目录

第 1 章 绪论

1.1 研究背景及意义

随互联网技术和信息科技进步，人脸识别算法进行了深入研究。目前，人脸识别算法的运用较为宽泛，存在于现实生活中各个方面，智能考勤系统则是人脸检测和识别算法的一项重要体现。人脸考勤作为一种智能化的考勤方式，有着重要的实际意义和使用价值，例如学生的上课签到，考试检查等等。随着教育的发展，学生人数也是大量增加，学生的考勤管理就显得非常重要，同时目前的考勤方式也难以满足大量人员管理的需求。因此，作为智能化的人脸考勤系统正是广大教育机构的需求，其系统及技术的发展也需要不断的优化和提高。

所谓人脸考勤系统，就是通过对摄像头实时的画面进行人脸检测，然后利用人脸识别算法，提取人脸图像中信息特征，将得到的信息特征与存储的人脸信息进行对比和匹配，完成人脸的身份认证，若辨认出身份并作好记录，表示考勤完成，反之则考勤失败。

在人脸检测和识别的研究中，需要考虑到人脸信息作为图像特征所带来的困难和挑战，一方面，需要考虑到人脸结构、面部形状相似所产生的类间微小变化，通过获取隐性特征的强判别性来增强这种类间变化，进而识别不同个体。另一方面，需要考虑到系统的安全性，避免非法用户使用含有合法用户人
脸的图片、视频、3D 人脸模型等进行假冒考勤。传统人脸识别系统主要有以下几个部分：人脸图片收集、图像前处理、人脸特征提取、特征匹配，但是此系统中缺少了人脸活体检测环节，无法准确辨别图像是否为真实人脸，人脸活体检测模块应该加入人脸识别系统，来保证人脸识别系统的安全性。

1.2 本文主要内容

本文根据智能考勤系统对人脸识别的需求，通过深入分析与研究人脸检测、活体检测和人脸识别算法，选择基于 Retinaface 的人脸检测算法、基于 MiniFASNet 的静默活体检测算法和基于 MobileFaceNet 的人脸识别算法构建本文人脸识别考勤系统，进而完成了系统设计。

通过相关算法整理和技术分析，设计和测试以人脸检测、活体检测与人脸识别算法为基础的智能考勤系统。本系统的整体算法设计如图 1.1 所示。

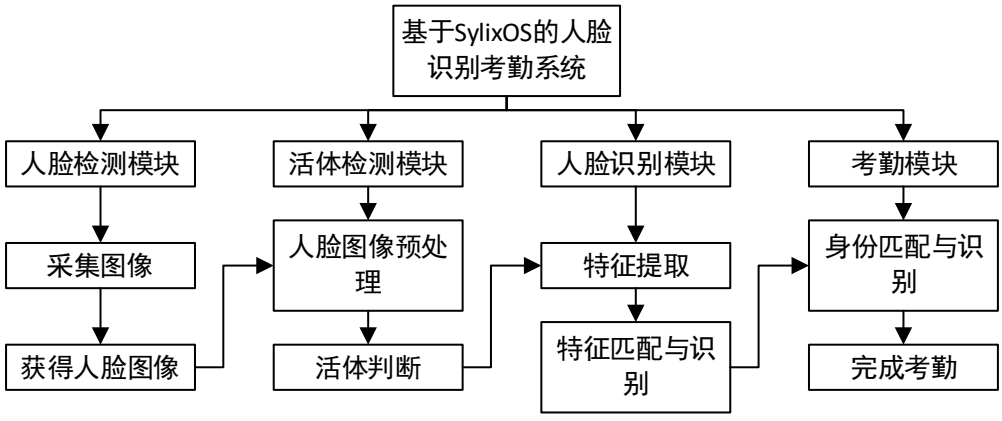


图 1.1 系统算法整体设计

第 2 章 人脸检测模块

复杂人脸复合了多种现实生活中常见的会为检测带来挑战的场景，如小人脸、遮挡、光源干扰造成的低质量成像等等，对模型的特征处理能力提出更高的要求，而模型的现实场景应用也对模型提出实时性检测的需求。

人脸检测算法根据特征处理方法的不同，分为双阶段检测器和单阶段检测器。双阶段检测器以 R-CNN 及其衍生网络为代表，检测时先经一次分类检测，滤除无关背景，生成人脸候选框，再用过滤得到的候选框作为样本进行模型训练。这种方法使用两阶段级连的方式，先粗后精的回归定位，优点是滤除背景信息，使精确度更好。然而，双阶段检测器也存在明显不足，即信息损失和检测速度低。

单阶段检测以 RetinaFace 为代表，直接对输入图片进行卷积运算，全图直接参与的检测特点使其速度更快，对信息的利用也更加全面。现阶段高精度人脸检测模型结构较为复杂，对计算资源要求较高，在移动端和边缘设备等现实场景应用受到一定的局限。RetinaFace 是当前性能最优的单阶段人脸检测模型之一。

2.1 基于 RetinaFace 的人脸检测

RetinaFace 是一种专门检测人脸的单阶段目标检测算法，在多级特征图信息的基础上使用了特征金字塔网络（FPN，Feature Pyramid Networks）的方法，更充分地使用特征信息，网络结构如图 2.1 所示。RetinaFace 使用多级特征的特征图（P2～P6）构成特征金字塔，在不同层级的特征图上设计了大小不同、数量众多的锚点框，使其获得了出色的检测性能。

RetinaFace 的主干网络使用了 ResNet152 网络结构，特征金字塔中的 P2～P5 层级对应 ResNet152 的 C2～C5 层级残差网络模块，P6 则是使用步长为 2、卷积核大小为 3×3 的卷积对 C5 的特征图进行卷积计算后得到的特征图。P6 层级采用 Xavier 方式对参数进行随机初始化。并对特征金字塔的每一层使用独立的语义模块，提高感受野并增强刚性语境的建模能力。其通过单级像素，分析人脸特征，采用额外监督和自监督对人脸进行分类和锚框检测。通过每个锚框对人脸进行打分，并定位到两个眼睛，一个鼻子，一个嘴巴的采用五官人脸标志，并将不容易处理的密集 3D 人脸顶点投射在一个平面上，进行降维处理。

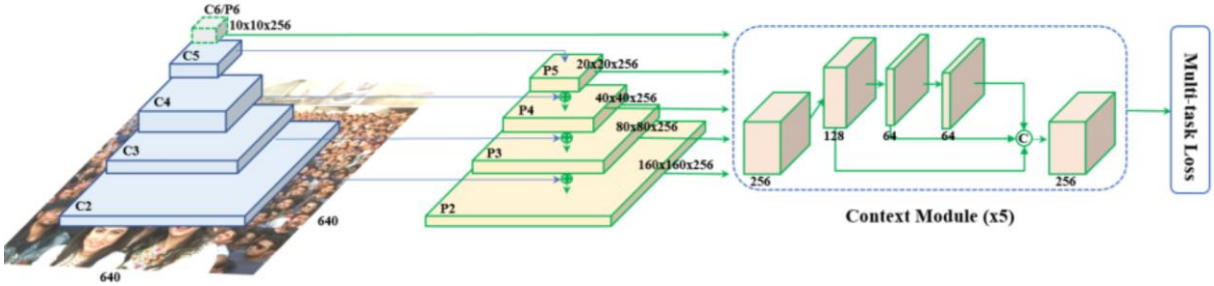


图 2.1 RetinaFace 算法网络结构

2.1.1 特征金字塔

Retinaface 采用了从 P_2 到 P_6 的特征金字塔层次，其中 P_2 到 P_5 是使用自上而下和横向连接，从相应的 ResNet 剩余阶段(C_2 到 C_5)的输出计算出来的，P_6 是通过 C_5 上 stride=2 的 3×3 卷积来计算的。C_1 到 C_5 是经过训练的 ResNet-152，分类网络在 ImageNet-11k 数据集，同时 P_6 用“Xavier”方法随机初始化。

2.1.2 上下文连接模块

在 SSH PyramidBox 的基础上，在五个特征金字塔层次上应用独立的上下文连接模块，来增加特征的全局感受野和增强上下网络的模型表达能力。从冠军 widerface 身上吸取经验，将横向连接和上下文模块中的所有 3×3 卷积层替换为可变形卷积网络，进一步增强了非刚性上下文建模能力。

2.1.3 损失头

对于负锚，只应用分类损失。对于正锚，计算提出的多任务损失。采用一个共享损失头，在不同的特征映射 Hn×Wn×256，n∈{2，…6}。对于网格解码器，应用预训练的模型，通过预训练模型来减少计算开销。

