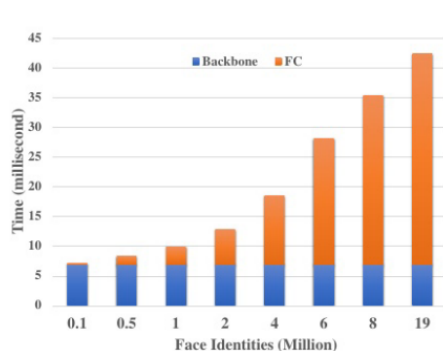


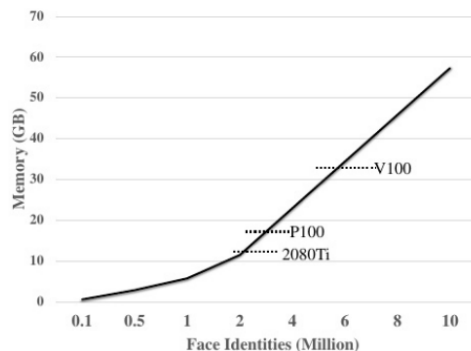
一种大规模人脸识别的有效训练方法

摘要: 由于超大规模和良好标记的数据集, 人脸识别在深度学习时代取得了重大进展。然而, 在超大规模的数据集上进行训练非常耗时, 并且占用了大量的硬件资源。因此, 设计一种高效的训练方法是必不可少的。全连接层(FC)的百万级维数导致了大量的计算和内存开销。为此, 我们提出了一种新的训练方法, 称为快速人脸分类(F2C), 以减轻时间和成本, 而不牺牲性能。该方法采用动态类池(DynamicClass Pool, DCP)动态存储和更新身份特征, 可以看作是 FC 层的替代品。DCP 具有体积小、不依赖于整个人脸身份的特点, 有效地节省了时间和成本。我们进一步在多个人脸基准和私有数据集上验证了所提出的 F2C 方法, 并显示了可比较的结果, 同时在识别精度和硬件成本方面, 速度比最先进的基于 fc 的方法更快。此外, 通过设计基于身份和基于实例的双重数据加载器, 进一步改进了该方法, 提高了 DCP 参数更新的效率。

简介: 深度神经网络(Deep Neural Networks, DNNs)在计算机视觉任务中取得了许多令人瞩目的成果[22, 23, 33, 34, 35, 36]。人脸识别可以看作是计算机视觉中最热门的研究课题之一。在过去的十年中, 已经发布了许多大规模且标记良好的数据集[8, 11, 43, 45, 49]。训练过程人脸识别的目的是学习与身份相关的嵌入空间, 在该嵌入空间中, 类内距离减小, 类间距离增大。先前的工作[8, 38, 39]已经证明, 在大数据集上的训练可以获得比小数据集更大的改进。为此, 学术界和工业界收集了包括 10 个甚至 1 亿个人脸身份在内的超大规模数据集。谷歌收集了 2 亿张人脸图像, 由 800 万个身份组成[24]。清华引入 WebFace260M[49], 包括 2.6 亿张人脸, 这是目前最大的公开人脸数据集, 实现了最先进的性能。总的来说, 这些超大规模的数据集大大提升了人脸识别性能。但是, 随着人脸身份的增长和硬件的限制, 在训练阶段主要存在两个问题。第一个问题源于训练时间和硬件资源占用。如图 1 所示, 当人脸身份达到 1000 万时, FC 层的时间成本和 GPU 内存占用都远远大于骨干网。为了解决这些问题, 以前的许多方法[1, 16]都侧重于减少 FC 层的时间和资源成本。以前的方法可以归纳为两类。一个[1]试图将整个 FC 分配给不同的 gpu, 引入了沉重的通信成本。另一个[46]试图通过从 FC 层随机选择一定比例的神经元来降低计算成本, 但它仍然需要存储整个 FC 参数。当标识达到 1000 万或 1 亿个时, 存储整个 FC 参数的成本非常高。如何有效降低高维 FC 层带来的计算和内存成本? 一个直观的想法是减少 FC 的大小或设计一个替代范例, 这在以前几乎没有被探索过。第二个问题与 FC 参数的更新效率和速度有关。正如[6]所指出的, 类中心的最优解实际上是这个类的所有样本的均值。具有罕见样本且采样频率非常低的身份将很少有机会通过其样本更新类中心, 这可能会妨碍特征表示。



(a) Comparison of backbone and FC time cost (ms).



