**标题**

基于云端图像处理的移动多节点实时人流量监控系统

An AI Image Processing Server Based Real-Time Multi-Node Crowd Monitoring System

**摘要**

为了展示5G技术和AI技术融合能够带来的令人激动的应用前景，受到疫情期间各种新型应用的启发，我们实现了一个实时的人流量监控系统。本项目的监控系统和传统的有线、闭路、相机固定的监控系统不同——我们使图像采集端设备能够受远程控制台控制，达到“指哪、到哪、看哪”的效果，同时我们创新地使用手机当作IP摄像头，不仅利用了废旧手机的计算力，也降低了开发的门槛和成本。最后，我们在服务器中部署了目标识别和人群计数的AI算法，让服务器知道人在哪里，人有多少，并把结果呈现给终端用户。

With the goal of demonstrating the promising future of 5G and AI technology, and inspired by novel applications created during the COVID-19 lockdown, we implemented a real-time crowd monitoring system. The system we created isn’t like the old ones which are wired and closed-circuit. Our cameras can move around being controlled remotely via internet connection. Plus, we adopt old smartphones as IP cameras, which makes good use of old phones while minimizing the cost during development. Finally, we deploy object detecting and crowd counting AI algorithms to let server computer calculate where people are and how many instances are there in a frame so it then can return the predicted value to our end users.

**关键字：**

一、提出背景和实用价值

二、系统方案

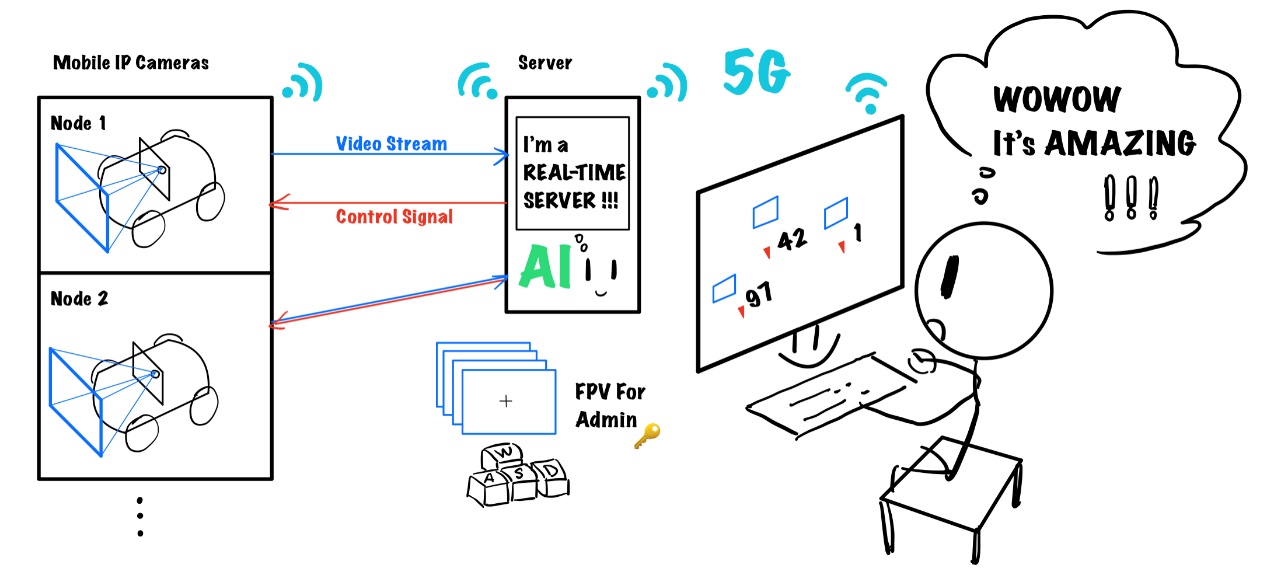


图1 系统方案概览

本团队设计的移动多节点实时人流量监控系统有两个主要部分。

第一个主要部分是图像采集端。考虑到大部分成品IP摄像头并不是为了移动而设计的，我们采用手机作为图像采集设备。由于手机体积小、重量轻，故可以轻而易举地将手机挂载到任意一个移动平台上面。移动平台可以是任何载具，如：麦克纳姆轮机器人、小型四旋翼机等。为了展示的便捷性，我们使用一个直流减速电机驱动的四轮小车作为移动平台。

