

山 人 七



谷歌人脸识别系统FaceNet解析



狗头山人七 · 9 个月前


论文笔记：[FaceNet- A Unified Embedding for Face Recognition and Clustering](#)

简介：

近年来，人脸识别技术取得了飞速的进展，但是人脸验证和识别在自然条件中应用仍然存在困难。本文中，作者开发了一个新的人脸识别系统：FaceNet，可以直接将人脸图像映射到欧几里得空间，空间距离的长度代表了人脸图像的相似性。只要该映射空间生成，人脸识别，验证和聚类任务就可以轻松完成。文章的方法是基于深度卷积神经网络。FaceNet在LFW数据集上，准确率为0.9963，在YouTube Faces DB数据集上，准确率为0.9512。

1 前言

知

 写文章

登录

和聚类（寻找类似的人？）。FaceNet采用的方法是通过卷积神经网络学习将图像映射到欧几里得空间。空间距离直接和图片相似度相关：同一个人的不同图像在空间距离很小，不同人的图像在空间中有较大的距离。只要该映射确定下来，相关的人脸识别任务就变得很简单。

当前存在的基于深度神经网络的人脸识别模型使用了分类层（classification layer）：中间层为人脸图像的向量映射，然后以分类层作为输出层。这类方法的弊端是不直接和效率低。

与当前方法不同，FaceNet直接使用基于triplets的LMNN（最大边界近邻分类）的loss函数训练神经网络，网络直接输出为128维度的向量空间。我们选取的triplets（三联子）包含两个匹配脸部缩略图和一个非匹配的脸部缩略图，loss函数目标是通过距离边界区分正负类，如图1-1所示。



Figure 2. **Model structure.** Our network consists of a batch input layer and a deep CNN followed by L_2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

图1-1 模型结构

脸部缩略图为紧密裁剪的脸部区域，没有使用2d，3d对齐以及放大转换等预处理。

本文中，作者探索了两类深度卷积神经网络。第一类为Zeiler&Fergus研究中使用的神经网络，我们在网络后面加了多个 $1 \times 1 \times d$ 卷积层；第二类为Inception网络。模型结构的末端使用

到同一个空间。而triplet loss尝试将一个个体的人脸图像和其它人脸图像分开。下文包含以下内容：

- 三联子 (triplets) loss
- triplets筛选方法
- 模型结构描述
- 实验结果
- 评论

2,三联子 (triplets) loss

模型的目的是将人脸图像X embedding入 d 维度的欧几里得空间 $f(x) \in R^d$ 。在该向量空间内，我们希望保证单个个体的图像 $x_i^a(anchor)$ 和该个体的其它图像 $x_i^p(positive)$ 距离近，与其它个体的图像 $x_i^n(negative)$ 距离远。如图5-1所示：



Figure 3. The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

图2-1 triplet loss示意图

The loss that is being minimized is then $L =$

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ . \quad (2)$$

其中， α 为positive/negative的边界。

3, triplets筛选

triplets 的选择对模型的收敛非常重要。如公式1所示，对于 x_i^a ，我们需要选择同一个体的不同图片 x_i^p ，使 $\operatorname{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$ ；同时，还需要选择不同个体的图片 x_i^n ，使得 $\operatorname{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$ 。在实际训练中，跨越所有训练样本来计算argmin和argmax是不现实的，还会由于错误标签图像导致训练收敛困难。实际训练中，有两种方法来进行筛选：

一，每隔n步，计算子集的argmin和argmax。

二，在线生成triplets，即在每个mini-batch中进行筛选positive/negative样本。

本文中，我们采用在线生成triplets的方法。我们选择了大样本的mini-batch（1800样本/batch）来增加每个batch的样本数量。每个mini-batch中，我们对单个个体选择40张人脸图片作为正样本，随机筛选其它人脸图片作为负样本。负样本选择不当也可能导致训练过早进入局部最小。为了避免，我们采用如下公式来帮助筛选负样本：