2.1.4 锚设置

如表 2.1 所示，在从 P_2 到 P_6 的特征金字塔层次，使用了尺度注意力机制锚。在这里，P_2的设计是通过平铺小锚来捕捉微小的面，但代价是更多的计算时间和更多的假阳性风险。把刻度步长设为 21/3，宽高比为 1:1。输入图像大小为 640×640 时，锚点可以覆盖 16×16 到 406×406 的特征金字塔水平。总的来说，有 102300 个锚，其中 75%来自 P_2。

表 2.1 特征金字塔锚框设置

特征金字塔	步长	锚框
P_2	4	16, 20.16, 25.40
P_3	8	32, 40.32, 50.80
P_4	16	64, 80.63, 101.59
P_5	32	128, 161.26, 203.19
P_6	64	256, 322.54, 406.37

2.1.5 多任务损失函数

RetinaFace 采用多任务损失函数，对于训练阶段的每一个锚点框，使用优化算法寻求如式 2.1 的多任务损失函数的最小化。

$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l, l_i^*) + \lambda_3 p_i^* L_{pixel} \quad (2.1)$$

公式中$L_{box}(t_i, t_i^*)$为人脸分类的损失函数，其中p_i；是锚框对于第i张特征图是否是人脸的预测概率，p_i^*；是对锚框的打分，当$p_i^* = 1$时代表是正锚框，当$p_i^* = 0$时代表是负锚框。人脸分类损失L_{cls}的具体是二进制形式的 softmax 损失函数，损失函数的评价结果就是是人脸和不是人脸两种。$L_{box}(t_i, t_i^*)$为锚框回归损失，其中$t_i = \{t_i, t_j, t_w, t_h\}$, $t_i^* = \{t_i^*, t_j^*, t_w^*, t_h^*\}$，$x$，$y$ 是锚框的中心点坐标，w 和 h 是锚框的宽和高。t_i是预测框的坐标，而t_i^*是真实框的坐标，也是在训练集标注时候的坐标。采用 Fast RCNN 的方法来规范方框回归目标(即中心位置、宽度和高度)，$box(t_i, t_i^*) = R(t_i - t_i^*)$，$R$ 是 Fast RCNN 里面定义的鲁棒损失函数(smooth-Ll)。$L_{pts}(l, l^*)$为人脸关键点回归损失，和锚框类似，分别表示人脸中的五个关键点的预测坐标和五个关键点的真实坐标。L_{pts}采用基于坐标点的目标归一化。密集区域回归损失L_{pixel} 如下公式所示，

$$L_{pixel} = \frac{1}{W * H}\sum_{i}^{W}\sum_{i}^{H}\left\|R\left(D_{pst}, P_{cam}, P_{ill}\right)_{i,j} - I_{i,j}^{*}\right\| \tag{2.2}$$

其中D_{pst}是特征形状和纹理参数，P_{cam}是相机参数（相机位置，焦距等），P_{ill}是点光源的位置、颜色值和环境光的颜色，R 代表一个从以上参数渲染得到的人脸图像，通过L_{pixel}函数计算渲染的图像和原始图像的差异。W，H 分别为锚框里面像素的宽度和高度，$I_{i,j}^{*}$是锚框里面的像素点。

2.2 基于 OpenCV 级联分类器的人脸检测

2.2.1 基于 OpenCV 的 Harr 级联分类器

Haar 特征最先由 Paul Viola 等人提出，后经过 Rainer Lienhart 等扩展引入 45°倾斜特征，成为现在 OpenCV 所使用的的样子。图 2.2 展示了目前 OpenCV（2.4.11 版本）所使用的共计 14 种 Haar 特征，包括 5 种 Basic 特征、3 种 Core 特征和 6 种 Titled(即 45°旋转)特征。总的来说，Haar 特征分为三类：边缘特征、线性特征、中心特征和对角线特征，组合成特征模板。特征模板内有白色和黑色两种矩形，并定义该模板的特征值为白色矩形像素和减去黑色矩形像素和。

Haar 特征值反映了图像的灰度变化情况。当其作为脸部的特征提取时，能由矩形特征进行描述，例如眼睛要比脸颊颜色要深，鼻梁两侧比鼻梁颜色要深，嘴巴比周围颜色要深等特征。但矩形特征只对一些简单的图形结构，如边缘、线段较敏感，所以只能描述特定走向（水平、垂直、对角）的结构.对于下图所示的 x2, y2 和 y3 这类特征，特征数值计算公式为：v=Sum 白-Sum 黑，而对于 x3 来说，计算公式如下：v=Sum 白-2*Sum 黑；之所以将黑色区域像素和乘以 2，是为了使两种矩形区域中像素数目一致。

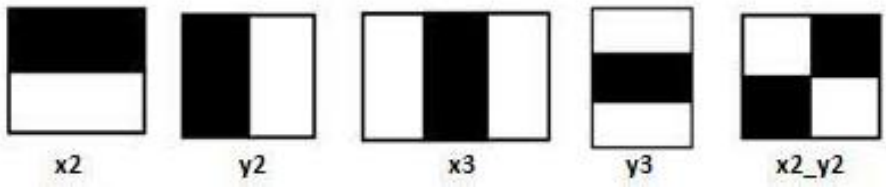


图 2.2 Harr 的五种类特征

通过改变特征模板的大小和位置，可在图像子窗口中穷举出大量的特征。上图的特征模板称为“特征原型”；特征原型在图像子窗口中扩展（平移伸缩）得到的特征称为“矩形特征”；矩形特征的值称为“特征值”。

通过调用训练程序 opencv_trancascade.exe 进行模型训练，一般选用 BASIC 模式，即训练内容包含如上图所示的五种特征。在实际中，Haar 特征可以在检测窗口中由放大+平移产生一系列子特征，但是白：黑区域面积比始终保持不变。

如下图 2.3 所示，以 x3 特征为例，在放大+平移过程中白：黑：白面积比始终是 1:1:1。首先在红框所示的检测窗口中生成大小为 3 个像素的最小 x3 特征；之后分别沿着 x 和 y 平移产生了在检测窗口中不同位置的大量最小 3 像素 x3 特征；然后把最小 x3 特征分别沿着 x 和 y 放大，再平移，又产生了一系列大一点 x3 特征；然后继续放大+平移，重复此过程，直到放大后的 x3 和检测窗口一样大。这样 x3 就产生了完整的 x3 系列特征。

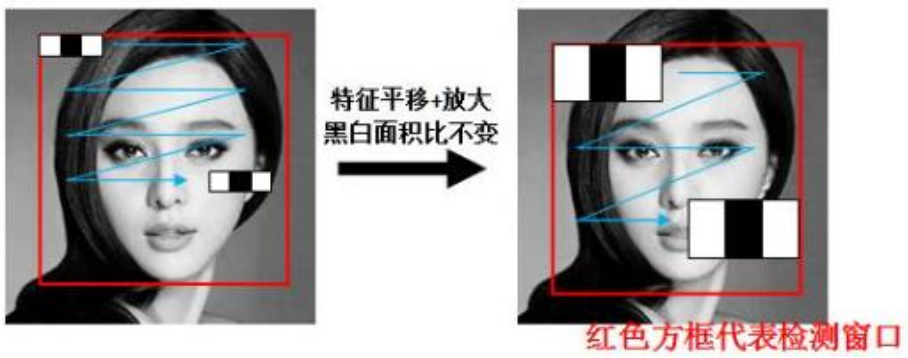


图 2.3 Harr 上 x3 特征提取

当经过一列列特征提取后，会生成大量的 Harr 特征，接下来就是计算其特征值。Opencv 中是通过如下公式对图像特征值进行计算：

$$featureValue(x) = weight_{all} \times \sum_{PixelAll} Pixel + weight_{back} \times \sum_{Pixelblack} Pixel \qquad (2.3)$$

式中对于 x3 和 y3 特征，weightall = 1，weightblack = -3。

对应的，在保存分类器的 OpenCV XML 文件中，每一个 Haar 特征都被保存在 2~3 个形如<x y width height weight>的标签中，其中 x 和 y 代表 Haar 矩形左上角坐标（以检测窗口左上角为原点），width 和 height 代表矩形的宽和高，而 weight 则对应了上面说的权重值，例如图 6 中的左边 x2 类型的 Haar 特征应该为<4 2 12 8 1.0>（整个 Haar，权重 1）和<4 2 12 4 -2.0>（黑色区域，权重-2）。

