导航需求的人；领航员是目标路径的采集者和部署者，如某大型商场里的一间店铺的店主，其熟知从商场入口到自己的店铺之间的一条或多条路径。因此系统也由供“领航员”所使用的负责前期数据采集和标注、模型训练以及构建地图的“领航员模块”，和供具有定位和导航需求的用户所使用的“导航模块”两部分组成。

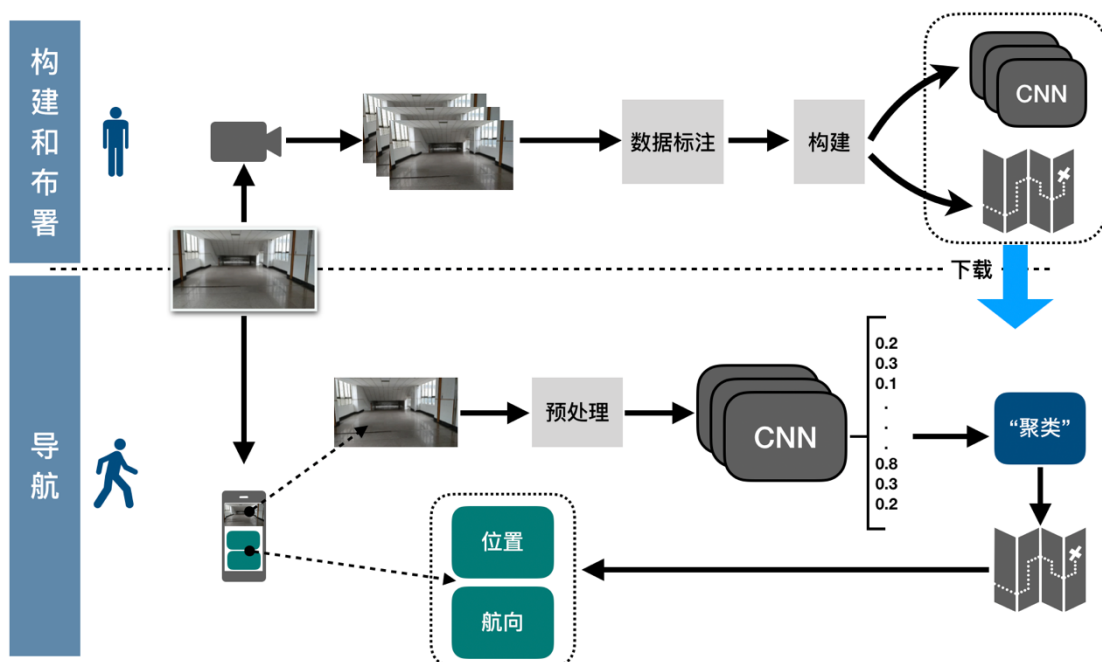


图 2-1 系统基本架构图

系统的基本架构如图 2-1 所示。在领航员模块中，领航员通过手机摄像头记录一段从起点（通常为一个建筑物入口）到目标 POI 之间的一条或多条路径的视频，这一步为数据的采集。完成数据的采集之后，系统会对数据进行标注，大致做法为，将这段视频以一秒为单位切分为 N 个等长的视频分量，每个分量记录一段长约一米的路径分量，于是这段路径也被划分为 N 个等长的分量，每个分量记为一个位置点。将目标路径划分为 N 个位置点之后，系统将构建一份特殊的地图文件来记录这些位置点之间的相邻和可达关系。接着，系统开始神经网络的训练，训练数据集来自 N 个视频分量。

领航员模块完成数据的采集和标注、模型的训练以及地图的构建之后便完成了对这段路径的部署。当对此段路径有导航需求的用户进入商场之后，安装并打开导航 App（即用户模块），下载这个商场的模型和地图文件，选中自己的起终点之后 App 开始对用户进行导航。导航 App 工作时会通过手机摄像头采集用户实时位置的图像信息，该图像信息会被输入 CNN 模型进行分类，CNN 输出该图像在 N 个位置的置信度，对置信度进行特殊运算后确定用户的位置，进而结合地图数据为用户给出导航建议，从而实现对用户的定位和导航。

2.5 本章小结

本章，我们从地图的构建、和定位模块的关键技术的设计、系统的布署和使用方式、以及系统架构几个方面对 DeepNavi 系统进行了概括和介绍。

第 3 章 系统的构建

在 DeepNavi 系统中,我们通过采集目标区域或目标路径的图像信息来进行系统的构建。构建工作由图像数据的采集、对图像数据的标注、精确的室内地图的构建以及神经网络的训练等几部分组成,总体流程如图 3-1 所示。

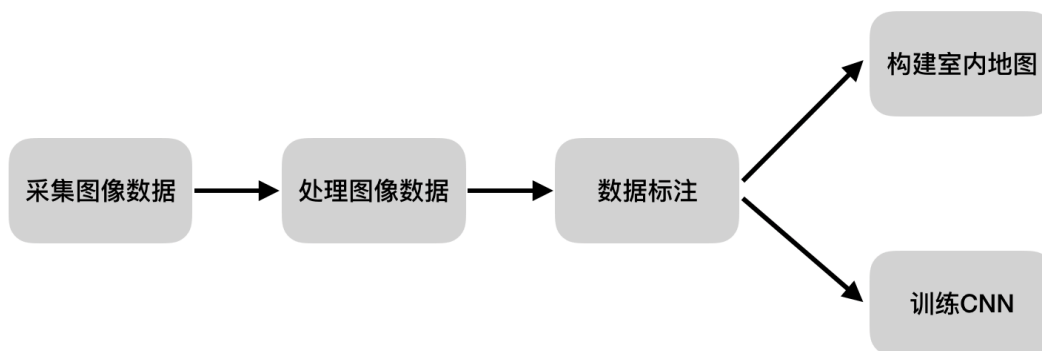


图 3-1 系统构建工作流程图

3.1 构建工作概述

在 DeepNavi 系统中,对目标的定位和导航依赖于 CNN 模型和地图文件。训练模型和构建地图所需的数据均来自领航员采集的目标路段的数据。在此对这一部分工作进行概述,详细做法在本章中分小节给出。首先需要采集数据,领航员从起点向目标位置行进时录制前方图像,完成后录制对这段视频以一秒为单位切分,即我们对这段路径以一秒记录的位移长度为单位切分成 N 位置点,如图 3-2 所示,离散的位置点是我们使用 CNN 进行定位的基础。将目标路径切分成离散的位置点之后系统根据位置点之间的相邻关系构建一份特殊的地图。另一方面, CNN 模型是我们整个系统的关键。在本系统中, CNN 拥有 1 输入 N 输出,即接收一帧实时位置的图片作为输入,输出其匹配 N 个位置点的置信度列表。同样的,领航员所采集的图像数据也将用于模型的训练。

3.2 数据采集和标注

为了进行 CNN 的训练和导航阶段为用户提供的图像化提示信息,我们在领航员模块中采集目标路径的图像信息。具体做法为:确定一条或多条从起点(通常为建筑物入口)到终点(目标 POI 位置)之间的路径,领航员从起点出发,持手机(手机摄像头朝向前进方向,横向握持,屏幕垂直于水平方向)匀速前进并全程开启摄像头来录

制，到达终点时结束录制。完成录制后，领航员会在供领航员使用的 App 中标明这段路径所在的建筑物信息以及起点和终点，之后这些信息会同这段视频一同上传至被称为“构建服务器”的特殊服务器进行后续的数据标注和模型训练以及地图构建等工作。