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 . \quad (3)$$

采用adagrad优化器，使用随机梯度下降法训练CNN模型。在cpu集群上训练了1000-2000小时。边界值 α 设定为0.2。总共实验了两类模型，参数如表4-1和表4-2所示。

表4-1 CNN模型1

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Table 1. **NN1**. This table show the structure of our Zeiler&Fergus [22] based model with 1×1 convolutions inspired by [9]. The input and output sizes are described in $rows \times cols \times \#filters$. The kernel is specified as $rows \times cols, stride$ and the maxout [6] pooling size as $p = 2$.

type	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj (p)	params	FLOPS
conv1 (7×7×3, 2)	112×112×64	1							9K	119M
max pool + norm	56×56×64	0						m 3×3, 2		
inception (2)	56×56×192	2		64	192				115K	360M
norm + max pool	28×28×192	0						m 3×3, 2		
inception (3a)	28×28×256	2	64	96	128	16	32	m, 32p	164K	128M
inception (3b)	28×28×320	2	64	96	128	32	64	L_2 , 64p	228K	179M
inception (3c)	14×14×640	2	0	128	256,2	32	64,2	m 3×3,2	398K	108M
inception (4a)	14×14×640	2	256	96	192	32	64	L_2 , 128p	545K	107M
inception (4b)	14×14×640	2	224	112	224	32	64	L_2 , 128p	595K	117M
inception (4c)	14×14×640	2	192	128	256	32	64	L_2 , 128p	654K	128M
inception (4d)	14×14×640	2	160	144	288	32	64	L_2 , 128p	722K	142M
inception (4e)	7×7×1024	2	0	160	256,2	64	128,2	m 3×3,2	717K	56M
inception (5a)	7×7×1024	2	384	192	384	48	128	L_2 , 128p	1.6M	78M
inception (5b)	7×7×1024	2	384	192	384	48	128	m, 128p	1.6M	78M
avg pool	1×1×1024	0								
fully conn	1×1×128	1							131K	0.1M
L2 normalization	1×1×128	0								
total									7.5M	1.6B

Table 2. **NN2**. Details of the NN2 Inception incarnation. This model is almost identical to the one described in [16]. The two major differences are the use of L_2 pooling instead of max pooling (m), where specified. The pooling is always 3×3 (aside from the final average pooling) and in parallel to the convolutional modules inside each Inception module. If there is a dimensionality reduction after the pooling it is denoted with p. 1×1, 3×3, and 5×5 pooling are then concatenated to get the final output.

5, 实验结果

作者采用了约8million个个体的将近100million-200million张人脸缩略图。人脸缩略图通过脸部检测器紧密裁剪生成。最后，在四类数据集上评价零FaceNet：

- hold-out 测试集：从训练集中分出100million图像作为测试集。
- 个人照片：总共包括12k个人照片。
- 学术数据集：我们采用了LFW数据集和Youtube Faces DB。

5.1 计算量与准确率权衡

在测试中，随着神经网络深度增加，计算量增加，准确率也增加，如表5-1和图5-1所示。

architecture	VAL
NN1 (Zeiler&Fergus 220×220)	$87.9\% \pm 1.9$
NN2 (Inception 224×224)	$89.4\% \pm 1.6$
NN3 (Inception 160×160)	$88.3\% \pm 1.7$
NN4 (Inception 96×96)	$82.0\% \pm 2.3$
NNS1 (mini Inception 165×165)	$82.4\% \pm 2.4$
NNS2 (tiny Inception 140×116)	$51.9\% \pm 2.9$

Table 3. **Network Architectures.** This table compares the performance of our model architectures on the hold out test set (see section 4.1). Reported is the mean validation rate VAL at $10E-3$ false accept rate. Also shown is the standard error of the mean across the five test splits.