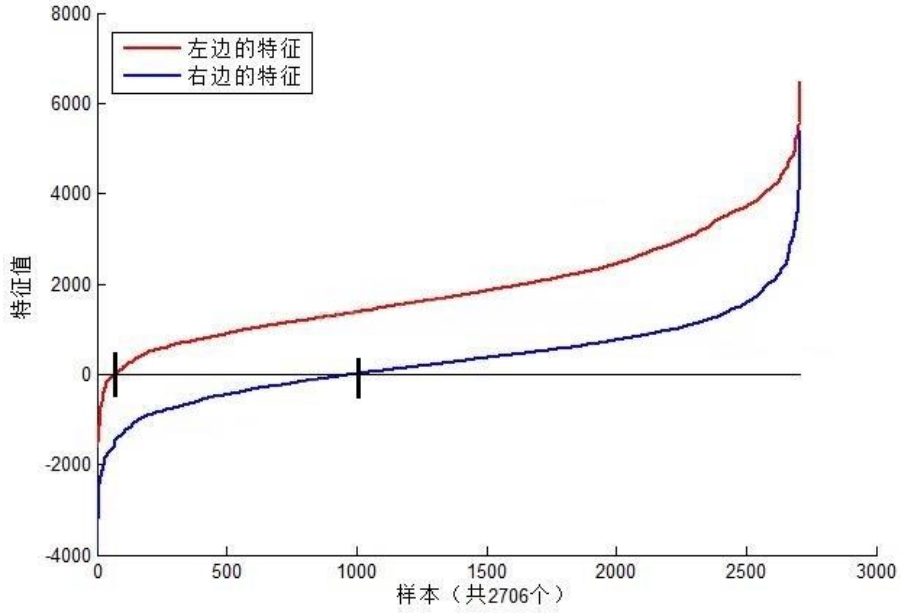


图 2.4 Harr 中 x2 和 y2 在 MIT 人脸样本中特征值分布

由上图可以看到，x2 和 y2 不同 Haar 特征在同一组样本中具有不同的特征值分布，左边特征计算出的特征值基本都大于 0，而右边特征的特征值基本均匀分布于 0 两侧（分布越均匀对样本的区分度越小）。所以，正是由于样本中 Haar 特征值分布不同，导致了不同 Haar 特征分类效果不同。显而易见，对样本区分度越大的特征分类效果越好，即红色曲线对应图 6 中的的左边 Haar 特征分类效果好于右边 Haar 特征。

由此可以得出结论：①在检测窗口通过平移+放大可以产生一系列 Haar 特征，这些特

征由于位置和大小不同，分类效果也各异；②通过计算 Haar 特征的特征值，可以有将图像矩阵映射为 1 维特征值，有效实现了降维。

由上图可得知 Harr 特征值计算出的特征值变化范围跨度非常大，这种跨度大的特性不利于量化评定特征值，所以需要进行“标准化”，压缩特征值范围。假设当前检测窗口中的图像为 i(x,y)，当前检测窗口为 w*h 大小（例如图 6 中为 20*20 大小），OpenCV 采用如下方式“标准化”：

①计算检测窗口中间部分(w-2)*(h-2)的图像的灰度值和灰度值平方和：

$$sum = \sum i(x,y)\ \ sqsum = \sum i^2(x,y) \tag{2.4}$$

②计算平均值：

$$mean = \frac{sum}{w \times h}\ \ sqsum = \frac{sqsum}{w \times h} \tag{2.5}$$

③计算标准化因子：

$$varNormFactor = \sqrt{sqmean - mean^2} \tag{2.6}$$

④标准化特征值：

$$normValue = \frac{featureValue}{varNormFactor} \tag{2.7}$$

经过上述公式变换，将特征值进行归一化后，为后续设定的阈值进行对比，当图像中的特征值与样本测试选择的特征值进行对比，判断图像中的人脸信息。

2.2.2 基于 OpenCV 的 LBP 级联分类器

局部二值模式（LBP）算法是由 Ojala 等人于 1996 年提出的，LBP 算法基本思想是对待测图像进行单元划分，得到很多个小的图像窗口，先计算出待测图像中固定窗口的特征值，然后统计整张图中每个窗口的特征值，经过统计运算得到整张图的特征信息。LBP 算有由于其思想简单，效果出众，对光照不敏感，计算量小和计算速度快等优点，已经广泛应用于图像特征提取当中了。

在特征提取的运算速度以及纹理特征提取方面 LBP 算法都有不俗的表现，其具体的图像特征提取过程如下：将经过预处理的图像进行扫描，然后定义一个 3×3 的像素区域作为计算单元，以像素区域内中心像素点的灰度值作为标准，按一定的顺序将 8 个边缘像素点的灰度值与其进行比较，若小于中心像素点灰度值则设置为 0，否则为 1，再按一定的顺序将设置的值组成一个 8 位的二进制数，该二进制数所对应的十进制数就是中心像素点的 LBP 特征值。LBP 特征描述方法的扫描及编码过程如图 2.5 所示。

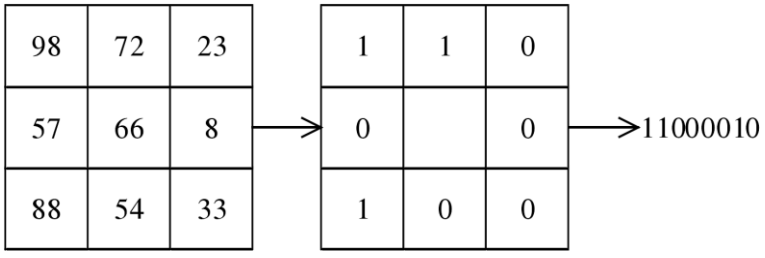


图 2.5 LBP 特征编码过程示例图

根据上述过程，将 LBP 编码过程用式 2.8 和式 2.9 表示:

$$LBP = \sum_{i=1}^{7} f(p_i - p_c) \cdot 2^i \tag{2.8}$$

$$f(x)=\begin{cases}1, x \geq 0\\ 0, x < 0\end{cases} \tag{2.9}$$

其中，p_c是所选区域里中间点的像素值，p_i是中间点领域的第i点的像素值。

LBP 特征描述方法通过比较各点像素值的大小关系来计算出LBP特征值，当某区域所有像素点增加或减小相同大小的像素值时，该区域的LBP特征值会保持不变。因此，该特征描述方法对线性光照强度具有良好的鲁棒性，如图 2.6 所示。

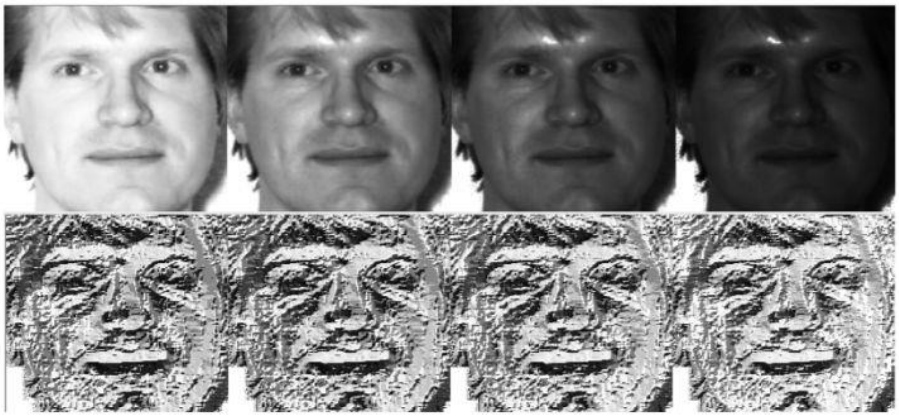


图 2.6 LBP 对不同线性光照的鲁棒性示意图

2.3 基于 RetinaFace 的模型训练与结果

2.3.1 实验平台

实验在 Windows10 操作系统上完成的。在 CPU 为 Intel(R) Xeon(R) Gold 5218 CPU、内存为 63G、GPU 为 RTX2080Ti 的环境下训练，再将训练好的模型在 CPU 为 i5 7300H、内存为 16G、GPU 为 GTX1050Ti 的环境下进行测试。实验使用的框架为 Pytorch，同时采用了很多的第三方库保证代码的正常运行。

2.3.2 实验数据集

Retinaface 人脸检测网络使用 Widerface 数据集进行训练。该数据集是人脸检测的 benchmark 数据集，包含 32203 张图像，以及 393703 个标注人脸。在这些数据据中有 158,989 个标注人脸位于训练集，39,496 个位于验证集。每一个子集都包含 3 个级别的检测难度：Easy，Medium，Hard。这些人脸在尺度，姿态，光照、表情、遮挡方面都有很大的变化范围。

2.3.3 实验结果解码及分析

Retinaface 网络主干特征提取网路在选择上有两种：①基于残差网络的 Resnet50 网络；②基于 Google 针对手机等嵌入式设备提出的一种轻量级的深层神经网络 MobileNet。从网络效果上来说，Resnet50 可以实现更高的精度，使用 MobilenetV1-0.25 可以在 CPU 上实现实时检测。训练过程中网络设置的输入图像大小为 1280×1280×3 的图像大小，具体结果如下图所示：