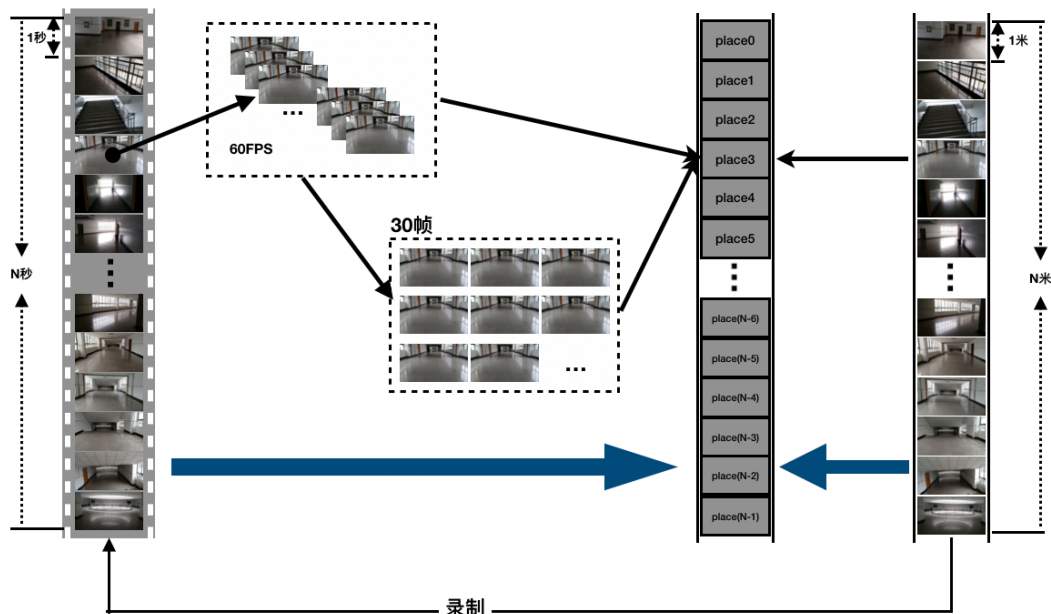


图 3-2 路径、视频和位置点之间的映射关系示意图

服务器接收到该视频之后首先开始数据的标注工作。为了后续的定位工作中对目标处于这段路径上什么位置进行确认，我们需要将这段路径分为多个离散的短距离分量，后续的定位工作中，系统会对处于该分量内部的所有位置确定为该分量所在的位置点。将这些位置点映射到这段记录有目标路径图像信息的连续视频上，我们需要对这段视频进行切分。考虑到小范围内相邻位置图像信息的相似性、成年人的平均步行速度以成年人平均步长之间的关系，我们以 1 秒为单位对这段视频进行切分，切分获得的每个离散的视频分量记录的路径长度在 1 米左右，也就是说，目标路径被切分为多个长度为一米左右的短距离分量。路径、视频和位置点之间的映射关系如图 3-2 所示。

在图 3-3 中我们展示了一个模拟的布署区域，该区域包含一个入口 S 和两个目标位置点 T1 和 T2，从 S 分别到两个目标位置点之间有两条路径，且有一部分重合。在该区域进行位置点划分时，我们通过不同颜色的矩形来表述一个由一个时长为 1 秒的视频分量记录的距离分量，一个距离分量将会代表该段路经上的一个位置点，如图中 A, B, c1 等。以 A 位置点为例，在后续的定位工作中，处于 A 矩形区域范围内的

位置都将会被确定为 A 位置。在系统和地图文件中以一个五位的格式化 placeId 来描述一个位置点，对地图的详细介绍详细描述在 3.3 节。

在对 DeepNavi 系统最后的实际部署中，我们以中南大学校内某一七层教学楼一楼入口至四楼卫生间之间的路段为目标路径，在该次部署中，领航员采集到的视频数据时长为 285 秒，因此，这段视频被切分为 285 小段，即该段路径上包含 285 个位置点。

3.3 地图的构建

地图的构建工作建立在对目标路段划分为一个个位置点的工作之上。考虑到首先定位部分的输出为用户所在位置点的 placeId，意味着我们需要根据这个 placeId 来为用户给出导航建议，其次，考虑到在实际的室内导航情境中我们需要为用户展示的信息主要有以下几种：导航行程的起终点位置，用户当前所在位置以及下一个需要转向的位置。并且总体来看，任何一段路径也基本这些关键位置点串联而成。因此在“基于 CNN 的室内导航系统”中，我们的地图文件的主要内容便是对这些位置点之间的邻接关系和可达性的记录（这两种关系的定义在下面给出）。

具体来说，这是一份 txt 格式的文本文件（便于我们在编码时写入和读取），它以每一行为一个记录单元，一个记录单元中描述一个位置点的全部信息，如实例地图文件所示，我们以描述图 3-3 中从入口 S 到目标位置 T1 这段路径的示例地图文件中的第 0 行、第 1 行和第 2 行内容为例对每其中的变量进行说明：

- **[0000A]:** 0000A 是 A 位置的 placeId，是一个固定格式的字符串。每一行初始通过[placeId]的格式来声明这一个单元所描述的位置点。
- **fm:** 整型。若从位置点 P 前进一个位置点可达位置点 Q，则 P 记为 Q 的一个 fromNode，Q 位置点的 fm 值加 1。并且在 fm 值后面记录 P 的 placeId，如示例地图文件地 1 行，B 有一个 fromNode，placeId 为 0000A。
- **tm:** 整型。若从位置点 P 前进一个位置点可达位置点 Q，则 Q 记为 P 的一个 toNode，P 位置点的 to 值加 1。并且在 tm 值后面记录 Q 的 placeId，如示例地图文件第 0 行，A 有一个 toNode，placeId 为 0000B。
- **sm:** 整型。如位置点 P 有 n 个可选前进方向， $n > 1$ （即 P 为岔路口），则每个方向的下一个位置点均作为一个 P 位置点的 switchNode，P 的记录单元中 sm 值为 n，在 sm 后面记录每一个 switchNode 的 placeId，如示例地图文件第 2 行中，由于图 2-1 中 C 位置为两个可选前进方向的岔路口，因此将其分为 c1，c2，c3 三个子位置点，c1 有两个 switchNodes，其 placeId 分别

为 0000c3 和 0000c2。

如前文所述，该地图文件用以记录目标路径上的位置点之间的相邻关系和连续性。相邻关系是一种特殊的连续性关系。在地图中，若两个位置点之间存在 fromNode, toNode, 或 switchNode 关系，则我们定义这两个位置点互相相邻。从对路径的划分来看，两个相邻的位置点所在的矩形区域相接。位于同一条路径上的两个不相邻位置点之间互有连续性，也记为这两个位置点互相可达，如图 3-3 中，位置点 A 和位置点 B 之间互为相邻位置点。位置点 A 和位置点 D 互相可达。在实际部署中与示例地图稍有不同的是，实际的部署中构建的地图文件中位置点使用数字而非英文字母来命名，如在对科教南楼的部署中，从起点到终点的 285 个位置点分别会被命名为 00000, 00001, 00002, ... , 00282, 00283, 00284。

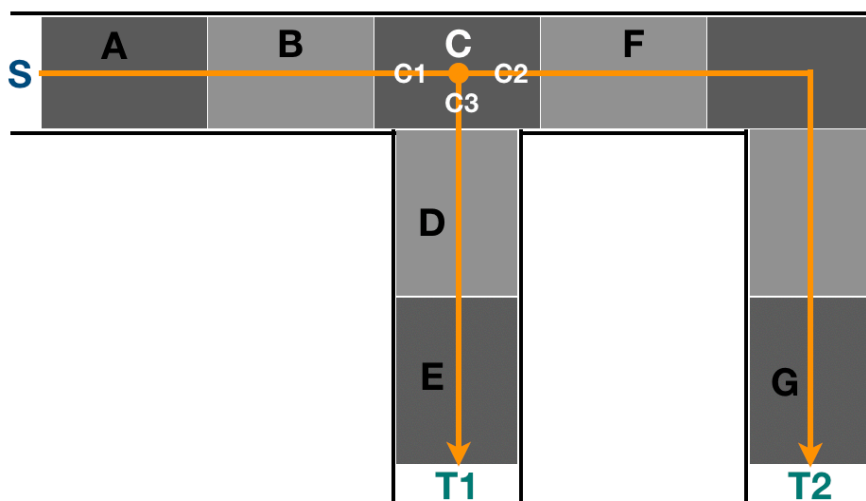


图 3-3 对示例区域路径分段

```
[0000A]:fm=0,tm=1,0000B,sm=0\n
[0000B]:fm=1,0000A,tm=1,0000c1,sm=0\n
[0000c1]:fm=1,0000B,tm=0,sm=2,
          0000c3,0000c2\n
[0000c3]:fm=1,0000c1,tm=1,0000D,sm=0\n
[0000D]:fm=1,0000c3,tm=1,0000E,sm=0\n
[0000E]:fm=1,0000D,tm=0,sm=0\n
```

