计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(3):223−228 [doi: 10.15888/j.cnki.csa.006821]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于轻量化卷积神经网络的服装分类方法①

罗梦研, 刘雁飞

(浙江理工大学 信息学院, 杭州 310018)
通讯作者: 罗梦研, E-mail: luo.mengyan@foxmail.com

摘　要: 考虑到电商平台的日益发展, 使用人工分类的方式对服装进行分类无法满足目前的需求. 本文从实际的应用场景出发, 针对于服装图像进行分类时会受到背景因素干扰、服装图像关键部位信息以及算法模型运行的的硬件要求三个方面, 分别进行改进设计. 提出: 1) 消除背景的干扰; 2) 图像局部信息的利用; 3) 模型的轻量化处理. 最终得到了在满足准确性的前提下, 可以在普通低配置 PC 端进行运行的算法模型, 提升了工作效率, 同时节省了成本.

关键词: 卷积神经网络; 图像分类; 轻量化

引用格式: 罗梦研,刘雁飞.基于轻量化卷积神经网络的服装分类方法.计算机系统应用,2019,28(3):223–228. http://www.c-s-a.org.cn/1003-3254/6821.html

Clothing Classification Method Based on Lightweight Convolutional Neural Network

LUO Meng-Yan, LIU Yan-Fei

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Considering the growing development of e-commerce platforms, the use of artificial classification of clothing classification cannot meet the current needs. In this study, based on the actual application scenarios, the design is improved in three aspects: the interference of background factors, the key position information of the garment image, and the hardware requirements of the algorithm model operation when classifying the garment image. Accordingly, it is proposed that to remove background interference, to use of local information of images, and to lightweight processing of the model. Finally, on the premise of satisfying the accuracy, the algorithm model that can be operated in the ordinary low-configuration PC terminal is obtained, which improves the work efficiency and saves the cost.

Key words: convolution neural network; clothing classification; lightweight

根据央视财经频道联合中国社科院财经战略研究院发布的《2017 中国电商年度发展报告》中国的相关数据显示, 在我国, 电商产业发展迅猛, 体量也日益庞大. 就目前而言, 电商交易额达到了全世界电商总交易额的 40% 以上. 一直以来我国网络购物市场销售占比最高的品类是服装鞋帽类, 2017 年服装鞋帽类网购率比例已达 8 成.

伴随着市场需求的增大, 一个待需解决的问题逐渐凸显出来: 电商平台对于商品往往需要对平台内的商品进行分类, 贴上合适的标签, 以便于用户通过类别进行挑选或者搜索. 但是采用人工进行分类和标注的方式往往会难以避免诸如时间和人力成本, 标注结果容易受到个体差异的影响等问题.

传统的图像分类算法主要是集中在对图像的全局特征和局部特征进行提取上, 其中全局特征又包括颜、形状、纹理等. 对于形状特征, 使用链码直方图[1]、P 阶矩阵[2]. 以及使用了几何不变矩[3]等方式. 除此之外, 有基于结构的 LBP 描述子[4]、基于频谱的 Gabor[5]

① 收稿时间: 2018-09-22; 修改时间: 2018-10-19; 采用时间: 2018-10-29; csa 在线出版时间: 2019-02-22

描述子等对纹理特征进行提取的方法. 对于局部特征而言, 可以利用 Harris[6]角点、SIFT[7]特征、HOG[8]特征进行分类. Bossard[9]利用了 HOG 和 LBP 等局部特征, 结合机器学习当中的 SVM 和随机森林等方式来进行分类.

以上这些传统图像分类算法, 在具有较高质量、没有复杂背景的服装图像中可以得到较好的效果. 但是在实际的使用场景中, 会因为背景信息的干扰, 图像采集过程不够规范, 图像质量不高等因素导致模型准确率下降, 不能很好地应用到实际的使用场景中来.