表 2.2 网络性能情况

训练数据集	主干特征提取网络	测试数据集	输入图片大小	Easy	Medium	Hard
Widerface-Train	mobilenet0.25	Widerface-Val	1280x1280	89.76%	86.96%	74.69%
Widerface-Train	resnet50	Widerface-Val	1280x1280	94.72%	93.13%	84.48%

Retinaface 为了进一步加强特征提取，使用了 SSH 模块加强感受野。其利用 3x3 卷积的堆叠代替 5x5 与 7x7 卷积的效果：左边的是 3x3 卷积，中间利用两次 3x3 卷积代替 5x5 卷积，右边利用三次 3x3 卷积代替 7x7 卷积。通过 SHH 获得三个有效网络特征层进行结果解码。

Retinaface 的预测结果分为三个，分别是分类预测结果，框的回归预测结果和人脸关键点的回归预测结果。模型在设计上使用输入图片的大小为 1280×1280，所以每种类型的输出分别为 67200×2、67200×4、67200×10 的数据。

分类预测结果用于判断先验框内部是否包含物体，原版的 Retinaface 使用的是 softmax 进行判断。此时我们可以利用一个 1x1 的卷积，将 SSH 的通道数调整成 num_anchors x 2，用于代表每个先验框内部包含人脸的概率。

框的回归预测结果用于对先验框进行调整获得预测框，我们需要用四个参数对先验框进行调整。此时我们可以利用一个 1x1 的卷积，将 SSH 的通道数调整成 num_anchors x 4，用于代表每个先验框的调整参数。

人脸关键点的回归预测结果用于对先验框进行调整获得人脸关键点，每一个人脸关键点需要两个调整参数，一共有五个人脸关键点。此时我们可以利用一个 1x1 的卷积，将 SSH 的通道数调整成 num_anchors x 10（num_anchors x 5 x 2），用于代表每个先验框的每个人脸关键点的调整。

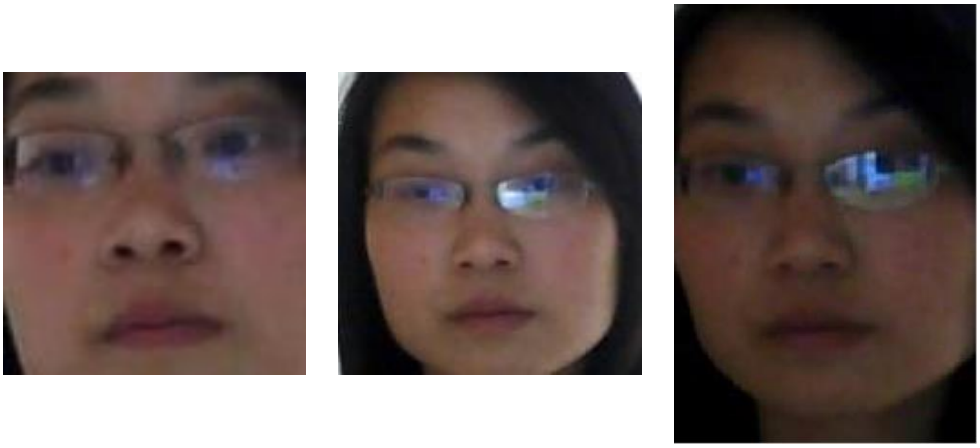
2.4 实验对比

本次人脸检测实验主要从 NUAA 数据集挑选对象名为 0004 的文件夹，其一共包含 681 张图片，拍摄均为正脸图像，主要包含 3 个不同的场景和着装，特征为戴眼镜，示例如图 2.7 所示。人脸检测实验主要对 OpenCV 中 LBP 特征和 Harr 特征的级联分类器以及 retinaface 人脸检测的方法进行速度和检测准确度的比较。

图 2.7 人脸检测测试集

对于不同的人脸检测方法，其得到的检测结果如下图 2.8，LBP 特征得到的人脸图像较小，包含的人脸信息较少，相较于其他两种不利于活体检测和人脸识别的检测；而 retinaface 的检测准确率虽然较高，包含的人脸信息也较为丰富，但其在 SylixOS 中运行时平均用时为 2.50s，远高于其他两种，使用这种方法将导致相机画面在画框时卡顿。因此，本次人脸检

测将选取 OpenCV 中 Harr 特征的级联分类器。



（a）LBP 特征检测结果　（b）Harr 特征检测结果　（c）retinaface 检测结果

图 2.8 不同方法检测结果图

表 2.3 不同方法检测结果对比

	用时/s	检测准确率
LBP 特征检测	0.14	99.41%
Harr 特征检测	0.15	99.41%
retinaface 检测	2.50	100%

第 3 章 基于 MiniFASNet 的静默活体检测

随着人脸识别系统在不同场景中的逐渐普及，人们对人脸识别技术的使用越来越熟悉，人脸识别技术的安全弱点也就逐渐被大众所知，这时就有许多不法分子开始利用人脸识别系统的弱点对其进行攻击，对个人和社会都会造成隐私和财产等诸多方面的损失，因此确保人脸识别系统在实际应用过程中的安全性是非常重要的。

其中人脸活体攻击是最常见的攻击手段，不法分子利用受害者的人脸照片或其他信息去人脸识别系统进行认证，盗取受害者的隐私或者财产等，对用户造成财产或其他方面的损失。尤其是随着互联网社交的日益普及和手机、摄像头等低成本拍照设备的出现，几乎所有人都可以很轻易地得到其他人的人脸照片，不需要任何创新技术就可以对人脸识别系统进行攻击，其中有一些攻击方法肉眼也很难区分活体人脸和非活体人脸，这些攻击手段使得人脸识别技术的安全性无法得到保证，对用户的隐私、财产等造成了巨大的威胁，因此验证是否为活体人脸对于人脸识别在现实中的应用十分重要，这时就有众多研究人员开始研究人脸活体检测技术来判断输入系统中的人脸图像是活体人脸图像还是非活体人脸图像，比如打印好的人脸照片、电子设备拍摄的人脸照片或视频、3D 打印的人脸面具等，该技术得到了研究人员的高度重视，并在过去的十几年内产生了大量的研究成果。

为了抵御不法分子的攻击，目前的人脸识别系统中都会包含人脸活体检测这一步骤，利用人脸活体检测进行真假人脸的检测，提高人脸识别系统的安全性，然后利用人脸识别进行身份的认证。人脸活体检测技术的加入极大地提高了人脸识别系统的安全性，对人脸识别系统在实际场景的应用产生了非常重要的作用，对用户的隐私和财产安全提供了更好的保障。

活体检测技术主要是判别机器前出现的人脸是真实还是伪造的，其中借助其他媒介呈现的人脸都可以定义为虚假的人脸，包括打印的纸质照片、电子产品的显示屏幕、硅胶面具、立体的 3D 人像等。目前主流的活体解决方案分为配合式活体检测和非配合式活体检测（静默活体检测）。配合式活体检测需要用户根据提示完成指定的动作，然后再进行活体校验，静默活体则在用户无感的情况下直接进行活体校验。

基于 RGB 图像的静默活体检测网络 MiniFASNet，是专门为工业落地场景设计的网络，兼容各种复杂场景。其采用轻量级网络作为骨干训练模型，使用 Softmax 作为训练分类的监督者。利用不同尺度的图像作为网络的输入训练数据，增加模型之间的互补性，从而融合模型。考虑到用于辨别真假人脸的有效信息可能不完全分布在人脸区域，可能在取景画面的任何地方，在骨干网络上增加了 SE（Squeeze-and-Excitation）注意力模块，动态适应分散的判别线索。同时引入傅里叶频谱图作为模型训练的辅助监督，有效提升了模型精度。

3.1 活体检测的网络结构

活体检测的网络结构如图 3.1 所示，采用轻量级网络作为骨干网络，使用 Softmax+CrossEntropy 作为网络的监督。使用不同尺度的图片作为网络的输入训练数据，增加模型间的互补性，从而进行模型融合。考虑到用于真假脸判别的有效信息不一定完全分布在脸部区域，可能在取景画面的任何地方，如边框，摩尔纹等，在网络中加入了 SE（Squeeze-and-Excitation）注意力模块，动态适应分散的判别线索。同时引入傅里叶频谱图作为模型训练的辅助监督以提升模型精度。