(b) The memory occupancy of the FC layer at training phase (G).

为了解决上述问题,我们提出了一种针对超大规模人脸数据集的高效训练方法,称为 Faster faceClassification(F2C)。在 F2C 中,我们首先引入了名为 Gallery Net (G-Net)和 Probe Net (P-Net)的双主干,分别生成身份中心和提取人脸特征。G-Net 具有与 P-Net 相同的结构,并以移动平均的方式继承了 P-Net 的参数。考虑到超大规模训练中最耗时的部分位于 FC 层,我们提出了动态类池 (Dynamic Class Pool, DCP)来存储 G-Net 的特征,并计算每个小批中具有正样本(其身份出现在 DCP 中)的 logits。DCP 可以看作是 FC 层的替代品,它的尺寸比 FC 小得多,这就是为什么 F2C 相比 FC 层可以大大减少时间和资源成本的原因。对于负样本(其身份不出现在 DCP),我们最小化负样本和 DCP 之间的余弦相似度。为了提高 DCP 参数更新的效率和速度,我们设计了基于身份和基于实例的双重数据加载器。双数据加载器根据实例和身份从给定数据集中加载图像,生成用于训练的批次。最后,我们在几个人脸基准上进行了充分的实验,证明 F2C 可以达到与普通基于 fc 的方法相当的结果和更高的训练速度。F2C 在识别精度和硬件成本方面也比以前的方法获得了更好的性能。我们的贡献可以总结如下。

1)我们提出了一种用于超大规模人脸识别训练的高效训练方法 F2C,该方法旨在减少训练时间和硬件成本,同时保持与最先进的基于 fc 的方法相当的性能。

2)我们设计了 DCP 动态存储和更新身份特征,这是 FC 层的替代方案。DCP 的大小比 FC 小得多,并且独立于整个人脸身份,因此可以大大减少训练时间和硬件成本。

3)设计了基于身份和基于实例的双重数据加载器,提高了 DCP 参数的更新效率。

相关工作 人脸识别。由于大规模的数据集、先进的架构和损失函数,人脸识别已经取得了巨大的进步。大规模数据集在促进人脸识别性能[5]方面发挥着最关键的作用。这些数据集可以根据人脸身份的数量分为三个区间:1-10K, 11-100K, >100K。VGGFace[21]、VGGFace2[3]、UMD-Faces[2]、CelebFaces[28]、CASIA-WebFace[45]属于第一个区间。IMDB-Face[30]和 MS1MV2[5]的人脸识别在 11K ~ 100K 之间。Glnt360k[1]和 Webface260M[49]的人脸身份约为 0.36M 和 4M。许多先前的工作[1, 46, 49]表明,在更大的人脸身份数据集上进行训练可以比在

较小的数据集上获得更好的性能。因此，使用 WebFace260M 作为训练数据集，在 IJBC[19] 和 NIST-FRVT 挑战中获得了最先进的性能。基于这些数据集，已经提出了各种用于提高性能的 CNN 架构，如 VG-GNet[26]、GoogleNet[29]、ResNet[10]、AttentionNet[32] 和 MobileFaceNet[4]。对于损失函数，对比损失[28, 44]和三重损失[26]可能是很好的候选者。但它们的缺点是计算成本高，收敛速度慢。为此，研究人员试图探索新的度量学习损失函数来提升人脸识别性能。已经利用了几种基于边缘的 softmax 损失[5, 17, 31, 340, 41]并获得了最先进的结果。综上所述，目前的方法和大规模数据集在人脸识别方面已经取得了优异的性能，但训练时间和硬件成本仍然是训练阶段的瓶颈，特别是对于百万规模甚至更多的人脸身份数据集的训练。大规模 FC 层的加速。如图 1a 所示，当人脸识别达到 10M 时，时间成本主要集中在 FC 层，而不是卷积层。自 2001 年以来，研究人员在加速大规模 FC 训练方面进行了一些尝试。一个直观的想法是设计一个近似函数来降低计算成本，层次 Softmax (HSM)[7] 试图将多类分类器重新表述为二元分类器的层次结构。因此，可以通过给定样本只需要沿着从根到相应类的路径遍历的方式来降低训练成本。但是，所有的类中心都存储在 RAM 中，随着人脸身份的增加，检索时间也不容忽视。Zhang 等人。[46]提出了一种可以在每个迷你批中识别少量“活动类”的方法，该方法动态地构建动态类层次结构。然而，当人脸身份太大时，识别“活动类”也很耗时。一些公司，如谷歌和微软，试图将所有类别平均划分为多个 gpu。服务器间的通信成本也不容忽视。为了解决这个问题，Partial FC[1] 尝试在单个 GPU 服务器上训练大规模数据集，在每次迭代中随机使用 10% 的身份。然而，它仍然受到单个机器中 gpu 内存的限制。如图 1b 所示，Partial FC 只能在人脸识别数量不是超大($<10M$)的情况下工作，否则 gpu 仍然会耗尽内存。有几种基于成对的方法[12]，利用人脸对训练大规模数据集，而时间复杂度为 $O(Nk)$ ，其中 k 表示成对的大小。最新的相关工作 VFC[16] 通过构建一些虚拟 FC 参数来降低计算成本，但其性能与普通 FC 相比要低得多。与以往的工作不同，我们的 F2C 可以大大降低 FC 训练成本，并且与普通的基于 FC 的方法相比，可以达到相当的性能。