第二个主要部分是服务器。在我们设计的系统中，服务器的第一个工作是接收视频、处理视频并进行人数检测。第二个工作是给系统管理员调动IP摄像头位置的通路，即让管理员能够通过键盘远程控制小车来达到移动监控设备，实现不同位置检测人流量的目的。第三个工作是将相关信息传输到网页端，让用户能够看到实际位置的监控信息，了解哪个地方人流量比较大，从而更好地规划自己的行程。在WEB方面，我们采用Flask框架来实现服务器搭建。

AI技术主要被用于人数检测，这个问题是机器学习中非常典型的问题；而5G技术则主要体现在实时视频传输、远程控制方面。但由于5G模块目前普及率不高，所以本小组用局域网来暂时代替5G网络。

三、功能与指标

1. **移动图像采集端和载具后台控制端**
2. **服务器功能**

服务器的主要功能有三：一是远程控制小车移动，二是不断接收图像并进行人数检测，三是将识别结果通过Flask框架呈现在网页上。

为了达到实时检测的性能，我们选用的目标检测算法为YOLO算法，使用Ultralytics在COCO数据集上训练的开源的YOLOv5s模型。这个模型整体大小只有15MB，而且使用CUDA加速在GTX960M（2GB）上面推理任意分辨率的一张图片平均只需要0.033s，即30.3FPS。

但是，由于COCO训练集和标注的原因，YOLO在人体重叠、俯视人群和人体非常小的时候漏检率很高。这就需要另外一种算法来应对这样的场景。我们采用的是SS-DCNet模型。SS-DCNet需要的显存比较多，若图像分辨率较大，比如推理720P分辨率的图片所需的显存基本需要2GB以上。同时，推理速度它会比较慢，在CPU上（为了防止显存不足）推理一张的图像需要0.65s。因为SSDCN不能满足实时性需求，而且它本身就是训练为在人群密度大、人数多的场景下检测人数，所以它在人体清晰，人数较少的时候非常不准。于是我们把YOLO和SS-DCNet的预测结果结合起来看，正好利用了他们互补的性质。