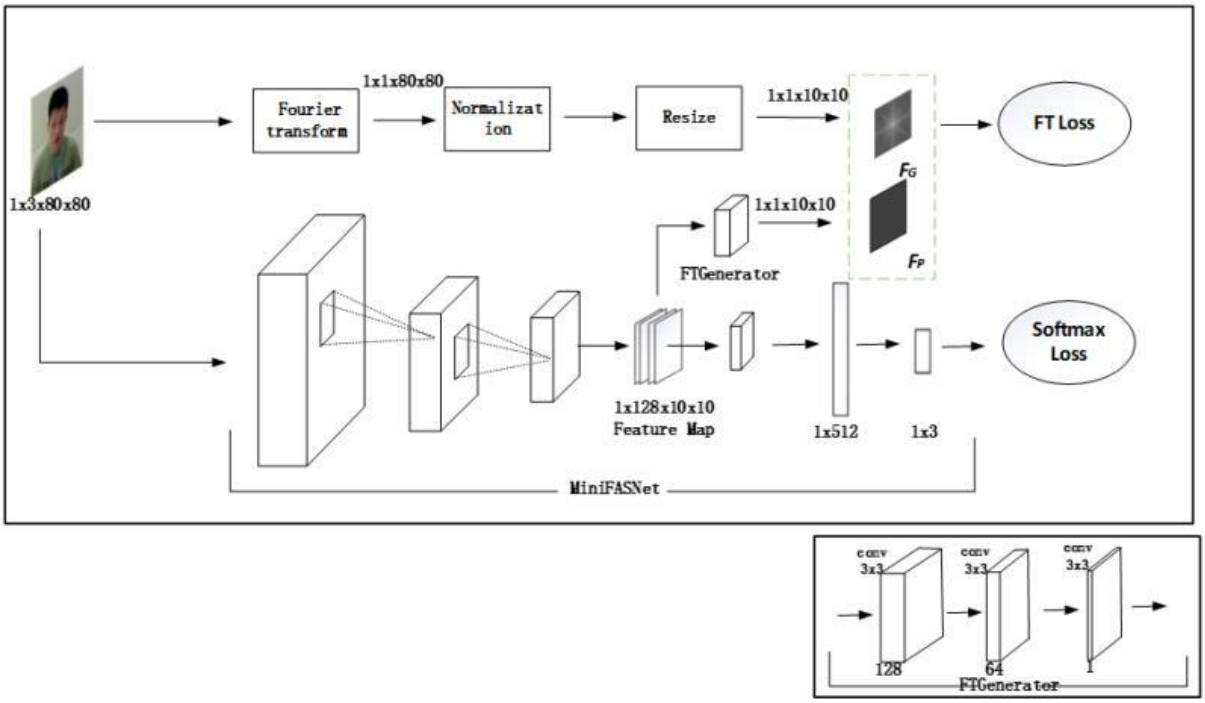


图 3.1 活体检测的网络结构图

3.1.1 数据预处理

根据成像介质的不同类型，将训练集分为真实人脸、2D 人脸成像和 3D 人脸模具三类，将相同类别的图片放入一个文件夹。因采用多尺度模型融合的方法，分别用原图和不同的 patch 训练模型，所以将数据分为原图和基于原图的 patch。

首先直接将原图 resize 到固定尺寸(width, height)，如图 3.2 左 1 所示；然后使用人脸检测器检测人脸，获取人脸框，再按照一定比例(scale)对人脸框进行扩边，为了保证模型的输入尺寸的一致性，将人脸框区域 resize 到固定尺寸(width, height)，图左 2、3、4 分别显示了 scale 为 1，2.7 和 4 的 patch 样例。

图 3.2 Patch 示例图

3.1.2 注意力机制

近年来，越来越多的研究者将注意力机制引入计算机视觉的研究，并取得了很好的效果。计算机视觉中的注意力机制本质上是在模仿人类观察物品的方式。人类在观察物品时，

不会直接对眼睛捕捉到的整个场景进行处理，而是快速扫描全局，有选择地找到需要关注的目标区域，提取目标区域的信息忽略掉其他无关区域的信息。人类的视觉注意力机制大大提高了视觉信息处理的效率。

而计算机视觉中的注意力机制与之类似，也是通过使模型重点关注目标区域提升模型的速度和准确率。早期的注意力机制是基于人类大脑的成像机制设计的，随着深度学习的发展，越来越多研究者注意到深度学习与注意力机制之间的联系，一方面深度学习强大的学习能力能够自主学习注意力机制，另外一方面深度学习目前还是”黑盒”，注意力机制有助于我们了解深度学习学习到了什么以及是如何进行学习的。

近年来的视觉注意力机制应用在深度学习中的形式主要是利用掩码引入新的权重，将数据中的关键特征标记出来，引导神经网络学习到需要重点关注的区域。这一思想目前演化成两种注意力机制：软注意力机制(softattention)和硬注意力机制(hardattention)。这两种注意力机制在定义和实现上差距比较大，其中软注意力机制是确定性的注意力，是可微的结构，可以随着整个网络一起进行前向和反向的传播，通过梯度下降学习到注意力权重。而硬注意力权重是随机性的注意力，它是不可微的，一般通过强化学习进行训练。

考虑到用于真假脸判别的有效信息不一定完全分布在脸部区域，也有可能在取景画面的任何地方，如边框，摩尔纹等，为了进一步提升网络的性能，本文继续在原有的 MiniFASNet 骨干网络结构中引入通道注意力机制。动态适应分散的判别线索，热力图如图 3.3 所示。对于假脸，模型更关注于边框信息和屏幕的摩尔纹信息，对于真脸更加关注脸部以及周围的信息。

图 3.3 活体模型热力图

SE 模块可以通过训练学习到特征通道的注意力遮罩(AttentionMask)，这个注意力遮罩能够重新定义不同特征通道之间的重要性，再与原先的特征进行结合从而增强对当前任务有用的特征通道，抑制对当前任务无效的特征通道，将其添加在特定任务分支的最后，能够有效地增强网络学习到的特征的特异性。整个 SE 模块可以分为两个部分：压缩(Squeeze)和激励(Exicitation)，SE 模块的结构图如图 3.4 所示：

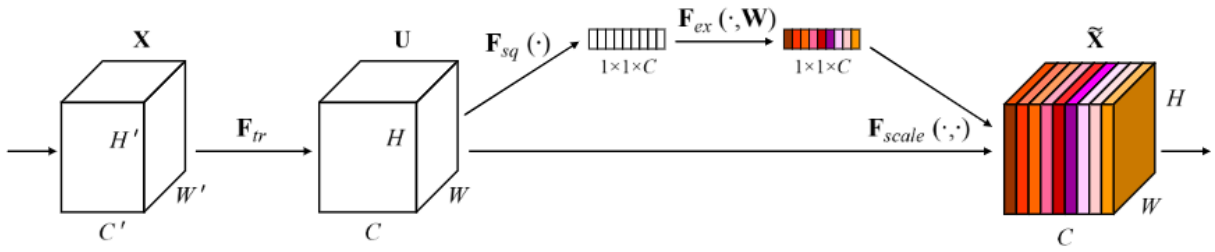


图 3.4 SE 模块的工作示意图

在 SE 模块中，首先 Squeeze 操作对输入的特征图U的每个通道的特征图进行全局池化操作，计算公式如下：

$$z_c = \mathrm{F}_{sq}(U) = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} U(i,j) \tag{3.1}$$

其中，C 指的是通道数，经过 Squeeze 操作得到了 1*1*C 的数组，其具有全局的感受野能够利用全局信息。随后 Exicitation 操作通过通道间的依赖关系为每个特征通道生成相应的权重，计算公式如下：

$$F_{ex}(z, W) = \sigma\big(W_2\delta(W_1 z)\big) \tag{3.2}$$

其中，$\sigma(*)$指的是 sigmoid 激活函数，$\delta(*)$指的是 ReLU 函数，在网络中具体实现是使用了两个先降维再升维的全连接层，这样就得到了 C 个特征通道的权重，再将权重应用到输入特征图对应的通道上。

3.2 损失函数

将假脸照片与真脸照片转化生成频域图，对比发现假脸的高频信息分布比较单一，仅沿着水平和垂直方向延伸，而真脸的高频信息从图像的中心向外呈发散状，如图 3.5 所示。根据以上的实验，发现真脸和假脸的傅里叶频谱存在差异，从而引入了傅里叶频谱对模型训练进行辅助监督。

图 3.5 傅里叶频谱图

基于对于真脸和假脸在频域的观察分析，提出基于傅里叶频谱图进行辅助网络监督的训练方法。网络的主分支采用 Softmax+CrossEntropy 作为网络的监督，如公式 3.3 所示。其中，f_j表示输出类别的第 j 个置信度，y_i表示样本的真实标签，N 为训练样本的个数。

$$L_{Softmax} = -\frac{1}{N}\sum_{i}\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \tag{3.3}$$