更快的人脸分类 在本节中，我们首先给出 F2C 的概述，以便简要了解我们的方法。然后介绍我们的动机和用于超大规模数据集训练的关键模块。之后，我们展示了对这些模块的理论/实证分析。最后，我们展示了训练细节，以便更好地复制。我们要解决的问题是在不造成明显性能下降的情况下，加快超大规模人脸数据集(人脸身份 $> 10M$) 的训练速度，降低硬件成本。为此，我们提出了用于超大规模人脸数据集训练的 F2C 框架。如图 2 所示，给定超大规模的人脸数据集，我们利用基于实例的加载器来生成一个实例批处理，就像数据加载器通常做的那

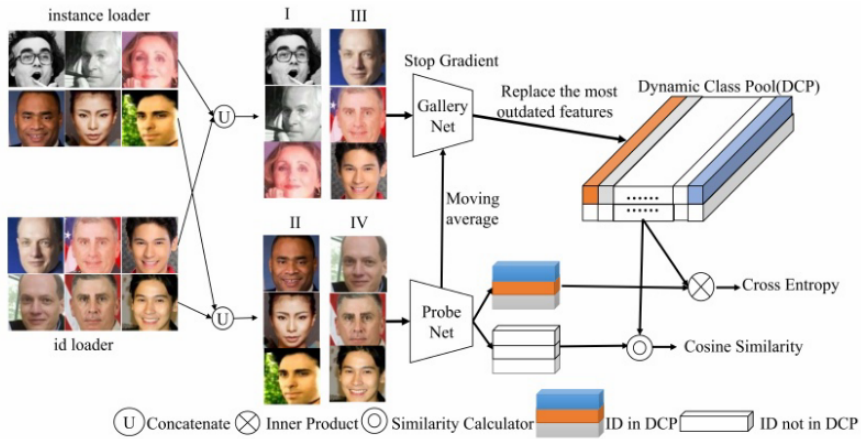
样。同时，基于身份的加载器从同一身份中随机选择两张图像，形成配对身份批处理。随后，我们将来自实例和配对身份批次的图像混合在一起，如图 2 所示，并将它们输入 G-Net 和 P-Net。受 MoCo[9] 的启发，G-Net 具有与 P-Net 相同的结构，并以移动平均方式继承 P-Net 的参数。利用 G-Net 和 P-Net 分别生成身份中心和提取人脸特征进行人脸识别。然后，我们引入 DCP 作为 FC 层的替代品。DCP 在每次迭代中随机初始化并由 G-Net 的特征更新。DCP 的更新策略遵循以下原则：用当前的特性替换 DCP 中最过时的部分特性。对于正样本，我们使用常见的交叉熵损失。对于负样本，我们最小化负样本和 DCP 之间的余弦相似度。整个 F2C 同时通过交叉熵损失和余弦相似度进行优化。在深入研究 F2C 之前，我们通过重新思考与 FC 层合作的损失函数来提供一些动机。为方便起见，我们

将 Softmax 考虑如下：

$$L = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{n_{ID}} e^{W_j^T x_i}} \quad (1)$$