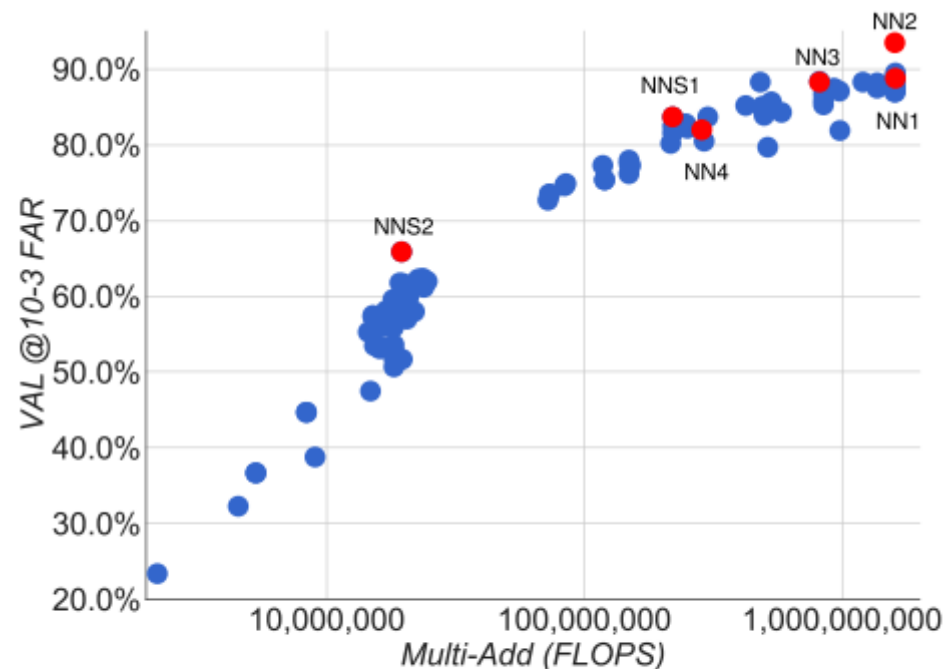


Figure 4. **FLOPS vs. Accuracy trade-off.** Shown is the trade-off between FLOPS and accuracy for a wide range of different model sizes and architectures. Highlighted are the four models that we focus on in our experiments.