图 3-4 示例地图文件

3.4 模型和 label

如本章开头所述，“基于 CNN 的室内导航系统”使用领航员采集并标注的目标路

径的图像信息作为数据集来训练神经网络模型。本节主要讲述从领航员采集并标注的数据中产生数据集的方法以及模型的概况。在 2.1 节数据采集和标注中对录制的目标路径的一段长度为 N 的连续视频以 1 秒为单位切分为了 N 个等长为 1 秒的视频分量，该视频分量映射到目标路径上后目标路径被分为 N 个长 1 米左右的短距离路径分量，于是产生“视频分量 – 短距离路径分量 – 位置点”之间的映射关系。又因为在定位工作中我们需要获得目标位置于路径上每一个位置点的匹配程度（即分类的置信度），因此我们以每一个视频分量中的图像数据作为神经网络模型每一个类别（相应位置点）的训练数据集来源，如图 2-1 中左半部分的对视频分割的示意图。

视频分量是一段时长 1 秒的连续视频，而训练集显然应是一张张照片组成，因此我们采用隔等长时间间隔后从视频中抽取一帧的方式来产生每一个训练集。考虑到领航员采集数据时录制视频的采样频率为 60FPS，我们以 1/30 秒为抽帧时间间隔。于是，在模型训练中我们拥有 N 个类别，每个类别拥有一个样本量为 30 的训练集。完成以上工作之后我们便可以开始神经网络的模型设计和模型训练工作了。

这个神经网络输入层为一个节点，输出层为 N 个节点，即它接受一张图片作为输入，输出该图片在 N 个类别下的置信度列表。另外，为了后续定位工作的顺利开展，此处我们还需要创建一份 label 文件，它同样是一份 txt 格式的文件，每一行记述一个 placeId，从第一行到最后一行分别为 00000~0000N。

在 4.3 节定位部分的工作中我们会通过一些特殊的算法对这个置信度列表进行运算，最终给出对目标的定位。同样，以我们在科教南楼的实际部署为例，对目标路径录制的时长为 285 秒的视频被切分为了 285 个视频分量，即目标路径被分为了 285 个关键点，因此我们构造 285 个样本大小为 30 的训练集，设计并训练了一个输入层为一个节点，输出层为 285 个节点的神经网络。在定位工作中，摄像头采集的一张实时图片输入到这个卷积神经网络，处理后输出该图片处于 285 个位置点每一个点的概率，详细内容在 4.2 节进行论述。

3.5 图片中行人的擦除

一个常见的现象是，大型公共区域内总是伴随着络绎不绝的行人，如大型商超、机场等，不难想象的是在大量行人的存在会对如“基于 CNN 的室内导航系统”这种依靠对室内环境图像信息的分类来确认用户所处位置的定位或导航系统的定位工作产生干扰。我们的实验数据也证实了这一点（对测试和实验的详细记述在第 5 章），但是出于系统稳定性等方面的考量，我们的系统最终实现并没有包含对行人的影响的消除这

部分功能（后文会解释放弃这部分功能的原因）。但与在早期设计阶段的我们实现的一个包含这部分功能的客户端版本的实验结果相比，在实时图像中出现行人时当前版本客户端的定位准确性会有下降，但这种下降并不严重。

对图片中行人的擦除具体的做法可以概括为：“识别-消除”。首先，通过在系统中部署另一个神经网络模型（早期实现版本适用 TensorFlow 官方示例中的 Object Detector 模块所用的神经网络）来对图片中行人进行检测和标识。系统工作时，手机采集到的一帧当前位置的图片首先被输入给 Object Detector，该神经网络会检测图片中的行人位置并输出这幅图片中一个或多个行人的位置坐标，接着我们再将该图片输入给一个的图片处理工具（另一个神经网络），该工具会将行人所在的矩形区域从该图片中抠除，然后以矩形区域边缘的色彩为基准对图片进行补全。接着，擦除掉行人之后的图片将被输入给我们在 3.3 节中训练好的神经网络进行分类，即开始定位工作。

我们曾在对 DeepNavi 在 Android 智能手机上进行实现的一个早期 debug 版本中加入了对于行人的擦除这一部分功能，但遗憾的是，一旦进入对图片中行人的抠除和补全工作，手机就会开始出现明显且较为严重的卡顿和异常的发热。优化无果，只好暂时放弃这个功能。而且第 5 节的实验结果显示与包含该功能的实现相比定位准确率虽有下降，但并不是非常大。

3.6 本章小结

我们在本章以地图和模型的构建为重点详细介绍了 DeepNavi 系统的数据采集和标注方式。数据采集来自于领航员对目标路径视频的录制，数据的标注关键在于对视频的切分，以及视频 - 位置点 - 路径之间的映射关系，如图 3-2 中所示。地图的构建部分我们以示例区域的地图为例对地图进行了详细介绍。而模型的构建部分我们主要描述了训练数据集的获取方式和网络的输入输出层。以及本章最后我们阐述了怎样对图片中的行人进行擦除。

第 4 章 定位和导航

在 DeepNavi 中，系统使用领航员对目标区域进行布署时构建的模型和地图文件来对用户进行定位和导航。定位和导航部分的大致工作流程为，客户端实时采集用户当前位置的图片信息，对图片进行裁剪和格式化等预处理之后输入神经网络模型；模型对图片进行分类，接着系统通过一些算法对分类结果进行运算后为用户做出精确的定位；然后系统根据定位结果查询地图后为用户给出航向推断，最后在客户端中对用户进行导航提示。定位和导航的流程如图 4-1 所示。