其中 N 为批量大小，nID 为整张脸身份的数量。对于训练过程的每次迭代，分类器 $\{W_j\}_{j=1}^{n_{ID}}$ 的更新按照以下公式执行：

$$\frac{\partial L}{\partial W_k} = -\frac{1}{N} \sum_{i=1}^N (\delta_{ky_i} - \frac{e^{W_k^T x_i}}{\sum_{j=1}^{n_{ID}} e^{W_j^T x_i}}) x_i \quad (2)$$



F2C 的流水线。我们使用实例和 id 数据加载器生成混合批 ($I \cup III$, $II \cup IV$)，然后分别输入 G-Net 和 P-Net。G-Net 的特征将以 LRU 的方式更新 DCP，P-Net 的特征将与 DCP 一起计算损失。

显然，所有分类器 $\{W_j\}_{j=1}^{n_{ID}}$ 将在每次迭代中更新，这意味着每个分类器都有相同的优化机会。人脸识别的目标是通过同一身份的特征被拉到一起，属于不

同身份的特征被推到一边的机制来区分不同身份的人。由于超大规模数据集训练的主要问题是 FC 层的爆炸性大小，我们可以将整个 FC 视为一组分类器。为了降低计算成本，在训练过程中，我们可以直观地在每次迭代中对分类器的固定比例进行优化。具体来说，我们使用如下的向量来表示给定的分类器是否在优化队列中。

$$V = \{\nu_1, \dots, \nu_{n_{\text{ID}}}\}, \forall i, \nu_i \in \{0, 1\} \text{ and } \#\{\nu_i | \nu_i \neq 0\} = C \quad (3)$$

其中 C 是一个常数，表示优化队列的长度， $\nu_i = 0/1$ 表示分类器 W_i 是(不在)/(在)优化队列中。我们为此设置画出相应的目标。

$$\hat{L} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{n_{\text{ID}}} \nu_j e^{W_j^T x_i}} \quad (4)$$

分类器根据以下等式进行更新：

$$\frac{\partial \hat{L}}{\partial W_k} = -\frac{1}{N} \sum_{i=1}^N \left(\delta_{ky_i} - \frac{\nu_k e^{W_k^T x_i}}{\sum_{j=1}^{n_{\text{ID}}} \nu_j e^{W_j^T x_i}} \right) x_i \quad (5)$$

形式上，方程 5 与方程 2 类似，向量 V 的选择机制将直接影响分类器的更新过程。我们应该设计可行的选择机制，以便在每次迭代只更新部分分类器的约束下，更好地优化分类器。然而，这种直截了当的方法仍然承受着存储整组分类器的沉重压力。事实上，在我们的新框架中，我们只提供有限的空间来动态存储固定比例的分类器/特征。

在本小节中，我们将详细介绍我们的双数据加载器。为方便起见，我们将批大小表示为 M。实际上，我们利用基于实例的加载器从给定的人脸数据集中随机采样 M 张图像以获得实例批。同时，基于身份的加载器通过随机选择 M 个身份而不从整个身份中替换来提供身份批处理，并为每个身份采样两张图像。我们将实例批分成两部分，每部分取 M/2 张图像。对于配对身份批，我们按人脸身份进行分割，形成具有相同人脸身份集的两部分。我们将这四部分混合得到 $I \cup III; II \cup IV$ (如图 2 所示)，其中 \cup 表示集合的并集运算。

在本小节中，我们将详细介绍动态类池 (DCP)。受对象检测任务中的滑动窗口 [14] 的启发，我们可以利用一个动态身份组，通过迭代滑动整个人脸的身份。我们把这个滑动身份群称为 DCP，它可以看作是 FC 层的替代品。首先，我们定义一个张量 T，其大小为 $C \times K \times D$ ，用高斯分布初始化，其中 C 是 DCP 可以容

纳的人脸身份的容量或数量， K 表示属于同一身份的特征的数量（我们设置默认值为 $K = 2$ ）。我们在 DCP 中存储 F_g ，并在每次更新中使用 F_g 更新 DCP 中最过时的特征。更新规则类似于最近最少使用 (LRU) 策略，其可以表述为：

$$\begin{aligned} T[1 : C - M, :, :] &= T[M + 1 : C, :, :] \in \mathbb{R}^{(C-M) \times K \times D} \\ T[C - M + 1 : C, 0, :] &= F_g \in \mathbb{R}^{M \times K \times D} \end{aligned} \quad (7)$$