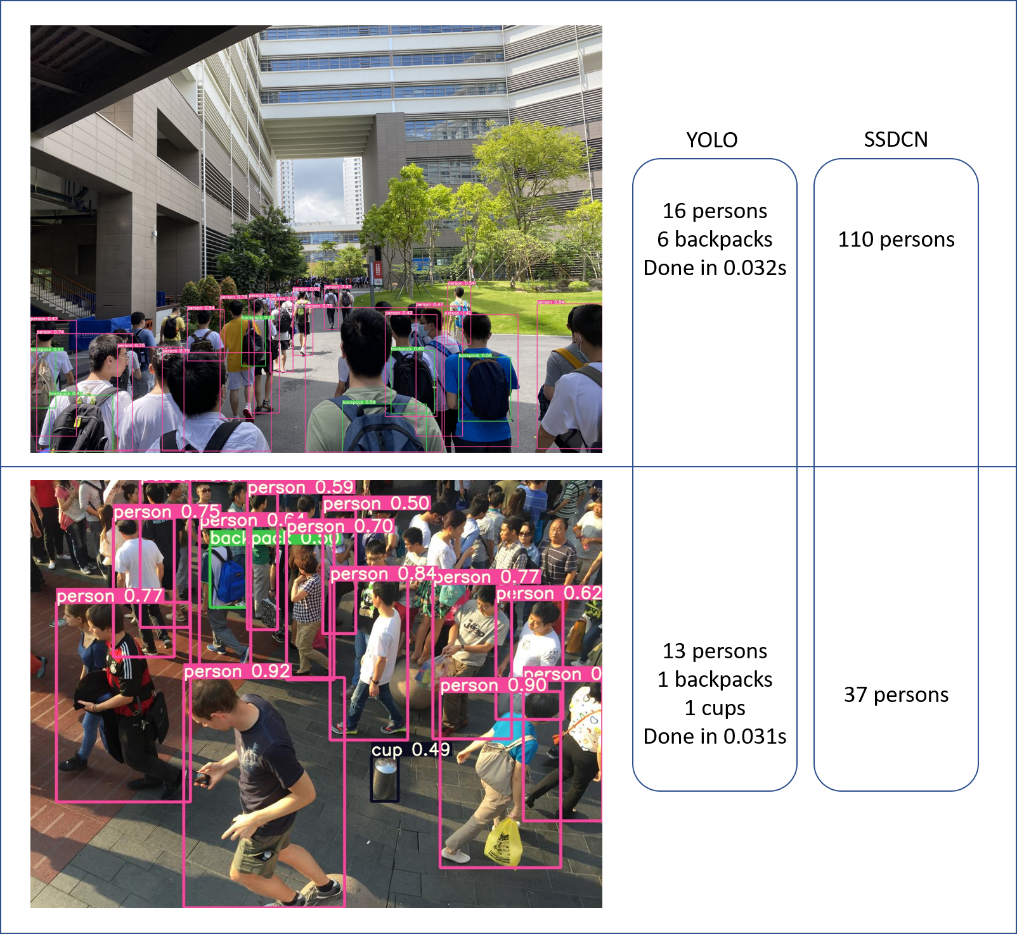


图2 YOLO和SSCDN识别结果对比

1. **网页部分**

四、实现原理

1. **目标识别与人群计数**
2. 原理中的基石：从NN到CNN

为了实现这两个功能，需要用到机器学习相关的算法，YOLO和SS-DCNet。而其中最基础的原理是神经网络。

这里以实现手写数字识别作为例子。我们使用一个三层神经网络来解决这个问题。假设输入的图像为分辨率的归一化后的灰度图，那么输入层可以设计为每一个像素对应一个感知器，于是共有784个感知器构成了输入层。输出层可以设计为10个感知器，因为我们要分类的数字只有可能是0~9中的一个。隐含层对网络的性能虽有影响，但是对原理的叙述没有影响，故这里不妨假定隐含层为单层的16个感知器。假设我们从某处得到了一组神奇的权重，这里的一组权重总共有个值，且输入层的激活值正好就是对应像素的灰度值，我们做一次前向传播，即一层一层地计算感知器的激活值，可以得到输出层的激活值，我们选取激活值最大的感知器对应的数字结果作为预测输出，这样就完成了手写数字识别的任务。其中的关键在于那一组神奇的权重，这个权重则就是通过训练学习出来的。

训练的过程本质上是优化的过程。我们一般都会根据不同的任务设计一个误差函数，即描述训练空间上预测效果与事实之间的偏差的一种函数，训练的目标就是使这个误差函数尽可能小，这显然是一个优化问题。训练开始时权重向量随机给出，做一次前向传播得到预测结果，计算出损失函数的值，然后根据一定规则往回走更新一次权重向量，即做一次反向传播，正向传播加上反向传播作为一个回合。流行的机器学习训练框架中使用的更新权重的算法一般是随机梯度下降法，如果不出差错，经过多个回合之后我们就得到了一个足够神奇的权重向量来解决预定的问题。

这种多层感知器的模型性能有限，且无法感知像素与像素间的关系，为了提取空间特征，卷积神经网络横空出世。卷积神经网络在多层感知机的基础上多了卷积层和池化层，所以最后优化的参数多了卷积核的权重。这个训练好的卷积核使得卷积神经网络可以提取图像的各种特征，所以它在解决分类问题上取得了巨大的成功，以至于YOLO和SS-DCNet都把卷积神经网络（e.g. VGG16、ShuffleNet、DarkNet）作为基础网络（backbone）。

1. YOLO：参数化目标检测

目标检测有许多算法，大概可以分为两类，第一类是两步走的算法，即先用某个网络来预测出图中有意义的物体的位置，再用CNN对目标进行分类，这类算法中的豪杰要数Faster RCNN，速度7FPS，MAP为0.73。第二类是一步到位算法，代表为YOLO和SSD，速度直接杀到30FPS以上，满足实时目标检测的需求。YOLO其实经过了很多版本的迭代了，现在官方的版本已经到了第四代，但核心思想一直都是把两步统一成一个回归问题。