本文设计了一套基于轻量化卷积神经网络的服装分类方法. 着重针对于传统图像分类中易受背景干扰, 鲁棒性较差等缺点进行改进. 利用物体检测算法对图像中的行人进行检测, 来对输入图片进行初步的预处理, 从而将其与背景相分离, 减少无关的背景信息对分类结果产生的干扰; 为了提升模型的分类效果, 增强鲁棒性, 使用局部信息辅助进行分类. 通过对服装图片中的关键点进行定位, 从而将局部特征和全局特征进行融合, 辅助提高模型的分类效果. 考虑到卷积神经网络的运行需要较高的硬件支持, 为了拓宽模型的应用范围, 在保证模型一定的分类效果的前提下, 对模型进行了压缩处理, 极大的减少了计算量, 使得在算法模型在低配置硬件或者移动设备也可以运行.

1 模型的设计方案

本论文从实际应用场景出发, 结合当下在图像分类领域取得巨大突破的卷积神经网络算法, 设计基于卷积神经网络的服装分类算法模型. 虽然基于卷积神经网络的图像分类算法在各大数据集竞赛 (ImageNet、Kaggle 等) 都取得了不俗的成绩[10,11], 但是针对于本文所涉及的实际应用场景, 还是存在一些难题需要进行解决. 于是本论文针对一些问题进行了改进和优化, 设计了应用场景更广, 模型的鲁棒性更强的算法模型.

1.1 剔除背景信息的干扰

在分类模型实际进行使用过程中, 进行分类的图片除了包含服装信息之外, 不可避免的会包含部分背景信息, 对最终的分类结果产生极大的干扰. 为了提升模型的分类效果以及鲁棒性, 背景信息的剔除就显得尤为重要了. 本文巧妙的利用了卷积神经网络在物体检测领域的优势, 使用物体检测算法进行行人检测, 从而将图片中的行人和背景相分离. 利用物体检测比赛 MPII 中的数据集 (包括行人在内的 20 种物体的标注数据), 将其中包含行人的图片提取出来, 得到了 5717 张标注了图片中行人位置的训练数据, 用来进行行人检测算法的训练. 这里我们分析对比了主流的物体检测算法模型的实现方案, 从检测效果 mAP 和处理速度 FPS 上进行考量. 采用了 Faster-RCNN 和 YOLO3 进行测试对比, 对比结果如表 1 所示.

表 1 运行时间和 IOU 对比

Model	Faster-RCNN	YOLO3
时间 (ms)	172	22
IOU (%)	92.36	83.47

虽然 Faster-RCNN 预测的准确率较 YOLOv3 有较大优势, 但是作为模型的预处理阶段, 不宜耗费太高的计算成本和时间成本, 于是本论文选择基于 YOLOv3 算法训练出的行人检测模型来进行背景信息的移除, 减少干扰.

1.2 局部特征的提取

针对于服装图像的分类问题, 往往可以对局部信息比如 Harris[6]角点、SIFT[7]特征、HOG[8]特征进行分类. 一般而言, 对于服装图片, 其袖口、衣领、双肩等位置的特征往往包含了更多的关于服装种类的信息. 传统的基于卷积神经网络的分类算法使用整张图片作为输入, 利用若干层层卷积层的叠加进行全局特征的提取, 然后再利用分类器进行分类操作. 为了对服装图片当中的局部信息加以利用, 论文考虑设计服装图片的关键点定位算法, 用来定位到服装图片当中的衣领、袖口等部位, 帮助进行局部特征的提取操作.

对于关键点定位问题, 本文对传统的卷机神经网络进行改进, 一方面借鉴了 Residual Learning 中的“跳层连接”, 对模型结构进行改进.

如图 1 所示, 首先通过卷积核大小为 1×1 的卷积操作调整 Feature map 的通道数量, 减少计算量. 然后再经过卷积核大小为 3×3 的卷积操作, 最终输出的 Feature map 的同道数量再经过 1×1 大小的卷积核进行调整.