对于当前批处理，通过更新 DCP，我们为 DCP 中的每个身份获得伪特征中心，包括包含在 $II \cup IV$ 中的身份。如公式 6 所述，与 DCP 相比，P-Net 的特征可以分为两种类型。一个是 F_pDCP ，另一个是 $-F_pDCP$ 。对于 F_pDCP ，我们可以通过以下公式计算其对数：

$$P = \frac{1}{K} \sum_{i=1}^K \langle F_p^{DCP}, T[:, i, :] \rangle \in \mathbb{R}^{I \times C} \quad (8)$$

其中 $h \cdot, \cdot i$ 表示内积运算， P 表示 F_pDCP 的 logits。因此，我们可以将 Cross-Entropy loss 表述为：

$$L_{ce} = -\frac{1}{I} \sum_{i=1}^I \log \frac{e^{W_{y_i}^T P_i}}{\sum_{j=1}^C e^{W_j^T P_i}}, \quad (9)$$

对于 id 不在 DCP 中的 $-F_pDCP$ ，我们添加了一个约束来最小化 $-F_pDCP$ 和 T 之间的余弦相似度，可以表示为：

$$L_{cos} = \frac{1}{M - I} \sum_{i=1}^{M-I} \varphi(F_p^{-DCP}, \bar{T}), \quad (10)$$

总损失为 $L_{total} = L_{ce} + L_{cos}$ 。

DCP 如公式 4 和 9 所示，我们用于 DCP 的交叉熵损失通常与 FC 的损失相似。通过在方程 3 中对向量 V 的特殊可以用方程 4 的形式表示 L_{ce} 。为了进一步验证这一机制对 DCP 训练的影响，我们进行了一些实证分析。

Algorithm 1: Update Mechanism of DCP

Input:

DCP: $T \in \mathbb{R}^{C \times K \times D}$ initialized with Gaussian distribution.

Index for the identity batch: t .

Batch Size: M .

```
1 for  $1 \leq t \leq \frac{n_{ID}}{M}$  do
2   utilize the G-Net to extract features from  $t$ -th
     batch as the pseudo feature centers denoted as
      $F_g$ ;
3   if  $1 \leq t \leq \frac{C}{M}$ :
4     store  $F_g$  sequentially in those unoccupied
     position in DCP.
5   else:
6     update DCP as shown in Equation 7
7 end
```

实验 在本节中，我们首先简要回顾人脸识别领域的几个基准数据集。然后，我们进行消融研究，以评估每个模块的有效性以及 F2C 中超参数的设置。最后，我们将 F2C 与相关的最新方法进行了比较。

我们分别使用 MobileFaceNet、ResNet50 和 ResNet100 在 MS1MV2、Glnt360k 和 Webface42M 上训练 F2C (Webface42M 是原始 Web-face260M 的清洗版本，具有 2M ID 和约 42M 图像)。我们主要在以下 9 个学术数据集中展示了 F2C 的性能: LFW[11]、SLFW[11]、CFP[25]、CALFW[48]、CPLFW[47]、AGEDB[20]、YTF[42]、IJBC[19] 和 MegaFace[13]。LFW 收集自 Internet，其中包含 13233 张图片，5749 个 id。SLFW 与 LFW 相似，但 SLFW 的规模小于 LFW。CFP 收集名人的正面和侧面照片。CALFW 是 LFW 的跨年龄版本。CPLFW 与 CALFW 类似，但 CPLFW 包含更多的姿态变化图像。AGEDB 包含精确到年份的图像注释，无噪声标签。YTF 包括来自 YouTube 的 3425 个视频和 1595 个 id。IJBC 由 IJBB 更新而来，包括 21294 张 3531 个对象的图像。megface 旨在评估百万干扰物尺度下的人脸识别性能，包括一个大型画廊集和一个探针集。在这项工作中，我们使用 Facescrub 作为 megface 的探针集作为图库。

我们选择了 3 种不同的骨干网，并使用 MS1MV2、Glnt360k 和 Webface42M 作为训练数据集，在 FC 和 F2C 之间的 9 个学术基准上评估了性能。如表 1 所示，F2C 可以达到与 FC 相当的性能。我们还提供了这些数据集之间的平均性能，并

