近些年来，深度学习技术得到了长足的发展，为图像分类、机器翻译、语音识别等多媒体问题提供了高精度高性能的解决方案。自高精度的图像分类CNN网络AlexNet【来源请求】问世以来，卷积神经网络凭借其精度优势，逐渐成为物体识别、图像分类、图像语义分析等图像、视频处理领域问题的主流选择。在AlexNet被提出之后，学界发展了大量精度更高、结构更复杂的卷积神经网络模型，如R-CNN【来源请求】、Fast R-CNN【来源请求】、Faster R-CNN【来源请求】、YOLO【来源请求】。基于卷积神经网络的图像处理技术在业界也得到了广泛应用，并渗透到了每个人的日常生活中。

如，在自动驾驶领域，Google基于图像物体检测技术启动了智能自动驾驶汽车计划【https://www.google.com/selfdrivingcar/】，截止至2012年，其推出的Waymo自动驾驶系统已经装上Lexus RX450h试验车已经完成了30万英里的无人驾驶旅程。2017年，Waymo项目已经推出了其第一款基于量产车型的自动驾驶汽车，并且在亚利桑那州开始召集体验用户为他们提供回馈；

在医疗方面【http://www.nvidia.com/object/deep-learning-in-medicine.html】，深度学习技术也可以通过预训练的神经网络对CT影像、组织切片影像等图像进行分析，为医护人员做出诊断提供可靠的参考。日前即将落幕的Kaggle大数据竞赛Data Science Bowl 2017大赛的主题【https://www.kaggle.com/c/data-science-bowl-2017】即为根据CT图像等数据集准确预测肺癌。

标榜“智能”的智能手机厂商们也在利用深度学习技术，为手机用户开发更加智能的应用、提供更加智能的用户体验。从较为简单基础的相机识别人脸、自动聚焦，到复杂的云相册服务（识别并分类相册中的人脸，自动根据照片中含有“谁”来进行整理归类；识别各照片中的物体并标记，以便用户检索相册中的特定照片，等）。

随着更新的卷积神经网络模型在深度、广度上进一步变复杂，这类任务的计算量对服务器的压力也进一步增大。一些智能手机厂商为了减少服务器的负担，决定采取直接在手机上运行卷积神经网络模型的做法。如最近更新的iOS10【来源请求】系统中引入的智能相册就应用了运行在手机本地的CNN模型，苹果公司近期也对iOS开发者开放了神经网络、深度学习相关的应用程序接口（API），让开发者能够调用这些现有函数开发更加智能的应用软件。

然而这种闭源的深度学习接口无法满足很多智能手机的开发需求：首先，它是闭源的，开发者无法根据需求调整，只能调用现成的函数；其次，这套接口是苹果公司开发的，只能用于iOS系统，安卓开发者、Windows Phone开发者只能望洋兴叹；第三，学界的CNN模型正不断推陈出新，很多近期较为复杂的CNN模型的网络结构、损失函数等已经无法用常规的CNN操作（如卷积层、全连接层、Max Pooling层、ReLU激活函数等）描述。如，R-FCN【来源请求】模型中引入了非常规的Pooling操作（ROI Pooling与PSROI Pooling）；YOLO【来源请求】模型中为了统一物体识别与分类，引入了之前从未被现成框架实现过的损失函数。可以预见的是，CNN模型的复杂度会随着学界的对精度的更高要求而进一步提升，这类无法用常规操作描述的模型也将变得越来越多。

在手机上，尤其是安卓手机上进行深度学习开发，目前可替代苹果公司神经网络API的选项中，最容易想到的当然是将各种开源的神经网络框架，如Caffe、TensorFlow、Torch等移植到手机上。这些框架一般允许用户直接通过预定格式的文本文件去定义较为基本的网络结构，并规定训练、测试、实际使用时所执行的动作。统计学习的数学研究者甚至无需掌握一定的编程技巧即可在这些框架上测试自己的研究想法。由于其开源性质，一些含有自定义复杂操作而无法用原版框架去描述的模型，也可以通过修改源码的方式实现。

在业界诸多开源深度框架中，Caffe比较适合移植到手机上供开发者进行深度学习相关开发：首先，Caffe的主体代码构成是C++，不使用NVIDIA CUDA进行GPU计算的话，其依赖库也均为开源项目。也就是说，理论上讲Caffe可以移植到任何支持C++编译工具链的平台上。安卓系统的主要开发语言虽然是Java，但拥有完整可靠的Native Development Kit工具链，可以交叉编译C++等底层语言到指定手机上，并封装native接口供Java语言写成的安卓应用调用。相比之下，以Theano等框架为代表的深度学习框架对外开放的接口主要是Python等高级脚本语言，缺乏底层支持，很难移植到没有原生Python支持的大部分手机上；其次，Caffe得益于其靠近底层的实现，虽然没有官方支持ARM手机架构，但在PC端主流CPU架构（x86-32，x86-64）上的性能表现比较突出；第三，Caffe因为设计上考虑了可扩展性，自定义的操作可以参考文档，规范化地实现。Caffe在底层封装的大量接口可以让开发者非常方便地直接访问到底层数据，加入自定义的操作抽象成层非常方便。如近期因高精度、高性能瞩目的CNN物体识别论文SSD【来源请求】、R-FCN【来源请求】等，其作者所在实验室放出的项目源码即是自定义版的深度学习框架Caffe。

综上，这篇论文选择Caffe作为基础，将其移植到ARM架构的手机芯片上，并在它的基础上进行优化，以提供一个高性能、可扩展的深度学习套件。