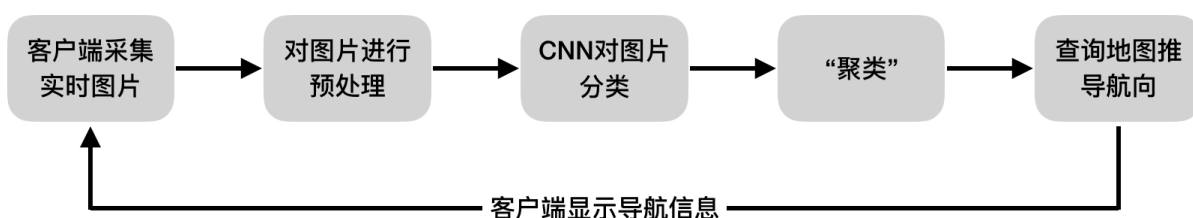


图 4-1 导航和定位阶段工作流程图

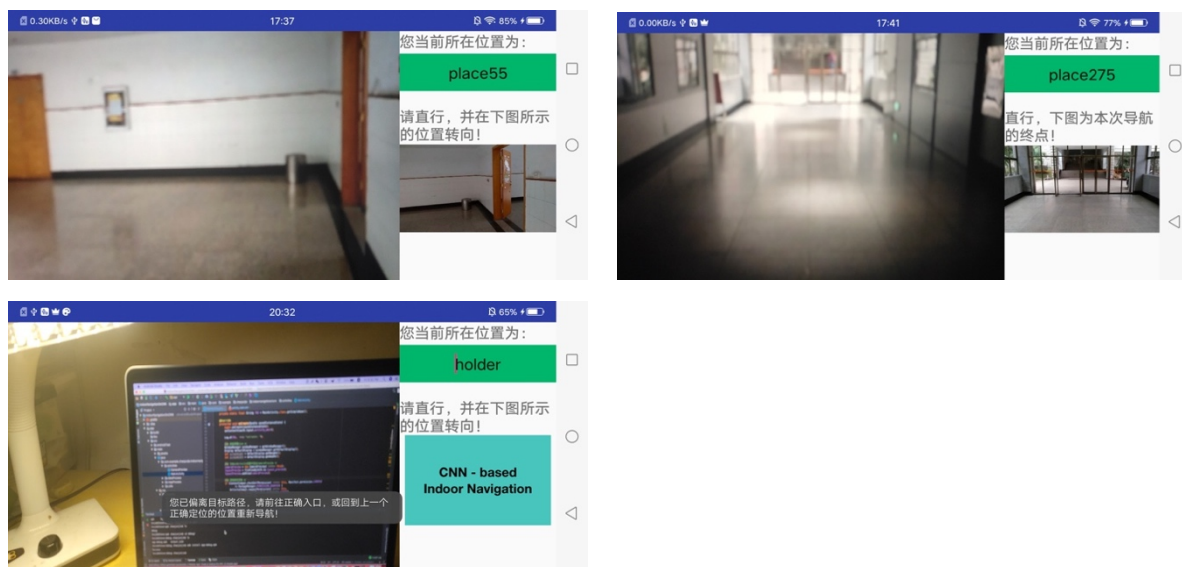


图 4-2 导航 App 运行时截图

4.1 DeepNavi 定位和导航概述

完成第 3 章中所述的数据采集和标注、模型训练以及地图的构建后，我们就可以开展定位和导航工作了。定位主要依靠模型文件对手机摄像头实时采集的每一帧图片进行运算来给出目标所在的位置，导航依靠定位工作确定的目标位置，配合地图来查

询目标应该关注的关键位置点信息，从而为目标提供导航建议。如图 4-2 所示为导航 App 运行状态下的界面截图，App 中左侧会展示相机对当前所处位置的实时预览画面，右侧会为用户展示当前所在位置的 placeId，和需要用户关注的下一个转向点位置的图像信息。需要指出的是，第一幅图中的转向点位置的图像信息会取自 3.4 节中从领航员采集的视频信息中产生的该类别的训练集中的一帧图片。定位和导航部分详细阐述如下。

4.2 实时图片的获取和分类

导航 App 运行时截图如图 4-2 所示。导航过程中，用户横向握持手机，保持手机主摄像头朝向前进方向且屏幕垂直于水平方向后以正常步幅匀速步行前进。手机摄像头会全程开启并在界面上显示用户所在位置的实时图像预览，用于对用户定位（对实时图像的分类）的输入图片便来自预览视频。应用中调用 Android 手机的摄像头接口以获取这些视频的输出字节流，我们在固定的时间间隔（1 秒）截取一帧图片的字节流数据将其压缩成 JPEG 格式的图片文件，因为模型文件可以接受的输入数据类型为 Bitmap，因此我们再进行进一步将其编码成为 Bitmap 格式。实际上当前图片的大小为 4032*3024，于是我们对 Bitmap 进行一次裁剪以达到模型文件接受的尺寸，即 224*224，最后对图片在 3 个色轴分别减去一个均值（128）后就完成了对输入图片的预处理。

完成对图片的预处理之后就可以通过模型对这一帧图片进行识别分类了。与规划稍有不同的是，因为目前只在科教南楼的一段实验路径上进行了部署，因此目前的模型文件、label 以及地图文件直接存放在应用的 assets 目录中而不是通过服务器下载，在后期的完善工作中我们会建立一个专门用于文件上传、存储和下发的服务器。识别工作中，CNN 会给出输入的图片所记录的位置对 N 个类别的置信度列表。以在科教南楼的部署为例，输出的 285 个 float 类型的置信度表示输入的位置与这 285 个位置点匹配的概率，部分数据如表 4-1 中所示是在 00005 位置出的一帧图片的原始识别结果。

表 4-1 00005 位置点一帧图片的置信度（部分）

placeId	置信度	placeId	置信度
00000	0.21359437	00010	0.0069586
00001	0.4418504	00011	0.0220535
00002	0.61590968	00012	0.02622369
00003	0.57759444	00013	0.01011506

00004	0.89327326	00014	0.02467362
00005	0.8118336	00015	0.09189534
00006	0.20680681	00016	0.01782522
00007	0.18309604	00017	0.01116372
00008	0.09223509	00018	0.01121536
00009	0.5337838	00019	0.0196555

4.3 定位算法的设计

观察表 3-1 中的识别结果我们会发现，作为 00005 位置点的一帧图片的识别结果，原始数据与我们的期望有一些出入，按照我们的设想，在所有识别结果中最高的置信度应该对应 `placeId` 为 00005，而实际上表 4-1 中给出的部分原始分类结果中显示最高的置信度却对应着 00004 位置。实际上在此处这个置信度应该被算作是一个异常结果，并做丢弃处理。但这依然无法解决我们需要从原始分类结果中找出最匹配的位置点的需求，因为我们无法确定直接确定哪些值是异常的，而且通过对实验结果的统计我们发现直接以最高的置信度对应的 `placeId` 作为分类结果其错误率是比较高的。因此我们需要对原始的分类结果进行处理。

通过对多组原始分类结果的统计和观察我们发现 285 个置信度呈现以下特点：

1. 相邻的几个位置点对应的置信度，其大小总是相近；
2. 当前这一帧图片所在位置点和与其对应置信度相近的几个相邻位置点的置信度在所有置信度中总体来看较大。
3. 当前这一帧图片所在位置点在与其相邻且置信度相近的几个位置点中其置信度也总是最大。
4. 在对可以正确分类的实验中的置信度进行排序后发现，靠前的数个置信度其算术平均值较小。

对以上四个关于原始置信度的特点的分析后，我们对其作以下一系列处理和运算以获取最佳的分类结果，相应算法的伪代码在表 3-2 和表 3-3 中给出。

在此对表 3-2 和表 3-3 中的两个算法做出解释。

在算法 1 中，首先，为了开发时的方便，我们需要对原始的置信度进行一次排序。在开发中，我们定义一种名为 `Recognition` 的数据结构来存储 `placeId` 和其置信度之间的对应关系，它包含 `placeId` 和处于该位置点的置信度两个变量，并使用优先队列来存

储这种数据结构，优先队列以置信度作为排序基准从而完成排序。接着，我们从排序后的队列中获取前 20 大 Recognition，我们对前 20 大结果做以下运算，比较第 6 至 20 大置信度的算术平均值与 0.1 的大小，若其均值大于 0.1 则表明当前位置已偏离目标路径，应当用户重新回到目标路径；否则，表明用户当前位置正确处于目标路径上，于是我们进入下一步运算（需要指出的是，在此处对于置信度的算术平均值的计算中之所以舍弃前 5 大位置，是因为无论当前用户所在位置是否处于目标路径中 CNN 给出的置信度几乎总是较大的，如总是大于 0.5，这对于判断是没有意义的，因此被舍弃）。

表 3-2 计算定位结果的算法，伪代码描述

算法 1 根据置信度列表计算定位结果 placeId 的算法

Input: List listOfRecognition = {r0, r1, ..., r283, r284}

Output: placeId or awayFromTrace

```

1  listSorted = doSort(listOfRecognition)
2  list20 = listSorted(0 ... 20)
3  confidenceSum1 = 0
4  for each Recognition con in list20{5 ... 20} do
5      confidenceSum1 += con
6  end
7
8  if confidenceSum1 / 15 >= 0.1 then
9      return awayFromTrace
10 else
11     clusters = doCluster(list20)
12     confidenceSum2 = 0, tmpSum = 0, maxCluster
13     for each Cluster c in clusters do
14         for each Recognition r in c do
15             confidenceSum2 += r.confidence
16         end
17         if tmpSum > confidence then
18             confidenceSum2 = tmpSum
19             maxCluster = c
20         end
21     end
22     placeId = maxCluster.maxRecognition.placeId
23 return placeId
24 end