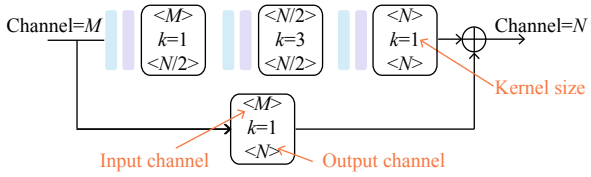


图 1 “跳层连接”结构示意

除此之外, 增加了一条由 1×1 的卷积核构成的卷积层作为旁路, 将输入该卷积模块的 Feature map 分别通过两条同道进行计算, 最终的结果进行通道上的叠加. 一方面可以保存原 Feature map 在更多尺度下的特征信息, 另一方面增加了旁路连接可以更好的帮助梯度进行传递, 帮助模型进行训练.

为了更好的处理不同尺度的特征向量, 一些常见的做法是针对不同尺度的特征分别进行处理, 然后再将处理的结果进行融合和叠加[12,13]. 本文则是采用了通过多次使用短连接的方式不断的将不同尺度的信息进行融合.

在通过堆叠的卷积模块进行特征提取的同时, 为了提升对多尺度信息的处理能力, 增加了若干层旁路连接, 多次将不同尺度的特征信息进行融合, 图中的橘黄色向下箭头表示使用 Max-Pooling 进行下采样操作, 橘黄色向上的箭头表示通过使用双线性插值的方式进行上采样. 从而产生了不同尺度下的特征信息, 很好的兼顾到了由于图片中衣服大小不一而导致的信息提取困难的问题.

考虑到对于不同种类的服装图像, 其图片中所含的关键点数量也不一致. 而 Convolutional Pose Machine 和 Hourglass 只能对图片中固定数量的关键点进行定位. 于是对上述两种算法的 *Loss* 进行改进, 综合考虑到各种不同服装图像中关键点数量, 每张图片中输出 20 个关键点位置进行损失函数的计算, 不存在的关键点的 *Loss* 值设置为 0. 采用了 Online Hard Keypoints Mining (CPN), 仅保留损失值较大的关键点进行回传, 阈值设置总关键点数量的一半.

1.3 卷积神经网络的轻量化处理

基于卷积神经网络的算法模型, 由于堆叠了大量的卷积运算, 模型的训练和使用阶段都需要较高硬件环境的支持, 由此而导致了时间成本和硬件成本的提升. 目前深度学习模型的研究主要也是集中在了提升模型的效果和减少参数和计算量这两个方面. 考虑到为了节省运行本设计所需要花费的硬件成本 (如果需要花费较高价格购买高配置的设备, 则与本设计节省成本的初衷相违背). 为了使得所设计的算法模型有更多的应用场景, 势必需要对本设计的模型采用轻量化的模型结构, 对计算量进行压缩. 以便于在保证一定准确度的前提下尽量减少运算开销和时间成本.

当使用 1×1 的卷积核进行卷操作时, 会使得输出的 Feature map 的尺寸不发生变化, 而只会改变 Feature map 的通道数. 所以卷积核大小为 1×1 的卷及操作往往被用来对 Feature map 的大小进行调节. 在进行卷积操作之前先对 Feature map 的通道数量进行压缩, 从而可以一定程度上减少计算量和参数量. GoogleNet 和 ResNet 均采用这种方式对模型的计算量进行一定程度的削减. 轻量化模型结构 SqueezeNet 中的 Fire modul 模块通过这种方式达到了轻量化模型的设计.

压缩 Feature map 通道的方式虽然可以大幅减少模型的计算量, 但是由于在压缩过程中往往会丢失部分数据信息, 对最终的模型结果产生一定的影响. 于是考虑从卷积操作的方式上进行改进. 将尺度为 $D_k \times D_k \times M$ 的 (D_k 为 size. M 为通道数量)Feature map 切分成 $M/2$ 个独立的 Feature map, 每个 Feature map 的通道数均为 2. 然后分别进行卷积核大小为 3×3, 输出通道数为 2, Padding 为 1 的卷积操作, 得到的结果按顺序进行通道上的叠加. 得到尺度为 $D_k \times D_k \times M$ 的 Feature map. 这种卷积方式称之为“分块卷积”具体操作如图 2 所示.