在最后一列中进行了演示，其中 F2C 仅低于 FC 1%。请注意，DCP 的大小仅占全部面部身份的 10%。我们对 F2C 的超参数和设置进行消融研究。在这里，

我们演示了使用 MobileFaceNet 和 ResNet50 在 MS1MV2 上的实验。单加载器还是双加载器？如方法学部分所述，双加载程序可以提高 DCP 的更新效率。为了评估加载器在 F2C 中的影响，我们使用了基于身份和基于实例的加载器的不同组合，并在表 2 中展示了结果。小数据集代表 LFW、SLFW、CFP、CALFW、本小节中的 CPLFW、AGEDB 和 YTF。我们展示了在 Small Datasets 上的平均精度。除非指定，否则 IJBC 使用 $\text{TPR@FAR}=1e-4$ 度量，而 Megafce 默认为 $\text{FPR@FAR}=1e-6$ 。使用基于实例的加载器或基于身份的加载器进行训练可以在小数据集上获得可比的结果。基于实例的加载器在 IJBC 和 megface 上的性能大大优于基于身份的加载器。可以这样解释，仅仅使用身份加载器并不能保证所有的图像都被采样。使用双数据加载器与每个单加载器相比，可以明显提高性能，这与我们的分析是一致的。请注意，为了进行公平的比较，结果是使用相同数量的样本馈送到模型，而不是使用相同数量的 epoch 获得的

Method	LFW	SLFW	CFP	CALFW	CPLFW	AGEDB	YTF	IJBC	MegaFace	Avg.
Training on MS1MV2										
FC-Mobile	99.04	98.80	96.94	94.37	88.37	96.73	97.04	92.29	90.69	94.92
F ² C-Mobile	98.93	98.57	97.16	94.53	87.80	96.47	97.24	91.06	89.30	94.56
FC-R50	99.78	99.55	98.80	95.76	92.01	98.13	98.03	95.74	97.82	97.29
F ² C-R50	99.50	99.45	98.46	95.58	90.58	97.83	98.16	94.91	96.74	96.80
Training on Glint360k										
FC-R50	99.83	99.71	99.07	95.71	93.48	98.25	97.92	96.48	98.64	97.67
F ² C-R50	99.71	99.53	98.30	95.23	91.60	97.88	97.76	94.75	96.73	96.83
Training on Webface42M										
FC-R100	99.83	99.81	99.38	96.11	94.90	98.58	98.51	97.68	98.57	98.15
F ² C-R100	99.83	98.80	99.33	95.92	94.85	98.33	98.23	97.31	98.53	97.90

探讨 K 对 DCP 的影响。K 表示属于同一恒等式的特征的个数。我们对表 4 中的 $K = 1$ 和 $K = 2$ 求值。由于 DCP 中的特征代表类别中心，因此直观的感觉是 K 越大，中心估计越可靠。实验结果也支持了我们的直觉。然而，我们必须在性能和存储之间做出权衡。更大的 K 意味着更好的性能，但代价是 GPU 内存和服务器之间的通信。因此，我们在 DCP 中默认设置 $K = 2$ 。

Backbone	Method	Small Datasets	IJBC	MegaFace
Mobile	ID.L	94.20	82.30	79.19
	Ins.L	94.24	89.30	86.40
	Dua.L	95.29	91.06	89.30
ResNet50	ID.L	96.70	91.75	93.65
	Ins.L	96.08	92.06	92.74
	Dua.L	97.07	94.91	96.74

表3:单网或双网的评价。

Backbone	Method	Small Datasets	IJBC	MegaFace
Mobile	Single	93.90	88.07	82.69
	Dual	95.29	91.06	89.30
ResNet50	Single	95.55	92.26	92.98
	Dual	97.07	94.91	96.74

表4:评价K的数量。

Backbone	K	Small Datasets	IJBC	MegaFace
Mobile	1	95.19	90.75	88.31
	2	95.29	91.06	89.30
ResNet50	1	96.58	94.38	96.49
	2	97.07	94.91	96.74

双数据加载器内的比率。我们将基于实例和基于身份的加载器之间的大小比例默认设置为 1:1。为了进一步探索双数据加载器中比率的影响，我们在表 5 中展示了实验。我们利用 ResNet50 作为主干来训练 MS1MV2 数据集。我们发现，双数据加载器内的默认比率在大多数数据集上达到了最高的结果，特别是在挑战 IJBC 和 megface 上面。