```

若算法 1 中前半部分的计算结果为用户在目标路径上，接着我们进入算法 1 后半部分的运算。我们主要考虑前三个特点体现出的问题，即分类结果在相近位置点之间

的相似性。为了在计算中获取和体现这种相似性，我们进行一步名为“聚类”的计算，参看算法 2。具体做法为：对于前 20 大分类结果中的每一个 Recognition 通过其 placeId 查询地图文件，将具有相邻关系的两个或多个位置点对应的 Recognition 归为一个“类”。什么是相邻关系呢？当一个位置点为另一个位置点的 fromNode 或 toNode，或一个位置点存在于另一个位置点的 switchNodes 中时，定义这两个位置点具有相邻关系，相邻关系之间可以传递。完成“聚类”后前 20 大分类结果会被归为 1 到 20 个子类。

在算法 1 的最后，对每一个子类求其中所有 Recognition 对象的置信度的算术平均值。而用户当前所在位置便最终被归为置信度的算数平均值最大的子类中，拥有最大置信度的 Recognition 对象对应的位置点。

至此，我们便完成了对用户的定位。

表 3-3 对置信度进行聚类的 doCluster 函数的伪代码描述

算法 2 对原始置信度列表进行聚类的算法

Input: List list20 = {r0, r1, ..., r18, r19}

Output: listOfCluster

```

1  List listOfCluster, i= 0
2  while list20.length > 0 do
3      Cluster c, c.add(list20[i]), list20.remove(list20[i]), len = c.length
4      for each Recognition r1 in list20(i + 1, list20.length) do
5          for each Recognition r2 in c do
6              if r1 in r2.fromNodes || r1 in r2.toNodes || r1 in r2.switchNodes then
7                  c.add(r1)
8                  list20.remove(r1)
9                  break
10             end
11         end
12         if c.length > len then
13             len = c.length
14         else
15             listOfCluster.add(c)
16             break
17         end
18     end
19 end
20 return listOfCluster

```

4.3 构造导航信息

完成 4.3 节中对用户定位工作之后，就可以根据用户当前所处的位置情况为用户给出导航信息提示了。根据 4.3 中算法-1 返回的定位结果，共有两大种导航信息：

- 用户已偏离目标路径。
- 用户当前正确处于目标路径上一个确定位置点。

而对于两种不同的定位结果，导航 App 又分别给出不同的导航建议。首先来说用户已偏离目标的情况，当用户偏离目标之后 DeepNavi 系统会尝试为用户提供参考信息以帮助用户回归到正确的路径。在当前版本的实现中，系统会记录用户在发生偏离目标路径错误之前的最后一次成功定位结果，然后一句此从服务器获取这个位置点的图片信息，在对用户进行已偏离正确路径的警告之外还会在界面上展示该图片信息，提示用户回退到图示的上一个位置点重新开始定位和导航。

而在第二种，定位用户至目标路径上的一个确定的位置点的情形中，DeepNavi 系统为用户提供的导航信息主要包括两类：

- 用户当前所处的位置点（在目前的系统版本中通过直接输出当前定位到的 `placeId` 的方法来告知用户所在位置，考虑到这种做法对用户使用不够友好，因此预计在后续的系统优化中会在构建地图时标出 `placeId` 所在位置点两米范围内的标志建筑物信息，完成定位后结合 `placeId` 与周围建筑物信息对用户展示，以期让用户可以更好的参考自己当前的位置）。
- 用户需要关注的下一转向位置或台阶（在此我们统称其为转向点）。我们知道，在走廊等区域行走时绝大多数时候唯一需要我们去分辨的位置只有转向或楼梯等区域，其他时候只需要沿着前后贯通的路段前进即可，鉴于此，在我们的系统中为用户提供的另一个导航信息便是转向点的图像信息，如图 4-1 中右下角的图片控件所示。

下一个转向点信息的获知需要结合定位结果和地图文件，我们结合图 3-3 中的从 S 到 T1 之间的示例路径和对应的示例地图文件来进行说明。一名不熟悉该区域内部环境的用户进入该建筑后打开部署本系统的导航 App，通过输入入口 S 和目标位置 T1 后 App 主动从服务器下载该区域的地图和模型文件，完成对文件的加载等工作后提示用户开始移动。

用户从 A 位置点出发，与此同时系统完成对用户的定位，获知当前位置点 `placeId` 为 0000A，于是在 App 中更新当前所处位置为：`placeA`。接着系统开始从地图中查询

下一个转向点，即 sm 不为 0 的节点，查询结果返回为 0000c1（并存储这个值为 $switchNodeId$ ），其 $switchNodes$ 包含 0000c2 和 0000c3 两个节点，也即两个 $toNodes$ ，于是系统进行一次选择以决定哪个 $toNode$ 可以到达终点 E。选择的过程为分别从两个 $toNodes$ 出发对地图进行遍历并返回最终到达 E 节点的 $toNode$ ，若两个 $toNode$ 对终点 E 均可达，则返回遍历代价（即层数）更小的那个，在这里返回 0000c3（并存储这个值为 $turnNodeId$ ），于是从服务器下载 0000c3 位置点的图片信息，在 App 中描述下一个转向点图片的控件上进行展示。

此时用户继续前进经过位置点 B，定位后更新当前所在位置的位置点 Id 为 $placeB$ ，查询下一个转向点发现返回结果和 $switchNodeId$ 相同，因此不对转向提示图片进行更新。接着用户继续前进至 c1 位置，发现面前有直走（到 c2）和右转（到 c3）两个选择，于是参考 App 上的转向提示图片后选择 c3。

接下来用户前进至 c3 位置，定位结果显示 $placeId$ 与当前 $turnNodeId$ 相同，于是除了更新当前所在位置信息之外，重新查询下一个转向点，对地图的遍历返回 $tm=0$ 的终点 E，于是更新 $switchNodeId$ 为 0000E，并在转向提示图片位置更新从服务器获取的 0000E 位置的图片。需要注意的是，此时图片上发的文字也会更新为：直行，您即将到达此次导航的终点，如下图所示！至此，我们便完成了定位和导航的全部工作。

3.4 本章小结

我们在本章重点论述了导航模块中对用户实时位置的定位，和依据定位结果查询地图并对用户给出导航提示。首先我们介绍了导航 App 对于模型和地图文件的下载，接着我们阐述了神经网络的输入内容——用户实时位置的图片的获取和预处理。然后我们结合对部分实验数据的分析详细介绍了定位部分的两个关键算法。最后我们以第 3 章中创建的模拟区域为例，详细地展示根据定位经过构造多种导航信息的方法。

第 5 章 测试和评估

为了对系统在定位和导航精度、系统布署时间和布署成本、以及系统对不同环境的适应性等方面的表现进行评估，我们在科教南楼一楼入口到四楼卫生间之间的这一段目标路径上对 DeepNavi 系统进行了布署。并在安装有 DeepNavi 导航 App 的 Android 智能手机上进行了测试实验，并通过对实验结果的记录和分析，对系统在各方面的表现进行了评估，以及最后给出了关于系统的优化和改进的后续研究方向和研究计划。

5.1 布署和实验

5.1.1 在不同性能手机上的实验结果

我们测试使用三部不同的智能手机（分别为 2014 年生产的小米 note，2018 年生产的小米 8 和 vivo Nex 手机）分 3 组，每组 2 次实验共 6 次，对于从起点到终点共 285 个位置点进行室内定位导航的实地测试。实验结果并记录为表 5-1～表 5-4，在这四个表中我们分别列出了 6 次实验中 placeId 定位正误次数、准确率，转向点获取正误次数、准确率，以及平均值。

表 5-1 在小米 Note 手机上的实验数据

	正确定位次数	定位成功准确率	正确获取转向点 次数	转向点获取正确 率
实验 1	263	92.28%	10	90.91%
实验 2	261	91.58%	11	100.00%
平均值	262	91.93%	10.5	95.46%