采用在线的方式生成傅里叶频谱图，使用 LT Loss 作为损失函数。输入图片的尺寸为

3×80×80，从主干网络中提取尺寸为 128×10×10 特征图，经过 FTGenerator 分支生成 1×10×10 的预测频谱图 F_P。通过傅里叶变换，将输入图片转化成频谱图，再进行归一化，最后 resize 成 1×10×10 尺寸得到 F_G，使用 LT Loss 计算两特征图之间差异，如公式 3.4 所示。

$$L_{FT} = \|F_P - F_G\|_2^2 \tag{3.4}$$

3.3 实验与结果

本次活体检测实验主要选取 NUAA 的活体检测数据集，主要为打印照片攻击，15 个人，三次拍摄，每次地点照明各不相同。该数据集一共包含 12614 张图像，其中有 5105 个真图像和 7509 个攻击图像。如图 3.6，本次实验选取不同性别的有无眼镜共 4 个对象进行测试，其中各对象真实图像与攻击图像分布如下表 3.1。

图 3.6 活体检测测试集示意图

表 3.1 测试数据集分布表

序号	真实图像	攻击图像
0001	250	614
0004	681	608
0006	730	458
0009	208	602

使用 ncnn 框架下的 minivision 的活体检测模型对测试集进行测试，结果按照置信度为 0.6 处理，即图片的活体检测置信度超过 0.6 作为真实图像，低于 0.6 作为攻击图像。如下表 3.2，对于二元分类结果，使用混淆矩阵进行展示，其中的正例为真实图像，反例为攻击图像，其中 TP 为被模型预测为真实图像的真实图像，TN 为被预测为攻击图像的攻击图像，FP 为被模型预测为真实图像的攻击图像，FN 为被模型预测为攻击图像的真实图像，其总体识别率为（TP+TN）/ALL=99.37%，错误率为（FP+FN）/ALL=0.63%，可以较好地完成活体检测任务。

表 3.2 混淆矩阵

		预测情况	
		正例	反例
真实情况	正例	1850	19
	反例	7	2275

第 4 章 基于 MobileFaceNet 的人脸识别

目前人脸识别算法是应用较为广泛的算法。尤其是对于标准的质量较高的人脸图片的识别，随着手机面部识别技术、人脸门禁技术等技术的迅猛发展，较好质量的人脸的识别技术已经比较成熟了。但随着应用场景的不断扩大，对人脸识别的需求越来越多样化，人脸识别算法还存在一些技术难点。

从前人大量的实验数据来看，虽然配合情况下采集到的人脸的识别情况普遍较好，但非配合条件下人脸的姿态、表情、面部信息保留情况等是不可控的，这样就会导致很难找到一张标准理想的人脸用于识别，这就要求识别算法对可能出现的人脸角度偏转、人脸遮挡等情况下的识别精度要具备可靠性。除此之外，识别精度与运算时耗也是需要平衡的点，这是算法能实际应用的基础。

人脸识别的总体流程是先对一张图片进行人脸检测和人脸关键点定位，然后将检测到的人脸与背景图片分离，利用关键点对人脸图片进行矫正和对齐，再提取人脸的特征，利用人脸识别算法对特征进行分类，将分类结果与人脸库中的数据进行比对判断是否为同一个人。

Mobilefacenet 是一种轻量化的人脸识别卷积神经网络模型，于 2018 年提出，其网络结构是基于 Mobilenet 改进的，将网络 Mobilenet 的尾部池化操作用深度卷积操作替代，网络关键点在于深度可分离卷积的运用和使用了逆残差模块。MobileFaceNet 是一个非常高效的 CNN 模型，是专门针对移动设备设计的轻量级神经网络。模型的参数量不到 1 百万，模型体积较小，但是准确率很高，非常适用于移动设备。

为了提高人脸识别模型的准确性和稳定性，伦敦帝国理工学院邓建康等人于 2018 年提出了 ArcFace 人脸识别方法。该方法中提出了一种加性角度间隔损失（Additive Angular Margin Loss，ArcFace）函数。该方法的原理是：使用深层卷积神经网络（Deep convolutional neural network，DCNN）来提取人脸特征，提取到的特征和权重归一化后的点积等于两者之间的余弦距离，使用弧余弦函数来计算当前特征和目标权重之间的角度，然后在目标角度上加上一个附加的角边距，再通过余弦函数得到目标 logit，之后用一个固定的特征范数重新缩放所有计算得到的 logits，随后的步骤与 Softmax 损失函数中的计算步骤完全相同。ArcFace 人脸识别方法的优点：有效性，ArcFace 在十个人脸识别基准（包括大规模图像和视频数据集）上实现了最先进的性能；简便性，ArcFace 方法易于在各种深度学习框架中变成实现，例如 MXNet、Pythorch 和 TensorFlow；高效率，训练过程中，ArcFace 只增加了微不足道的计算复杂度。当前的 GPU 可以轻易地支持数百万的训练实体，模型并行策略可以很容易地支持更多的实体。

4.1 人脸识别的网络结构

本系统使用 MobileFaceNet 作为人脸识别特征提取的基础网络，人脸识别卷积神经网络 MobileFaceNet 的网络结构如图 4.1 所示，网络的输入三通道的图片尺寸为 112×96 的人脸图片，网络输出该图片的特征是大小为 1×512 的向量。表 4.1 说明了 MobileFaceNet 每层的参数设置。

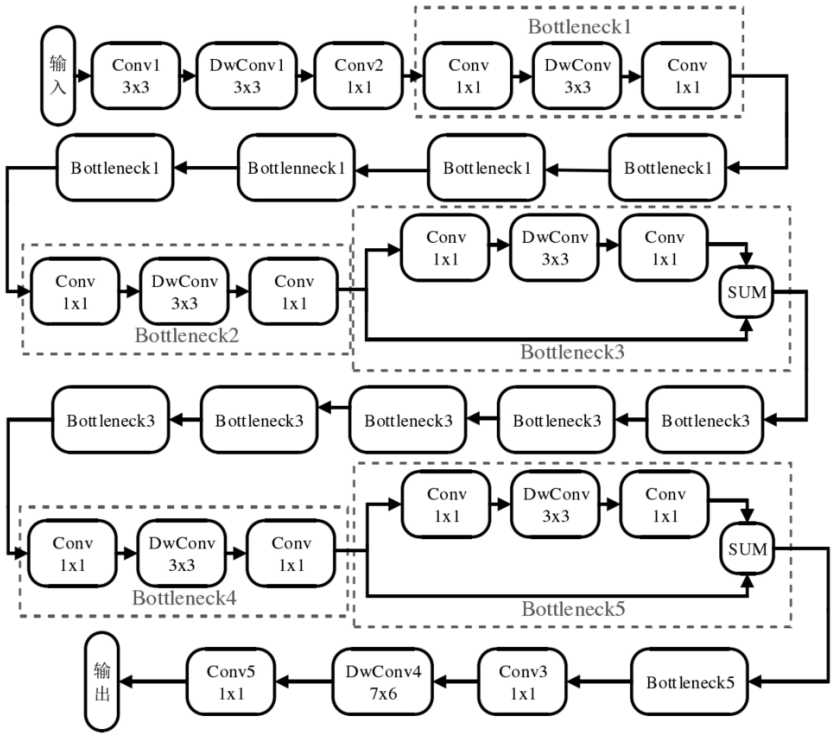


图 4.1 MobileFaceNet 网络结构

表 4.1 MobileFaceNet 参数设置表

输入大小	层名称	包含	步长(stride)	卷积核数目
112×96×3	Conv1		2	64
56×48×64	DwConv1		1	64
56×48×64	Conv2		1	64
56×48×64	Bottleneck1	Conv	1	128
56×48×128		DwConv	2	128
28×24×128		Conv	1	64
28×24×64	Bottleneck2	Conv	1	512
28×24×512		DwConv	2	512
14×12×512		Conv	1	128
14×12×128	Bottleneck3	Conv	1	256
14×12×256		DwConv	1	256
14×12×256		Conv	1	128
14×12×128	Bottleneck4	Conv	1	512
14×12×512		DwConv	2	512
7×6×512		Conv	1	128
7×6×128	Bottleneck5	Conv	1	256
7×6×256		DwConv	1	256
7×6×256		Conv	1	128
7×6×128	Conv3		1	512
7×6×512	DwConv4		1	512
1×1×512	Conv5		1	512

4.1.1 深度可分离卷积

为了提高卷积神经网络卷积操作的速率，前人在轻量级的网络结构例如 Mobilenet、Shufflenet 中提出采用深度可分离卷积（depthwise separable convolution）来替代普通的卷积计算 feature map。深度可分离卷积和普通卷积操作相比在损失较少精度的情况下，保留相对来说更少的参数，以此加快卷积运算的速度。

普通的卷积运算是对输入图片的各个通道都在同一时间进行卷积运算操作。而对于深度可分离卷积并非如此，深度可分离卷积在进行运算时会进行下面两个部分操作:逐通道卷积和逐点卷积。假设输入的尺寸是$D_F \times D_F \times M$，计算输出 Feature Map 尺寸是$D_F \times D_F \times N$且卷积核尺寸是$D_K \times D_K$，普通卷积的运算量为$D_K \times D_K \times M \times N \times D_F \times D_F$，深度可分离卷积的运算量为$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$。以下是一个具体的例子显示了两种卷积操作的差距。