一步到位指的是：将整个图片划分到的格子之中，每一个格子可以看成一个输出单元，负责预测目标框的中心点坐标和宽高、框内包括有效目标的概率和在包括有效目标条件下为各个类别的条件概率。如下图所示，每一个格子实际上负责预测两个目标框，以及20个条件概率，避免两步走的方法本质上是把问题参数化，即把位置参数和每个分类的概率参数同时预测出来。

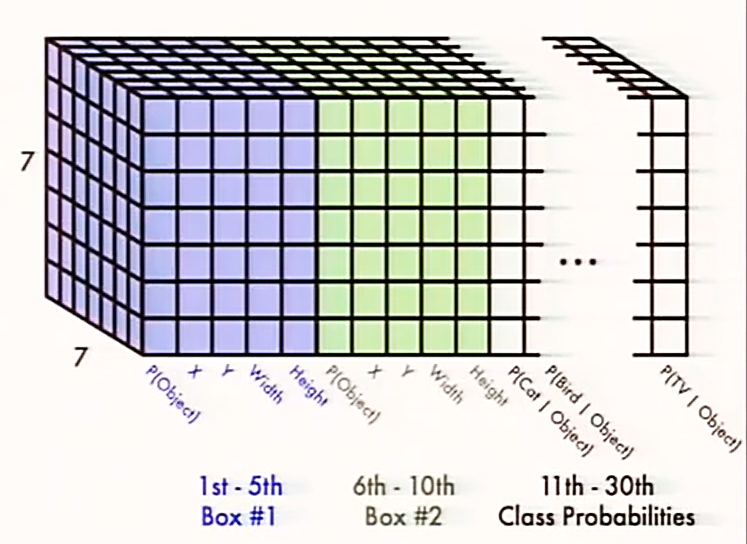


图3 网络的输出数据(CVPR2016)

假设我们已经有一个训练好的网络模型，输出层给出的则是总共7\*7\*2=98个目标框，每个目标框对应了一个物体的分类和对应的置信度，注意这里给出的分类和置信度已经是排序后取最值后的结果。接着，为了从一堆预测框中选取合理的、有意义的预测，用阈值滤除低置信度的目标框，用NMS算法滤去重合的多余的框。