表 5-2 在 VivoNex 手机上的实验数据

	正确定位次数	定位成功准确率	正确获取转向点 次数	转向点获取正确 率
实验 1	268	94.03%	11	100.00%
实验 2	270	94.74%	11	100.00%
平均值	269	94.39%	11	100.00%

表 5-3 在小米 8 手机上的实验数据

	正确定位次数	定位成功准确率	正确获取转向点 次数	转向点获取正确 率
实验 1	273	95.79%	11	100.00%
实验 2	276	96.84%	11	100.00%
平均值	274.5	96.32%	11	100.00%

表 5-4 三组实验的平均值

	正确定位次数	定位成功准确率	正确获取转向点 次数	转向点获取正确 率
平均值	267.83	93.78%	10.83	98.48%



图 5-1 122, 123, 124, 125 四个位置点

5.1.2 不同路段的实验结果和对比

我们知道，导航系统的精度和稳定性等的表现在不同的环境中受影响较大。为了测试 DeepNavi 在不同路段的表现，我们记录了系统在科教南楼的几种路段上的实验数据。这几种路段分别是：走廊，楼梯间，出入口。对应的三种路段如图 5-1 所示。其中，走廊路段选取四楼的一段，共 21 个位置点；楼梯间路段选取 4 层楼共 3 个楼梯间，总共 54 个位置点；出入口选取布署路段内共 7 个出入口，13 个位置点。

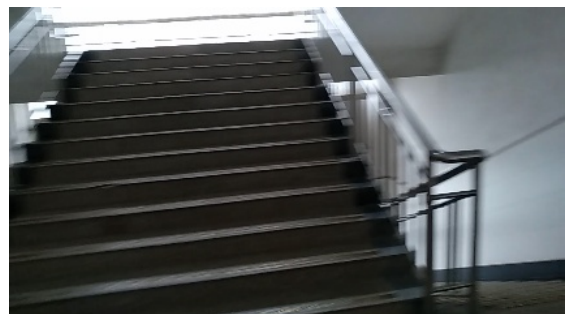
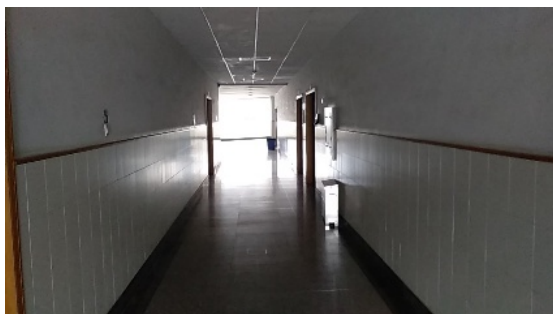




图 5-2 走廊、楼梯间、出入口三种路段

对这三种路段的实验结果记录在表 5-5 到表 5-7 中。

表 5-5 走廊路段的实验数据

	错误定位次数	错误定位概率
小米 Note	5	23.81%
Vivo Nex	4	19.05%
小米 8	2	9.52%

表 5-6 楼梯间路段的实验数据

	错误定位次数	错误定位概率
小米 Note	2	3.70%
Vivo Nex	3	5.56%
小米 8	2	3.70%

表 5-7 楼梯出入口路段的实验数据

	错误定位次数	错误定位概率
小米 Note	1	7.69%
Vivo Nex	0	0
小米 8	0	0

5.2 分析实验结果

分析表 5-1 到表 5-4 的实验数据，我们首先来看导航 App 搭载的实验设备对于实验结果的影响，一个显而易见的现象是每一部手机对应的一组中两次实验结果相差较小。其次比较三组实验的组内平均值发现第二组和第三组相差很小，而第一组实验结果与另外两组相比误差相对而言稍为高一些，考虑到小米 Note 一代手机发布距今已有

五年，因此其处理器、摄像头等硬件与另外两部发布于去年的手机会产生较为明显的差距，即便如此这个 3% 左右的正确率差距单纯从数值角度来看依然算不上很高，因此我们可以得出结论“基于 CNN 的室内导航系统”DeepNavi 在不同硬件上的差异并不明显，具有相当不错的稳定性，即使在产于五年前的性能较为落后的手机上也可以去的较好的表现。

接着我们来看定位的准确率，在六组实验中对于 285 个位置点其定位准确率平均值为 93.98%（正负误差不超过 2%），尽管从数值上来看这是一个很不错的准确率，但考虑到在一个导航系统中，因此这个结果并不能让我们非常满意。但通过对实验过程的记录，定位出错的位置点总体来看具有以下特点：

1. 该位置点和与其相邻的几个位置点相比，图像特征差异度较低，如图 5-1 中所示，这是 00122，00123，00124，00125 四个位置点的照片，我们可以发现这四张图片具有很高的相似度，尤其每一张图片的中间部分几乎是相同的。
2. 在定位出错的情形中，定位偏差（定位得到的位置点与正确位置点之间的距离）不超过 2 个位置点的情形接近 3/4。
3. 较为特殊的是，在同一个位置（距一楼大厅十米左右位置）六次实验定位出错的次数达到 5 次。

分析以上三个关于定位准确率的特点我们可以得出以下结论，首先对于特点 1 中描述的图像信息与相邻位置相似度较高导致的定位准确率低的问题，我们推测这在很大程度上与神经网络有关。预计这个问题可以通过对模型训练数据集的增大来改善。如 3.4 小节中所述，DeepNavi 的当前版本对每一个位置点的训练集的获得方式为，从领航员手持手机录制的视频切分成的时长为 1 秒的视频分量中以每秒 30 帧进行采样获得的一个 30 张图片的数据集。由于这 1 秒视频记录位移约为 1 米左右，因此不难理解这 30 张图片相似度会非常高，此外，对训练数据集的查看也证实了我们的这种猜想。

通过分析特点 1，特点 2 中描述的情形也不难理解。由于在某一些位置点其图像特征差异度相比附近位置点很小，因此会出现这几个相邻的位置点所在的聚类中最大的置信度和正确的位置点之间出现了 1 个到 2 个位置点的偏差。通过对实验数据的进一步观察我们发现，这种情况多发生在如图 5-1 中所示的单一直通路径，如不需要转向的走廊。而这 1 到 2 两个位置点的定位偏差在这种不需要转向的直通路径中，并不会

影响系统对于下一个转向点的查询和判定，而用户在这种位置也不需要做过多判断直行即可，因此并不会导致较严重的问题。

再来看表 5-5 到表 5-7 三个表格的实验数据，对比三种路段来看，走廊路段的定位出现误差的几率要明显远高于另外两种路段。参考实验过程我们可以发现，定位的误差多数不超过 2 个位置点。这从侧面也印证了我们上面的讨论结果，走廊中前后相邻的几个位置点图像特征差别较小，所以导致模型分类时出现偏差。6 个楼梯间出入口和一个教学楼入口之间图像特征差异较大，这也反映在了表 5-7 中非常低的定位错误率上。而楼梯间因为窗外景物的差别较大，以及与楼梯相邻的出入口也具有明显的图像特征，因此定位错误率也较小。

至于第 3 点中提到的在同一个位置连续定位出错的问题，我们认为这是一个特殊现象或个例。因为这个问题并没有出现在实验路径的其他位置。值得一提的是，这个位置的定位结果的偏差不同于第一点和第二点中描述的情形，即定位偏差总是出现在图像特征差异度较小的相邻位置点之间，在这个位置点的偏差非常严重。我们推测这与模型有关，不过依然有待验证。我们将在后续更多的实验和部署中持续关注这个问题。

最后我们来看一下表 5-1 中后两列关于 6 此实验中对转向点的获取的情况。从一楼入口到四楼卫生间的这段路径共包含 11 个转向点，3 组实验中只有第 1 组中出现了一次位置点获取错误的情况，6 组实验中转向点获取的正确率平均达到 98.48%，这是一个相当不错的结果。同时，这个结果也从侧面证明了我们在定位准确率部分的分析，即定位偏差多出现在图像特征差异度较小的相邻位置点之间，而这些位置点的定位偏差几乎不会影响系统对于转向点的查询和判定。