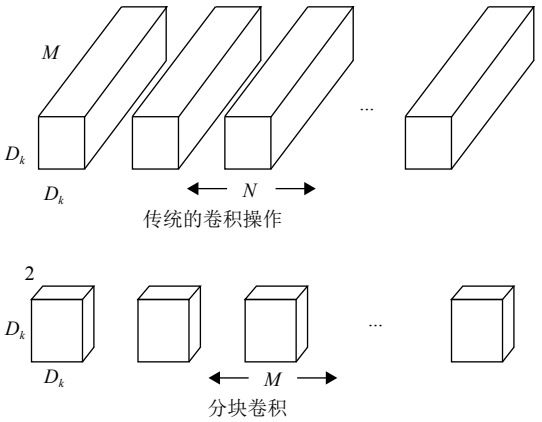


图 2 分块卷积操作示意

使用传统的卷积方式, 一次卷积操作的计算量为 (卷积核大小为 D_f):

$$C_1 = D_k \cdot D_k \cdot M \cdot N \cdot D_f \cdot D_f$$

由于分块卷积是将 Feature map 分离开来进行操作, 会在一定程度上减少 Feature map 不同通道的信息之间的相互关联性. 于是在进行卷积操作之后通过 1×1 大小的卷积核进行卷积操作. 有三点作用: 1) 帮助不同通道之间的信息进行融合. 2) 可以对通道数量进行融合. 3) 可以对通道数量进行调节 (输出的同道数量

设置为 N). 于是完整的一次分块卷积操作所需要的计算量大小为:

$$C_2 = 2 \cdot D_k \cdot D_k \cdot M \cdot D_f \cdot D_f + M \cdot N \cdot D_f \cdot D_f$$

$$\frac{C_2}{C_1} = \frac{2 \cdot D_k \cdot D_k \cdot M \cdot D_f \cdot D_f + M \cdot N \cdot D_f \cdot D_f}{D_k \cdot D_k \cdot M \cdot N \cdot D_f \cdot D_f} = \frac{1}{N} + \frac{2}{D_k^2}$$

现在最常用的卷积核大小为 3×3, 即则计算量减少的到原来的 1/4 以内 (N 一般远大于 9, 可忽略不计).

2 算法实现及结果对比

2.1 训练与测试数据

训练数据采取的是 Large-scale Fashion (Deep Fashion) Database 数据集, 其中针对于服装的种类, 材质, 风格等方面进行的划分. 关于服装种类的标注数据有 20 万张. 足以满足本算法的需求.

2.2 不同结构模型效果对比

在 Ubuntu16.04、CUDA9.0 环境下, 采用 Pytorch 框架进行模型结构的搭建. 首先采用了 DenseNet-121、DenseNet-169、DenseNet-201、DenseNet-264 模型进行测试. 为了使得输出尺寸大小不收限制, 将最终的输出层采用 Global Average Pool+1×1 的卷积层 (卷积层输出通道数为分类数量) 进行替换. 模型结构如图 3 所示.