结论: 本文提出了一种用于超大规模人脸识别训练的高效训练方法 F2C，其主要创新点是动态类池 (Dynamic ClassPool, DCP) 用于存储和更新人脸身份特征，以替代 FC 和双加载器来帮助 DCP 高效更新。综合实验和分析的结果表明，我们的方法可以减少硬件成本和训练时间，并获得与最先进的基于 fc 的方法相当的性能。更广泛的影响。所提出的方法在人脸训练数据集上进行了验证，由于种类繁多，该方案可以扩展到其他数据集和情况。然而，它不包含任何作者进行的涉及影响伦理或人权的任何研究。

本科毕业设计(论文)指导记录表

学院	信息工程学院（人工智能学院）	专业	人工智能
学生姓名	朱金康	学号	1520203586
指导教师	梁苗苗		
课题题目	面向大规模人脸识别模型的深度学习方法研究		
<div>指导记录 1</div> <div>主题： 论文大纲与开题指导</div> <div>指导时间： 2024 年 12 月 1 日</div> <div>主要工作：</div> <div>按照任务书的要求和文献参考，确定论文的大纲以及论文的具体内容，指导填写开题报告。</div> <div>指导教师签名：梁苗苗</div> <div>2025 年 4 月 30 日</div>			
<div>指导记录 2</div> <div>主题： 开题报告的修正</div> <div>指导时间： 2024 年 12 月 4 日</div> <div>主要工作：</div> <div>根据导师的要求，进行开题报告的规范化的修改。</div> <div>指导教师签名：梁苗苗</div> <div>2025 年 4 月 30 日</div>			
<div>指导记录 3</div> <div>主题： 开题报告的修正</div> <div>指导时间： 2024 年 12 月 4 日</div> <div>主要工作：</div> <div>根据导师的要求，进行开题报告的规范化的修改。</div> <div>指导教师签名：梁苗苗</div>			

2025 年 4 月 30 日

指导记录 4

主题：

外文译文的整理

指导时间：

2025 年 1 月 6 日

主要工作：

根据导师的指导选取了合适的论文并进行了翻译整理上面的知识点，整理了很多的相关的外文文献。

指导教师签名：梁育南

2025 年 4 月 30 日

指导记录 5

主题：

进度报告与论文的书写规范

指导时间：

2025 年 3 月 10 日

主要工作：

汇报当前的进度，配置好了实验要用到的环境，搭建好了实验用到的网络，正在调整代码里面存在的 bug，根据老师的指导明确了后续的进展方向。

指导教师签名：梁育南

2025 年 4 月 30 日

指导记录 6

主题：

论文初稿撰写

指导时间：

2025 年 3 月 20 日

主要工作：

完成了论文的初稿部分的研究背景，现状，研究意义部分的撰写。

指导教师签名：梁育南

2025 年 4 月 30 日

指导记录 7

主题:

动态队列池的完善

指导时间:

2025 年 3 月 24 日

主要工作:

解决了代码里面出现的 bug, 实现了人脸识别需要的队列池, 动态的更新里面的人脸数据。

指导教师签名: 梁育南

2025 年 4 月 30 日

指导记录 8

主题:

参数调整和结果分析

指导时间:

2025 年 4 月 7 日

主要工作:

对实验里面遇到的超参数进行了参数的调整, 观察合适的参数并分析了原因, 对实验的结果的好坏采用了常见的量来量化, 以求得到当前模型的好坏程度。

指导教师签名: 梁育南

2025 年 4 月 30 日

指导记录 9

主题:

FFC 新方法的构建与训练

指导时间:

2025 年 4 月 14 日

主要工作:

构建了 FFC 的新方法进行人脸识别, 构建相关的网络和用到的模块。

指导教师签名: 梁育南

2025 年 6 月 5 日

指导记录 10

主题:

FFC 方法对大量数据集进行训练

指导时间:

2025 年 4 月 21 日

主要工作:

使用新的方法进行训练，采用 LRU 来构建动态的人脸数据库实现大范围的人脸训练，并且减少对资源的消耗。

指导教师签名：梁育南

2025 年 6 月 5 日