5.3 系统优化

通过对实验结果的记录和分析，结合系统设计和布署中出现的一些问题，我们对 DeepNavi 的后续研究将主要关注对模型训练方式的优化，导航提示信息的优化，以及对图片中行人的擦除等可以继续深入研究的点。

我们对 DeepNavi 的定位结果在图像特征差异度较小的相邻位置点之间出现偏差的问题进行了详细探讨。我们分析导致这个问题的主要原因在神经网络的构建，如训练数据集不够。DeepNavi 的当前版本对每一个位置点的训练集的获得方式为，从领航员手持手机录制的视频切分成的时长为 1 秒的视频分量中以每秒 30 帧进行采样获得的一个 30 张图片的数据集。由于这 1 秒视频记录位移约为 1 米左右，因此不难理解这 30

张图片相似度会非常高，对这 30 张图片的查看也证实了我们的这种猜想。因此我们可以通过采集更多的数据从而产生更大的训练集来改善我们的模型的表现，如同时持三个手机进行采集。

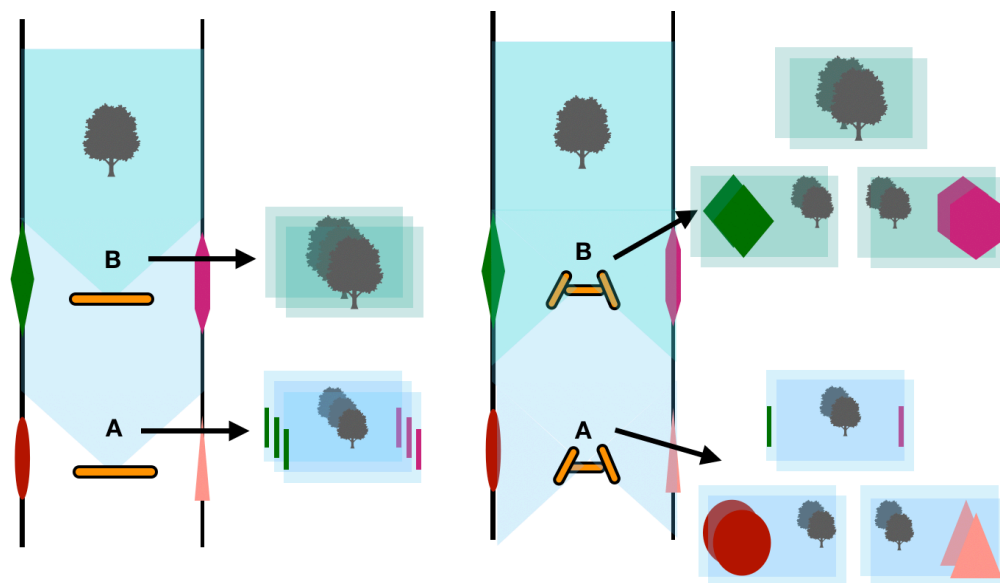


图 5-3 数据采集方式的改进方法

如图 5-3 左侧部分为目前的数据采集方式，在一条直道路径上领航员持手机（保持垂直于水平方向）稳定前进拍摄，这带来的一个问题就是几乎无法采集到当前位置的左右两侧的丰富图像信息，图像中的绝大部分信息来自前方景物，从而导致从视频分量中取出的训练图片相似度极大，以及两处获取到的图片中主要信息几乎相等的问题，进一步考到在 CNN 模型对图片进行分类之前还会进行裁剪，因此在实验中出现对 B 位置定位结果为 A 等错误。

如图 5-2 中右半部分所示，我们对数据采集方式进行优化。与当前的持一台手机进行数据采集的方式不同的是我们额外增加两台手机分别同时对左右两侧的图像进行采集，即三台手机同步录制。采集完成后从三段视频中同步开始采样来获取训练集，（即首先对 3 段视频同时进行 1 秒为单位的切分，于是每个位置点会同时映射到 3 段视频分量上。然后对这 3 段视频分量以同样的时间间隔来进行采样，如 1/20 秒，从而每一段视频分量中会产生一组共 20 帧，三组共计 60 帧图片，3 组图片组间差异较大，并且更大的采样时间间隔也会产生更大的组内图片之间的差异性。从而大大丰富了我们的训练集）。

另外一个需要优化的点在于当前版本的客户端实现中，定位的结果甚至定位的偏差对于用户来说的感知并不明显或直观，因为我们为用户展示的仅仅是一个五位的形

如 00001 的格式化 placeId，而用户实际上是并不清楚这个 placeId 的含义的，也不清楚这个位置点意味着目标路径上的哪一个具体位置。因此在后续的改进工作中我们可以考虑对用户目前所处的位置同样使用图像信息进行告知，并且考虑到相邻位置点之间有时会出现定位偏差的问题，我们将为用户提供除了定位结果对应位置的图像之外，还附加该位置前后两个相邻位置的图像作为用户对附近位置的参考。

最后我们再来说一下对图片中行人的擦除问题。实际上在早期的将运算集成在服务器中的架构中我们实现了这个功能，而测试结果也显示这个功能对于系统定位的准确性有较大的提升。我们在 3.5 节中阐述了在当前版本中去掉该功能的原因和考量。但在后续的工作中，我们将继续尝试在客户端实现该功能，以期获得更高的精度。

5.4 本章小结

本章中我们对系统在科教南楼进行了布署和测试。并对实验结果从定位精度和转向点的获取的准确率进行了详细分析。实验结果表明，DeepNavi 拥有相当不错的精度和稳定性，这让我们对 DeepNavi 的大规模部署和推广产生了很高的期待。最后我们根据对实验结果的分析 and 探讨，对 DeepNavi 的一些不足的地方做出了改进规划。

第 6 章 总结和展望

6.1 总结

为了解决精确室内导航问题，我们在本文中提出了 DeepNavi，一种基于 CNN 的室内导航系统。DeepNavi 部署方式简单而高效，即插即用，且保证相当高的使用精度和稳定性。

与多数对于室内导航的研究不同的是，DeepNavi 不需要任何额外的基础设施支持，部署方式简单，覆盖绝大多数应用场景。系统通过目标路径的图像信息训练卷积神经网络和生成地图文件，神经网络对用户所在实时位置的图像进行分类从而对用户定位，地图文件根据定位结果对用户给出导航提示完成导航。

DeepNavi 采用自激励的方式来鼓励领航员对系统的主动布署。我们在 DeepNavi 中设计采用了一种特殊的数据标注方法对目标区域的图像数据进行处理，并基于此进行了对目标区域位置点的划分和地图的构建，以及模型的训练工作。此外，我们开发了一些高效的算法来对神经网络的分类结果进行计算来定位用户的精确位置。

最后我们在 Android 智能手机上实现了 DeepNavi 系统，并在一栋 7 层的教学楼内部进行了实地布署和测试评估。实验结果展示了 DeepNavi 系统拥有相当不错的精度和稳定性表现。

通过对系统的实现和布署，以及对实验结果的分析，我们认为 DeepNavi 系统很好地完成了本文的研究目标。当然，当前版本的 DeepNavi 并不完美，还有很多可以优化的地方。我们将在后续的研究中对系统在定位精度和易用性等方面持续做出改进，以提供更好的服务。我们期待有一天可以对 DeepNavi 进行商业化部署和大范围推广。

6.2 展望

至此，我们已经完成了“基于 CNN 的室内导航系统”从数据采集、标注、模型和地图的构建、以及定位和导航等所有步骤的设计和工作。并且在对系统的测试和评估当中获得了相当不错的表现。

在以后的工作中，我们的研究重点将集中在对系统定位精度的进一步提高，如设计更优的模型、开发更加高效的算法等。其次，我们还将会开展更多的测试和实验，对系统在实际环境中的表现继续优化。此外，我们将尝试在一些区域布署和推广 DeepNavi 系统。