Layers	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	7×7 Conv, stride 2			
Pooling	3×3 Max pool, stride 2			
Dense block (1)	[1×1 conv, 3×3 conv] ×6	[1×1 conv, 3×3 conv] ×6	[1×1 conv, 3×3 conv] ×6	[1×1 conv, 3×3 conv] ×6
Transition layer (1)	1×1 conv 2×2 Average pool, stride 2			
Dense block (2)	[1×1 conv, 3×3 conv] ×12	[1×1 conv, 3×3 conv] ×12	[1×1 conv, 3×3 conv] ×12	[1×1 conv, 3×3 conv] ×12
Transition layer (2)	1×1 conv 2×2 Average pool, stride 2			
Dense block (3)	[1×1 conv, 3×3 conv] ×24	[1×1 conv, 3×3 conv] ×32	[1×1 conv, 3×3 conv] ×48	[1×1 conv, 3×3 conv] ×64
Transition layer (3)	1×1 Conv 2×2 Average pool, stride 2			
Dense block (4)	[1×1 conv, 3×3 conv] ×16	[1×1 conv, 3×3 conv] ×32	[1×1 conv, 3×3 conv] ×38	[1×1 conv, 3×3 conv] ×48
Classification layer	7×7 Global average pool			

图 3 模型结构

训练过程中, 使用了 Color Jittering, Random Crop Resize, Rotate 等数据增强方法, 对比了使用行人检测消除背景前后的分类结果差异.

使用以上几种模型结构进行训练并且测试, 测试结果如图 4 所示.

2.3 关键点定位效果对比

针对关键点定位问题, 采用这里采用的是阿里天池大赛中服装标注点定位比赛里所采用的衡量标准 NE 值来对模型的定位效果进行衡量和比较, 数值越大则代表定位效果越好.

$$NE = \frac{\sum_k \left\{ \frac{d_k}{s_k} \delta(v_k = 1) \right\}}{\sum_k \{\delta(v_k = 1)\}} \times 100\%$$

其中, k 为关键点编号, d_k 表示预测关键点和标注关键点的距离, s_k 为距离的归一化参数 (上衣、外套、连衣裙为两个腋窝点欧式距离, 裤子和半身裙为两个裤头点的欧式距离), v_k 表示关键点是否可见.

1) Convolutional Pose Machine 进行关键点定位, 使用了 6 个 stage 的 CPM 训练之后 NE 为 13.37%.

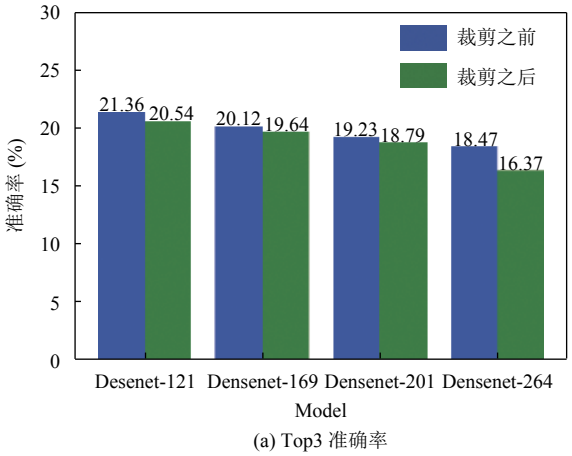
2) 尝试使用 Hourglass 进行定位, 仅使用了翻转进行了数据增强, NE 值为 10.78%.

3) 增加了 Color Jittering, Random Crop Resize, Rotate 等数据增强方法, 降低到 8.8.

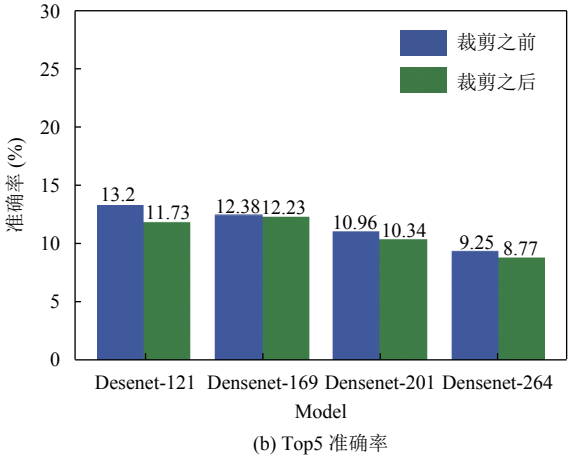
4) 使用 Hourglass 在 MPII 比赛上的模型参数进行参数初始化, 降低到 7.86%.