假设 3 通道数的 5×5 大小的输入，假设进行卷积核为 3×3 且通道数为 4 的卷积操作，那么卷积核的大小是 3×3×3×4，最后输出的 Feature Map 的数量是 4 个，并假设 padding 值相同，即输出尺寸与输入的尺寸一样都是 5×5，此时普通卷积层的 Filter 数为 4，每一个 Filter 都包含了 3 个大小为 3×3 的卷积核 Kernel。所以此时普通卷积操作参数量为 4×3×3×3=108。上述操作示意图见图 4.2。

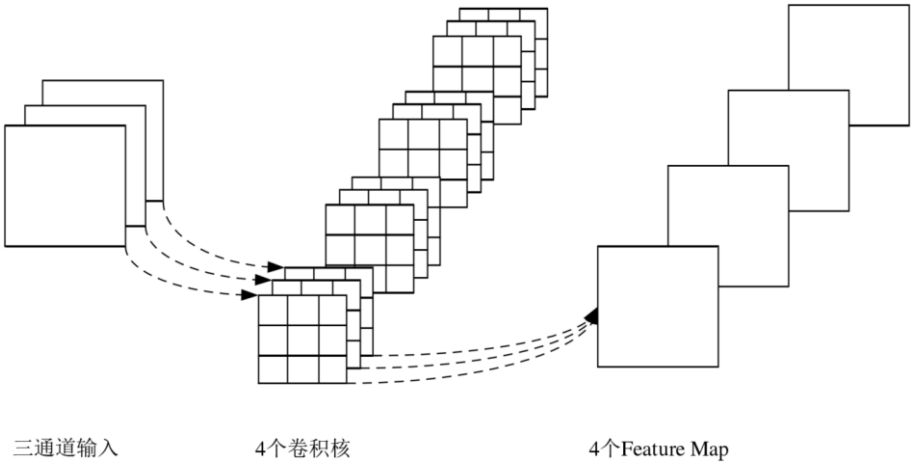


图 4.2 普通卷积操作示意图

深度可分离卷积在上述同等情况下首先进行逐通道卷积，这步操作时每个卷积核仅与对应的单一通道进行操作，这样的操作可以看作只是一个二维平面内的运算，逐通道卷积进行第一次运算之后，输入的 3 通道的图片得到了 3 个 Feature Map，每个内核尺寸为 3×3，此时的参数量是 3×3×3=27。深度可分离卷积的逐通道卷积操作示意图见图 4.3。

然后进行逐点卷积，经过第一步逐通道卷积后得到的 Feature Map 不能被扩展，并且对各个通道在同一空间位置上的信息并没有完整的利用上，所以需要逐点卷积将上一步得到的 Feature Map 做组合操作以生成利用了位置信息的 Feature Map。逐点卷积操作如图 4.4 所示，根据上一层的通道数 m 设置卷积核大小为 1×1×m，将逐通道卷积计算出的 Feature Map 在深度上进行融合扩展，得到新的利用完所有信息的 Feature Map。这次操作的参数量为 1×1×3×4=12。

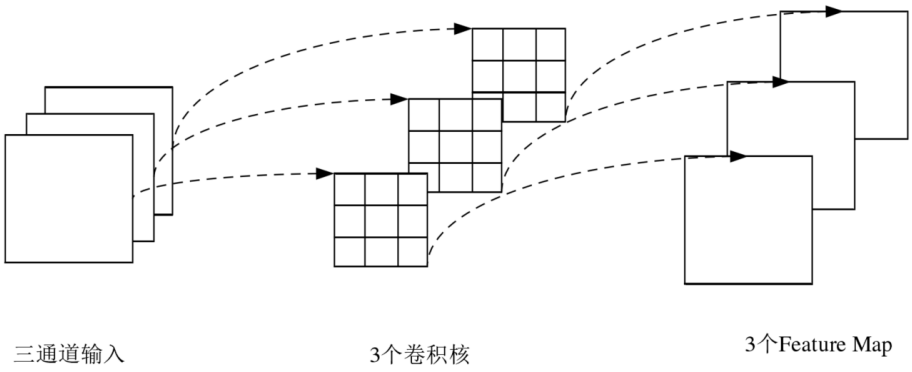


图 4.3 逐通道卷积操作示意图

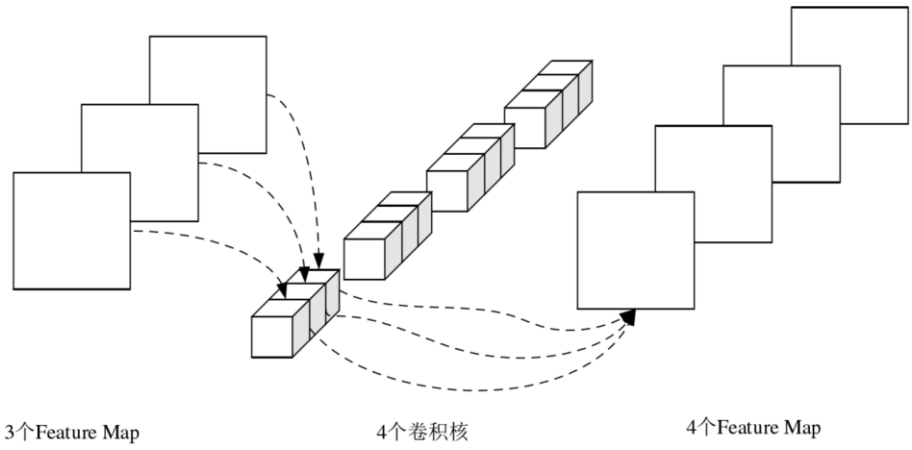


图 4.4 逐点卷积操作示意图

至此完成了一次深度可分离卷积操作，该操作整体参数量是 39。可以看出深度可分离卷积比普通的卷积操作完成相同计算时在参数量上精简了大约 66.66%。

4.1.2 逆残差模块

逆残差模块旨在解决深度网络模型退化问题的 ResNet 网络，推出了残差模块，凭借这一设计还拔得了 2015 年 ImageNet 大赛的头筹，基础的残差模块如图 4.5 所示。输入 *X* 经过垂直向下的计算之后得到有一定程度表示能力的 *F(X)*。为了避免卷积神经网络出现退化的情况，额外将输入 *X* 与及 *F(X)*相加，希望获得的计算后的表达能力至少比上一层好，这就是图中直接将 *X* 也一并输出的原因。这样的模块会让网络参考浅层的信息，从而加大整个网络的表征能力。这样的模型结构已经被广泛运用。