致谢

时光飞逝，转眼间本科阶段最后一个，也是最重要的一个课程项目的研究和学习结束了，自己的大学学习也即将迎来结尾。回首在中南这四年的学习和生活经历，有太多的良师益友对我提供了帮助和指导。

在此，首先特别感谢本次毕设的指导老师任炬教授。在毕业设计和毕业论文的开展过程中，任老师从研究方向和题目的拟定，到导航系统的方案设计和开发各个阶段都对我提供了详细的指导和大量的帮助。任老师悉心负责，事无巨细，详细安排和推动本次毕设每一个阶段的工作。在繁忙的研究和教学工作之外任老师依然坚持定期开会交流毕设进展，并对我们的不足之处进行提醒和开导，确保了本次毕业设计顺利落实。任老师对研究和教育认真负责的态度值得我们每一个人学习。

此外，我还要感谢弓箭博士。弓箭博士对我本次毕业设计全程给予了无私的帮助和指导。从早期参考文献的阅读，到设计和开发中数据的采集，地图的构建，和模型的训练等方面均作出了关键的指导。在我毕设开展过程中遇到难题时，他总能给我从技术到思路甚至心态个方面的建议。祝愿弓箭博士的研究取得丰硕的成果。

我还要感谢我的同窗好友许嘉晨等人，他们在我并不擅长的深度学习和 `python` 开发等领域对我提供了相当大的指导和帮助，在研究和开发陷入困境时是他们的帮助让我能够走出困难。

最后，我还要感谢我的父母和家人。学习计算机技术是我的爱好和兴趣所在，感谢他们当年对我的选择的理解和这四年来支持。

参考文献

- [1] 王贝贝, 王宗爽, 赵秀阁, 等. 我国成人室内外活动时间研究[J]. 环境与健康杂志, 2014, 31(11):945-948.
- [2] Zheng Y, Shen G, Li L, et al. Travi-Navi: Self-Deployable Indoor Navigation System[J]. IEEE/ACM Transactions on Networking, 2017, 25(5):2655-2669.
- [3] Shu Y, Shin K G, He T, et al. Last-Mile Navigation Using Smartphones[J]. 2015.
- [4] Zhang C, Subbu K P, Luo J, et al. GROPING: Geomagnetism and cROwdsensing Powered Indoor NaviGation[J]. IEEE Transactions on Mobile Computing, 2015, 14(2):387-400.
- [5] Shu Y, Bo C, Shen G, et al. Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing[J]. IEEE Journal on Selected Areas in Communications, 2015, 33(7):1443-1457.
- [6] Bang W, Chen Q, Yang L T, et al. Indoor smartphone localization via fingerprint crowdsourcing: challenges and approaches[J]. IEEE Wireless Communications, 2016, 23(3):82-89.
- [7] Zhang C, Subbu K P, Luo J, et al. GROPING: Geomagnetism and cROwdsensing Powered Indoor NaviGation[J]. IEEE Transactions on Mobile Computing, 2015, 14(2):387-400.
- [8] Yuan Z, Syed Z, You L, et al. Evaluation of Two WiFi Positioning Systems Based on Autonomous Crowd Sourcing on Handheld Devices for Indoor Navigation[J]. IEEE Transactions on Mobile Computing, 2016, 15(8):1982-1995.

- [9] Wang X , Gao L , Mao S , et al. CSI-based Fingerprinting for Indoor Localization: A Deep Learning Approach[J]. IEEE Transactions on Vehicular Technology, 2016, 66(1):763-776.
- [10] Wu C, Zheng Y, Xiao C. Automatic Radio Map Adaptation for Indoor Localization using Smartphones[J]. IEEE Transactions on Mobile Computing, 2018, PP(99):1-1.
- [11] Atia M M, Liu S, Nematallah H, et al. Integrated Indoor Navigation System for Ground Vehicles With Automatic 3-D Alignment and Position Initialization[J]. IEEE Transactions on Vehicular Technology, 2015, 64(4):1279-1292.
- [12] You L, Yuan Z, Peng Z, et al. An improved inertial/wifi/magnetic fusion structure for indoor navigation[J]. Information Fusion, 2017, 34(C):101-119.
- [13] Guo X, Ansari N. Localization by Fusing a Group of Fingerprints via Multiple Antennas in Indoor Environment[J]. IEEE Transactions on Vehicular Technology, 2016, PP(99):1-1.
- [14] Chang L , Xiong J , Wang Y , et al. iUpdater: Low Cost RSS Fingerprints Updating for Device-Free Localization[C]// 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE Computer Society, 2017.
- [15] Chen X , Ma C , Allegue M , et al. Taming the Inconsistency of Wi-Fi Fingerprints for Device-Free Passive Indoor Localization[C]// IEEE INFOCOM 2017. IEEE, 2017.
- [16] Guo B , Han Q , Chen H , et al. The Emergence of Visual Crowdsensing: Challenges and Opportunities[J]. IEEE Communications Surveys & Tutorials, 2017, PP(99):1-1.
- [17] eng X , Guo D , Zhou X , et al. Poster: An Indoor-Outdoor Navigation Service for Subway Transportation Systems.[C]// Acm Conference on Embedded Networked Sensor Systems. ACM, 2015.

- [18] Xiang L , Tai T Y , Li B , et al. Tack: Learning Towards Contextual and Ephemeral Indoor Localization with Crowdsourcing[J]. IEEE Journal on Selected Areas in Communications, 2017:1-1.
- [19] Yin Z , Wu C , Yang Z , et al. Peer-to-Peer Indoor Navigation using Smartphones[J]. IEEE Journal on Selected Areas in Communications, 2017:1-1.
- [20] Li Z , Shu Y , Karlsson, Börje F, et al. Demo: Towards Flexible and Scalable Indoor Navigation[C]// International Conference on Mobile Computing & Networking. ACM, 2017.
- [21] Liu Z , Zhang L , Liu Q , et al. Fusion of Magnetic and Visual Sensors for Indoor Localization: Infrastructure-Free and More Effective[J]. IEEE Transactions on Multimedia, 2017, 19(4):874-888.
- [22] Huang W , Xiong Y , Li X Y , et al. Swadloon: Direction Finding and Indoor Localization Using Acoustic Signal by Shaking Smartphones[J]. IEEE Transactions on Mobile Computing, 2015.
- [23] Liu K , Wu D , Li X . Enhancing Smartphone Indoor Localization via Opportunistic Sensing[C]// 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2016.
- [24] Li W, Wei D, Lai Q, et al. Geomagnetism-Aided Indoor Wi-Fi Radio-Map Construction via Smartphone Crowdsourcing.[J]. Sensors, 2018, 18(5).
- [25] Cheok A D, Yue L. A Novel Light-Sensor-Based Information Transmission System for Indoor Positioning and Navigation[J]. IEEE Transactions on Instrumentation & Measurement, 2010, 60(1):290-299.

- [26] Wang X, Gao L, Mao S. CSI Phase Fingerprinting for Indoor Localization With a Deep Learning Approach[J]. IEEE Internet of Things Journal, 2017, 3(6):1113-1123.
- [27] Jie W, Xiao Z, Gao Q, et al. Device-Free Wireless Localization and Activity Recognition: A Deep Learning Approach[J]. IEEE Transactions on Vehicular Technology, 2017, 66(7):6258-6267.
- [28] 包志军, 马培荪, 仝建刚, et al. 人行走速度规律的实验研究[J]. 实验室研究与探索 (06):39-42.
- [29] 瞿晓娜, 张腾宇, 王喜太, et al. 行走速度对步态参数影响的实验研究[J]. 中国康复医学杂志, 2012, 27(3).
- [30] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[C]// NIPS. Curran Associates Inc. 2012.
- [31] Simonyan K , Zisserman A . Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
- [32] Szegedy C , Vanhoucke V , Ioffe S , et al. [IEEE 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Las Vegas, NV, USA (2016.6.27-2016.6.30)] 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Rethinking the Inception Architecture for Computer Vision[J]. 2016:2818-2826.
- [33] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.