5) 仅训练单个模型进行预测, 对 $Loss$ 进行修改,

不存在的关键点 *Loss* 为 0, 采用了 CPN 中提出的 Online Hard Keypoints Mining (OHKM), 仅保留损失较大的关键点进行回传. 之前训练过的模型进行权重的初始化, 预测时通过 Yolo 进行裁剪, *NE* 降低到 6.76%.



(a) Top3 准确率



(b) Top5 准确率

图 4 不同模型结构测试准确率对比

使用以上 5 种方式进行关键点定位, 测试得到的 *NE* 值对比, 如图 5 所示.

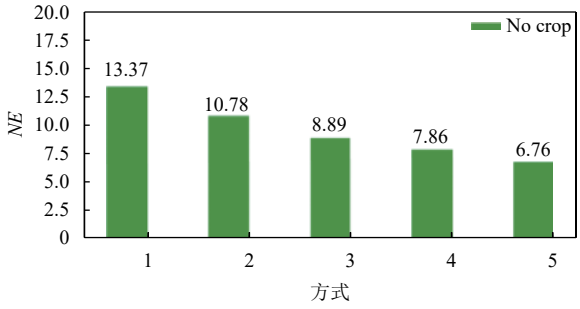


图 5 5 种不同关键点定位的 NE 值对比

使用关键点定位算法来进行局部特征提取, 对原模型的提升效果如图 6 所示.

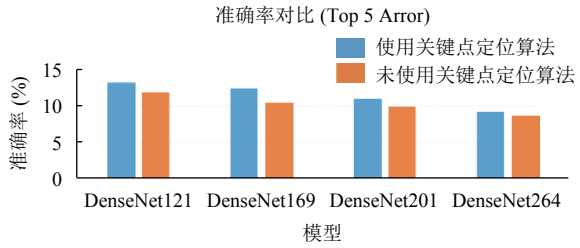
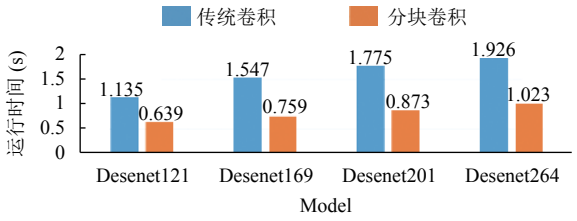


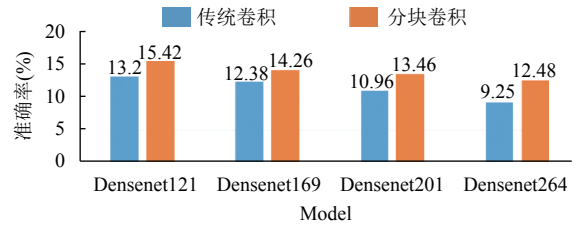
图 6 关键点定位对模型准确率效果提升对比

2.4 模型轻量化的实现及效果对比

考虑到大多数应用场景下无法提供较高的硬件设备支持, 故使用入门级的硬件设备进行测试对比. 本文采用的是 Inter Core i5 7300H CPU 搭配 NVIDIA 940mx 低端入门级显卡进行测试. 使用 CUDA9.0 搭配 Pytorch 框架进行模型结构的搭建. 对上述采用的模型进行测试, 得到不同模型结构进行预测时所花费的时间对比. 如图 7 所示.



(a) 运行时间对比



(b) 准确率对比 (Top 5 Error, no crop)

图 7 分块卷积方式和传统卷积方式在不同的网络结构下的运行时间以及准确率 (均未使用数据增强操作和背景信息的剔除操作)

从图中的实验数据我们可以看到, 使用分块卷积来替代传统的卷积神经网络中的卷积方式, 可以大幅度减少进行计算时间, 加快模型的训练以及预测阶段的速度. 降低了对硬件设备的要求, 从而可以节约成本, 以便于拓宽应用场景. 除此之外, 使用分块卷积的方式也减少了部分权重参数, 节约了存储开销. 当然, 不可避免的是, 随着模型参数的减少, 模型的拟合能力也随之下降, 准确率也有了一定的损失. 但是减少模型的参

数量, 在一定程度上降低了模型的复杂程度, 从而起到减少过拟合的作用, 使得模型的可迁移性更强, 适用于更多场景下采集的服装图像.