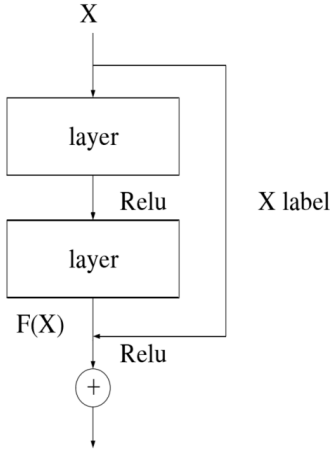


图 4.5 残差模块示意图

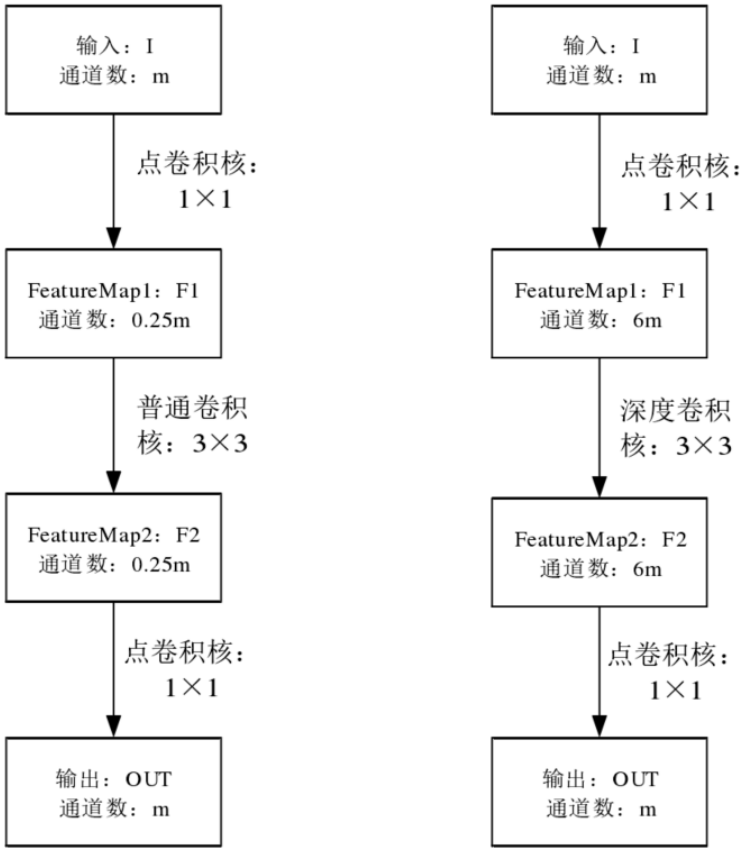


图 4.6 残差和逆残差模块对比示意图

在残差模块提出后，在此基础上 MobilenetV2 又做出了改进，将残差网络改进成逆残差网络，这主要是为了解决从深度可分离卷积获得的特征数目受到输入的通道数的限制。残差模块为了提高计算效率，用卷积核大小为 1×1 的卷积操作压缩 Feature Map 的通道数，但逆残差模块在计算时会先扩展 Feature Map 之后再压缩。残差模块和逆残差模块的垂直向下计算过程对比如图 4-6 所示，逆残差直路在计算过程中卷积核大小依次为:1×1→3×3→1×1，并且卷积时采用的是深度可分离卷积，不同于普通卷积先使用卷积核为 1×1 的卷积压缩 Feature Map 得到较少的通道数后再进行抽取特征，最后扩张 Feature Map。MobilenetV2 的深度可分离卷积则是卷积核大小为 1×1 的卷积操作将 Feature Map 增加通道数之后再用深度卷积压缩得到的 Feature Map。

4.2 ArcFace 损失函数

进行大规模人脸识别时，普遍是使用深度卷积神经网络学习嵌入人脸特征。最关键挑战包括提出一个合理的 Loss 函数来提高模型的识别性能。ArcFace 提出了一个加性角度间隔的 Loss 函数，以获得人脸识别的高判别性的人脸特征。

现在许多被广泛使用的分类损失函数都是在 Softmax Loss 的基础上进行变形，得到了很好的效果。原始的 Softmax Loss 如下所示：

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{W_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \mathbf{x}_i + b_j}} \tag{4.1}$$

式中，$x_i \in R^d$定义为第 i 个输入的深层次的特征，属于y_i类别。人脸嵌入的特征向量维数 d 设为 512。$W_j \in R^d$表示权重$W \in R^{d \times n}$的第 j 列，$b_j \in R^n$为偏移项，batchsize 与标签身份的数量定义为 N 和 n。虽然经常使用经典的 Softmax Loss 函数用作人脸识别模型得监督训练，但是 Softmax Loss 从未明确地优化人脸特征向量，来增进类内人脸特征之间的相似程度和类间的人脸特征之间的差异性，是深度人脸识别的性能存在差异的原因。

为了简单起见，将偏置项固定为$b_j = 0$。将 logit 变换为$W_j^T x_i = \|W_j\|\|x_i\|\cos\theta_j$，其中$\theta_j$，是权重$W_j$，和特征$x_i$之间的角度。通过$L_2$归一化固定单个权重$\|W_j\| = 1$。同样的$L_2$归一化操作，将人脸嵌入的特征向量$\|x_i\|$固定起来，再进行重新缩放成 s。由于对权重和特征向量的归一化操作使预测结果仅依赖权重与嵌入的特征向量之间的角度。因此，使通过网络学习得到的人脸嵌入的特征向量分布在半径为 s 的超球体上。

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\cos\theta_{y_i}}}{e^{s\cos\theta_{y_i}} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{4.2}$$

ArcFace 在x_i，和W_{y_i}，之间增加了一个加法角度间隔惩罚m，从而拉近类内距离，同时增大类间差异。

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{4.3}$$

ArcFace 工作原理如图 4.7 所示。在 ArcFace Loss 函数的监督下，训练一个分辨人脸特征的深度网络。卷积神经网络得到的嵌入向量和全连接层的点积$W_j^T x_i$表示为归一化之后的嵌入向量x_i和权重W的余弦距离$\cos\theta_j$，通过计算$arccos\,\theta_{y_i}$，得到嵌入向量x_i，和真实权重W_{y_i}，之间的夹角。事实上，W_j为每个类别提供了一个中心。做法为在 groundtruth 角度θ_{y_i}，上加上一个角度间隔惩罚m．计算$\cos(\theta_{y_i} + m)$，获得目标 logit。接着用一个固定的特征缩放因子s重新缩放所有的 logit。然后将这些 logit 通过 Softmax 损失函数，将得出的概率投放到交叉熵损失中计算。

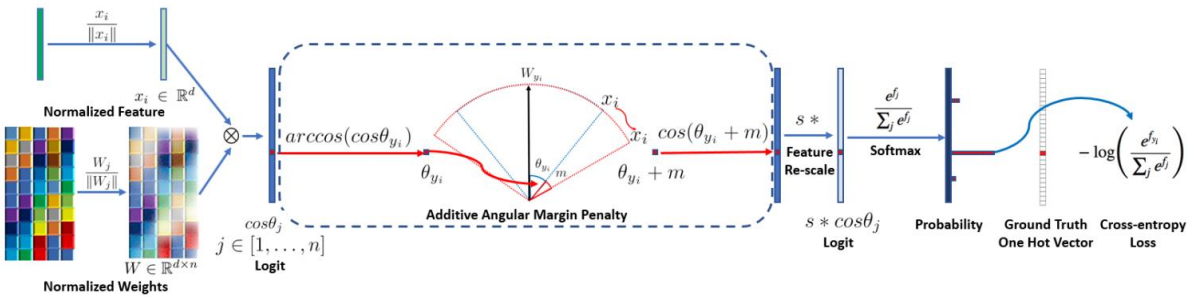


图 4.7 ArcFace 模型的概述

直接在特征x_i和真实权重W_{y_i}，之间的夹角θ_{y_i}。上加角度间隔m，而之前的基于 Softmax 的损失函数，如 CosinFace 中的损失函数是在余弦距离上做的改进。角度间隔m以加法形式惩罚网络得到的特征嵌入与之对应的权重之间的角度，从而达到同类更加紧凑，不同类更加分离的效果。惩罚的意味着:训练时 ArcFace 损失函数降低到某一个固定值时，有间隔m和无间隔m的指数项值是相同的，那么有间隔m的夹角相对就减小了。就是有间隔m的训练会把特征x_i，和真实权重W_{y_i}，之间的夹角θ_{y_i}缩小了，则间隔m把θ_{y_i}，挤得更类内紧凑，相比之下θ_{y_i}，和其他类别就更加分散了。

4.3 实验与结果

4.3.1 数据集

采用 CASIA-WebFace 作为训练数据，训练之后的模型，跑在 LFW 数据集上来测量模型的效果。CASIA-WebFace 数据库有 10575 个类别 494414 张图像，每个类别中有很多文件夹，一个文件夹包括几张或者几十张人脸图片。每个文件夹代表一个人，文件夹保存这个人脸的所有图片。LFW 数据集对 5000 多人在自然场景下采集，一共 13000 多张图像。是从互联网上收集的自然场景中不同姿势、不同光照的人脸图片，共有 5749 个人的 13233 张图片，其中 1680 个人拥有两张或两张以上的人脸图片，LFW 数据集如下图 4.8 所示：

图 4.8 LFW 数据集

4.3.2 数据集预处理

数据集预处理主要是为了使图片更够更好的适应网络模型或者之后的一系列操作而作的简单处理。传统图像主要是使用灰度图像处理或者是高斯滤波等一系列操作，但是对于卷积神经网络，可以对图像直接进行处理，也可以根据自己的网络模型或者实验的具体情况对图像进行预处理。本文选用数据集均为在自然场景下收集的图片，对人脸数据集进行检测，对检测到的人脸进行防射操作达到人脸对齐的效果。同时为了减少图像噪声对训练的影响，对图像进行均值处理，对每个像素点的位置进行均值计算，用训练集图像对应位置减去该位置对应均值，从而减小计算量。

4.3.3 实验平台

实验在 Windows10 操作系统上完成的。在 CPU 为 Intel(R) Xeon(R) Gold 5218 CPU、内存为 63G、GPU 为 RTX2080Ti 的环境下训练，再将训练好的模型在 CPU 为 i5 7300H、

内存为 16G、GPU 为 GTX1050Ti 的环境下进行测试。实验使用的框架为 Pytorch，同时采用了很多的第三方库保证代码的正常运行。

4.3.4 网络性能

ArcFace 人脸识别网络输入端口的图像大小为 128×128×1，人脸图像输入需要将整个图像进行灰度化处理后再输入识别网络。通过 LFW (Labeled Faces in the Wild)数据集进行 ArcFace 网络性能测试，在该数据集上，将人脸识别的阈值设置在 0.381 能够使得 LFW 数据集上人脸识别的准确率达到 96.5%。