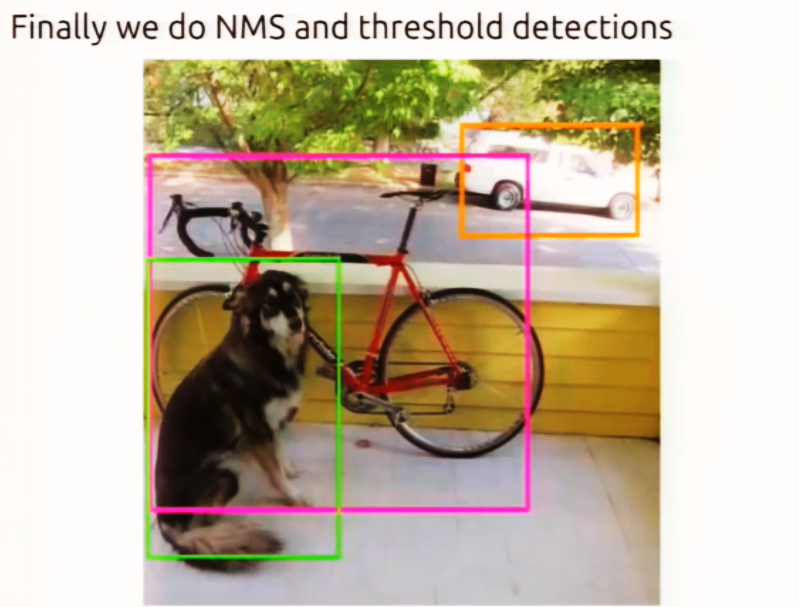
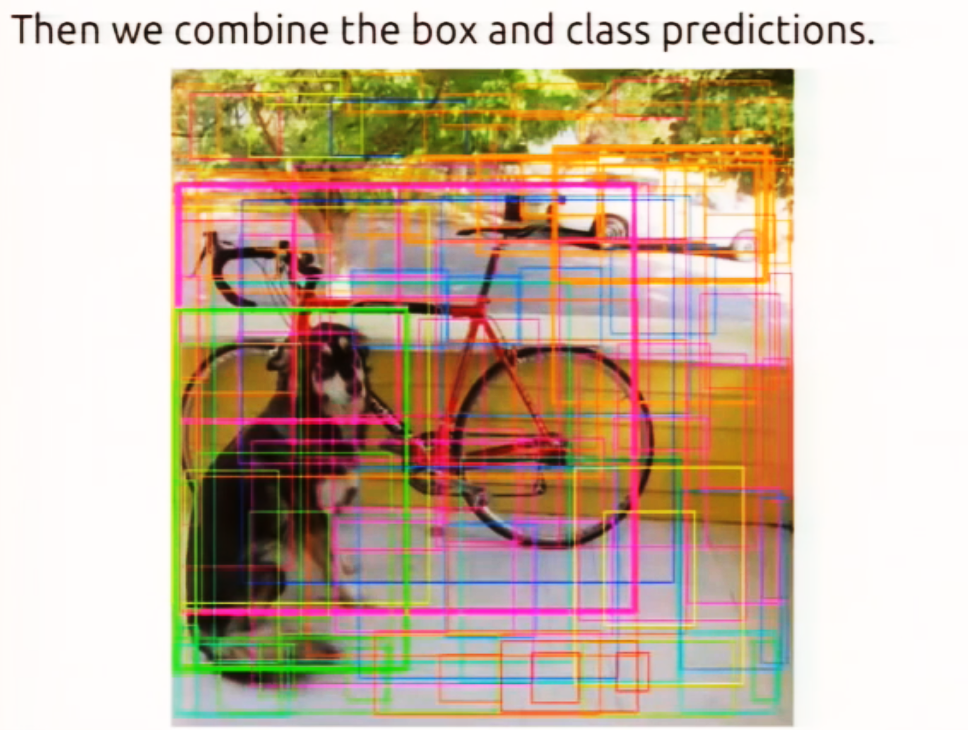


图4 所有目标框和过滤后的结果(CVPR2016)

1. YOLO的网络架构和版本迭代

网络架构是这样子的：24个穿插了池化层的卷积层——两层全连接层——输出层。卷积层的意义在于提取各种特征和物体与物体间、物体与背景间的关系。显然，这个网络的基础网络，又称特征提取网络，就是一个层数比较多的卷积神经网络。作者在论文中也指出，如果用VGG-16（另一种CNN）作为特征提取网络，识别准确度会上升但是速率会下降。