3 总结

通过以上的理论论述和实验部分的结果, 得到了我们最终的模型设计方案. 首先使用基于 YOLOV3 算法训练出的行人检测模型来对输入数据进行预处理操作, 减少背景信息对最终结果造成的干扰. 将进行裁剪后的图像输入关键点定位模型中, 得到关键点的位置信息, 与全局信息一同传入卷积神经网络中进行特征提取, 辅助提高模型的预测效果以及算法模型的鲁棒性. 除此之外, 使用分块卷积对模型进行轻量化的处理, 可以帮助在较差的硬件环境下进行使用, 提高算法模型的运行效率.

本文从实际的应用需求出发, 分析了使用计算机辅助进行服装类别的分类的意义以及可行性. 利用当下在图像分类领域取得极好效果的卷积神经网络进行尝试和研究. 同时结合了实际的应用场景, 从背景信息干扰, 对关键部位信息的利用以及考虑到硬件条件限制这三个角度, 设计了三种辅助操作, 帮助提升基础分类模型 DenseNet 在服装类别分类这一应用场景下的效果进行提升.

最终得到的实验结果可以基本满足日常的使用需求. 通过一系列的改进和优化, 极大程度的减少了拍摄场景对最终的分类效果的不良影响. 很好的拓宽了模算法模型的应用场景, 很多在非幕布场景下拍摄的图片也可以取得很好的分类效果. 除此之外, 在普通的 PC 设备上运行基本可以达到每张 1~2 s. 在训练所使用的 1080Ti 显卡上进行计算可以达到 4FPS 的速度, 相对于使用人工进行分类的方式更加的高效快捷. 可以很好替代人工分类, 或者提供辅助.

参考文献

1 Iivarinen J, Peura M, Särelä J, *et al*. Comparison of combined shape descriptors for irregular objects. Proceedings of the 8th British Machine Vision Conference. Essex, Great Britain. 1997. 430–439.

2 Rangayyan RM, El-Faramawy NM, Desautels JEL, *et al*. Measures of acutance and shape for classification of breast tumors. IEEE Transactions on Medical Imaging, 1997, 16(6): 799–810. [doi: 10.1109/42.650876]

3 Teh CH, Chin RT. On image analysis by the methods of moments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988, 10(4): 496–513. [doi: 10.1109/34.3913]

4 Ojala T, Pietikäinen M, Harwood D. A comparative study of texture measures with classification based on featured distributions. Pattern Recognition, 1996, 29(1): 51–59. [doi: 10.1016/0031-3203(95)00067-4]

5 Daugman JG. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1988, 36(7): 1169–1179. [doi: 10.1109/29.1644]

6 Girshick R. Fast R-CNN. arXiv: 1504.08083, 2015.

7 Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91–110. [doi: 10.1023/B:VISI.0000029664.99615.94]

8 Dalal N, Triggs B. Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. 2005. 886–893.

9 Bossard L, Dantone M, Leistner C, *et al*. Apparel classification with style. Asian Conference on Computer Vision. Daejeon, South Korea. 2012. 321–335.

10 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 770–778.

11 Tompson JJ, Jain A, LeCun Y, *et al*. Joint training of a convolutional network and a graphical model for human pose estimation. Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal, Canada. 2014. 1799–1807.

12 Wei SE, Ramakrishna V, Kanade T, *et al*. Convolutional pose machines. 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 4724–4732.

13 Huang G, Liu Z, Laurens VDM , *et al*. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2017. 2261–2269.