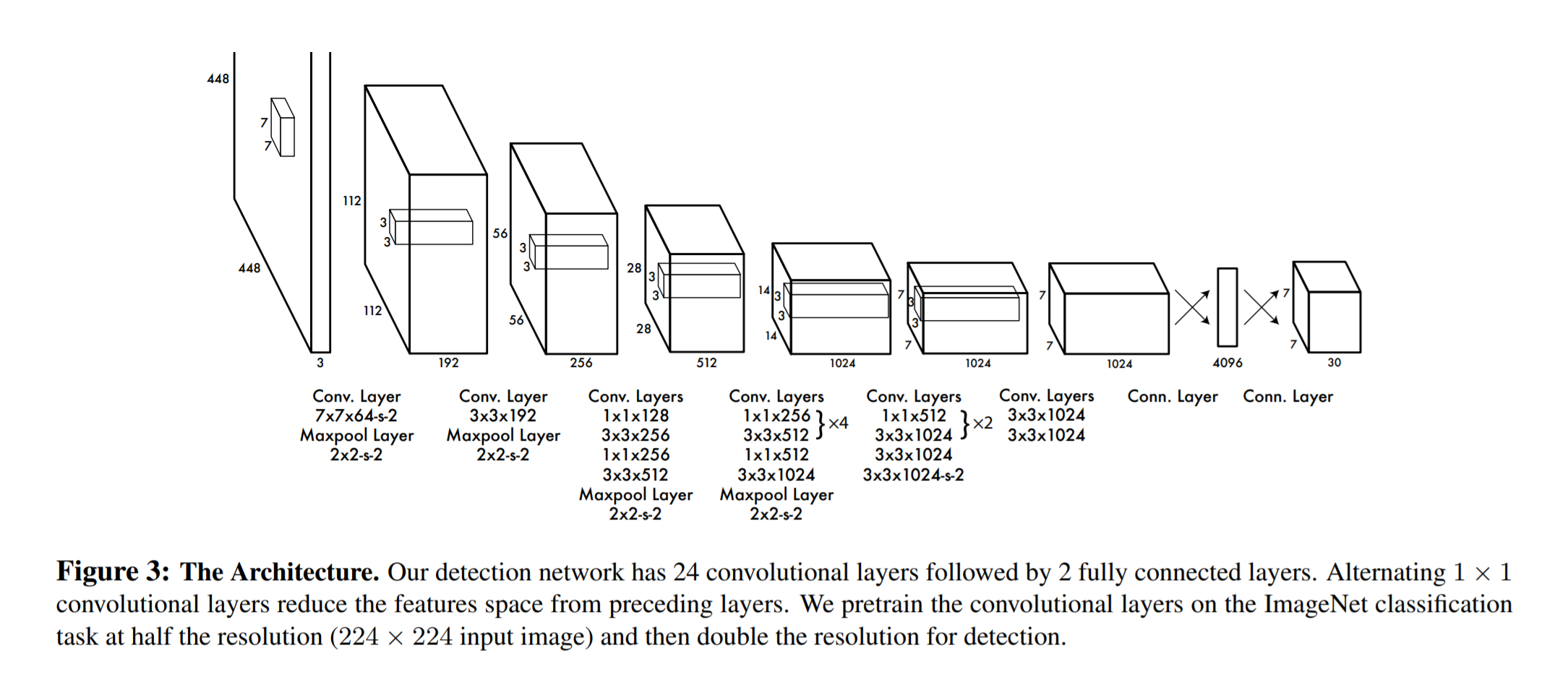


图5 YOLOv1网络架构[1]

作者在论文2.4节也明确分析了YOLO的缺点，大致复述如下。首先，YOLO对靠的很近的物体，特别是挨在一起且中点都落在同一个格子上的情况，还有很小的群体，检测效果都不好，这是因为一个网格只负责预测两个框，并且只属于一类。第二，当同一类物体出现的不常见的长宽比和其他情况时泛化能力偏弱。第三，由于损失函数的问题，定位误差是影响检测效果的主要原因，尤其是大小物体的处理上，还有待加强。

YOLO的后续版本用了各种技巧，比如添加数据集归一化层、借鉴Faster RCNN思想引入anchor box、优化特征提取网络和损失函数、细分图片网格以及各种数据集增强……YOLO随着这些技巧的应用也变得更加准确、更加快速。虽然网络的架构已经和最初差别很大了，但是“一步回归”的基本思想一直都没有变。至于本项目所用的第五版，并不是真正意义上公认的第五版，是Ultralytics在pytorch框架上的一个开源项目，训练出来的模型小，推理速度快，部署方便。基于以上几种原因，我们选用了所谓的第五版。



图6 目标检测速度对比[4]

1. 人群计数的密度图法

目前前沿的人群计数方法都是训练一个以原图像为输入、人群密度图为输出的、以CNN为基础的网络。我们对这个网络得到的密度图求和，得到一个估计的人数。下图是一个专用于人数识别的机器学习框架C3F的流程示意图，很好的描述了目前实现人群计数的一般方法。

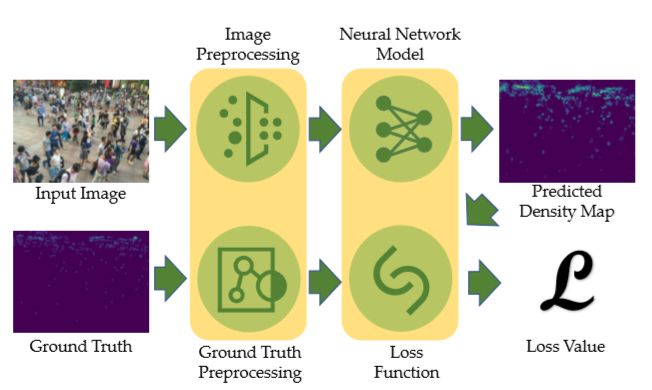


图7 人数识别的密度图法[2]

1. SS-DCNet

SS-DC意思是“有监督的空间分治法”，SS-DCNet的做法是在特征图上实现分治法，即在CNN基础网络的输出图（feature map）上实行上采样（upsampling）增大分辨率，然后对每个区域进行预测。在特征图上做分治法而不在原图上上采样的原因在于后者引入的附加计算量太大了。

有意思的是，这个网络并不是为了生成密度图，事实上它根本用不到密度图，因为它把回归问题转化为了分类问题。基本思想是人数可以被分成有限多个区间——{0}，（0，5]，……（20，inf），每个区间代表了一个类别。每个被分割的特征图被分类器分到一个人数区间，最后取中位数（最后一个区间取下界）进行求和得到最终的人数。



图8 SS-DCNet架构[3]

网络结构相对复杂，其中预测步骤也较为繁琐，篇幅所限不便展开，详情可见论文3.2节。值得注意的是，这个网络相当于一个多任务网络，所以推理耗时较多，这点在实验中也可以得到印证。

1. **计算机网络基本原理**
2. **服务器搭建的工具**

五、硬件框图

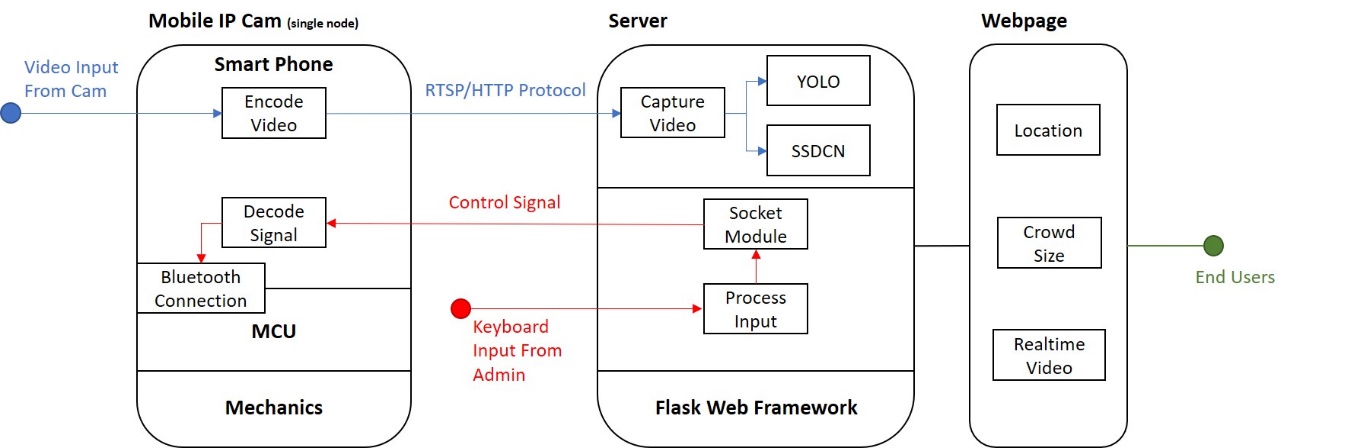


图9 系统实现框图

<电路图>

六、软件流程

1. **MCU流程图**
2. **车上的手机流程图**
3. **服务器**

七、外观设计

1. **UI设计**
2. **手机**
3. **网站**
4. **小车硬件设计**

八、系统测试

九、参考资料

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640v5 [cs.CV] 9 May 2016

[2] Junyu Gao1, Wei Lin1, Bin Zhao1, Dong Wang1, Chenyu Gao1, Jun Wen, C^3 Framework: An Open-source PyTorch Code for Crowd Counting, arXiv:1907.02724v1 [cs.CV] 5 Jul 2019

[3] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Chunhua Shen, Zhiguo Cao, From Open Set to Closed Set: Supervised Spatial Divide-and-Conquer for Object Counting, arXiv:2001.01886v2 [cs.CV] 31 May 2020

[4] https://paperswithcode.com/sota/real-time-object-detection-on-coco

十、附录