图5-1 计算量（FLOPS）与准确率关系

5.2 CNN模型结构对loss的影响

作者考察了不同CNN模型对结果的影响，如图5-2所示。

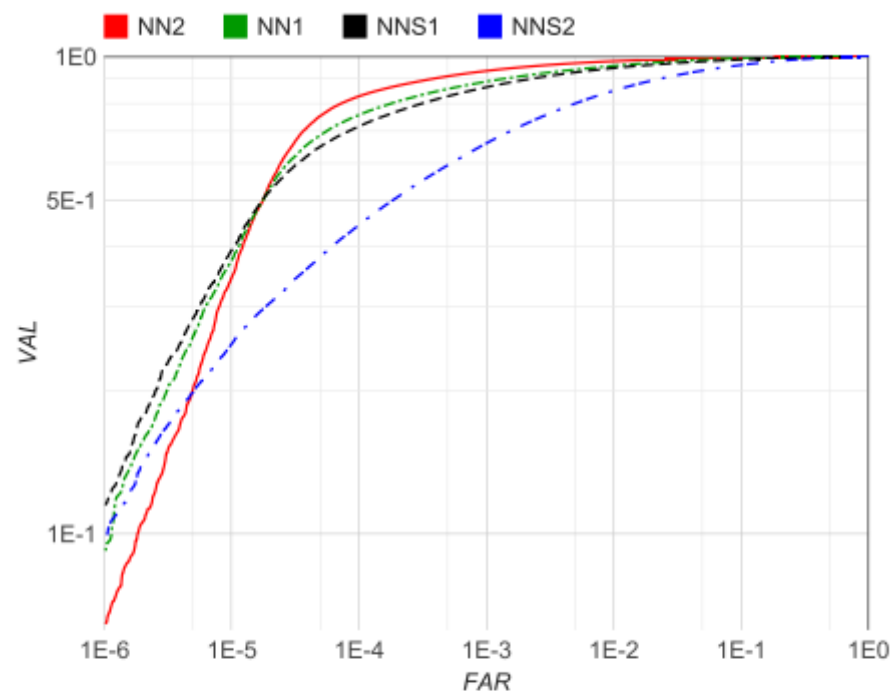


Figure 5. **Network Architectures.** This plot shows the complete ROC for the four different models on our personal photos test set from section 4.2. The sharp drop at 10E-4 FAR can be explained by noise in the groundtruth labels. The models in order of performance are: **NN2**: 224×224 input Inception based model; **NN1**: Zeiler&Fergus based network with 1×1 convolutions; **NNS1**: small Inception style model with only 220M FLOPS; **NNS2**: tiny Inception model with only 20M FLOPS.

图5-2 网络结构对VAL的影响

5.3 图像质量对结果的影响

示。

表5-2 图像质量（像素值）对结果的影响

jpeg q	val-rate	#pixels	val-rate
10	67.3%	1,600	37.8%
20	81.4%	6,400	79.5%
30	83.9%	14,400	84.5%
50	85.5%	25,600	85.7%
70	86.1%	65,536	86.4%
90	86.5%		

Table 4. **Image Quality.** The table on the left shows the effect on the validation rate at 10E-3 precision with varying JPEG quality. The one on the right shows how the image size in pixels effects the validation rate at 10E-3 precision. This experiment was done with NN1 on the first split of our test hold-out dataset.

5.4 Embedding维度对结果的影响

作者测试了不同的embedding维度，结果如表5-3所示，发现128维度是最为合适的。

表5-3 不同输出维度对结果的影响

#dims	VAL
64	86.8% \pm 1.7
128	87.9% \pm 1.9
256	87.7% \pm 1.9
512	85.6% \pm 2.0

Table 5. **Embedding Dimensionality.** This Table compares the effect of the embedding dimensionality of our model NN1 on our hold-out set from section 4.1. In addition to the VAL at 10E-3 we also show the standard error of the mean computed across five splits.

5.5 训练数据量对结果的影响

随着训练数据量的增加，准确率也随之增加，如表5-4所示。

表5-4 训练数据量与VAL

#training images	VAL
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

Table 6. **Training Data Size.** This table compares the performance after 700h of training for a smaller model with 96x96 pixel inputs. The model architecture is similar to NN2, but without the 5x5 convolutions in the Inception modules.

5.6 评价结果

FaceNet在LFW数据集上取得了 $99.63\% \pm 0.09$ 的准确率；在Youtube Faces DB数据集上获得了 $95.12\% \pm 0.39$ 的结果。在个人照片的数据集上，对单个个体进行embedding后聚类测试，结果如图5-3所示。



Figure 7. **Face Clustering.** Shown is an exemplar cluster for one user. All these images in the users personal photo collection were clustered together.

6, 评论

FaceNet是google的工作，工作量非常大，结果也很好。FaceNet是一种直接将人脸图像embedding进入欧几里得空间的方法。该模型的优点是只需要对图片进行很少量的处理（只需要裁剪脸部区域，而不需要额外预处理，比如3d对齐等），即可作为模型输入。同时，该模型在数据集上准确率非常高。未来的工作可以有几个方向：

- 一，分析错误的样本，进一步提高识别精度，特别是增加模型在现实场景中的识别精度。
- 二，以该模型为基础，将其用于现实应用开发中。（预告：后续文章中，我将对使用FaceNet进行人脸识别的项目源码进行解析，敬请关注）
- 三，减少模型大小，减少对cpu计算量的消耗，以及减少训练时间（作者在cpu集群上需要1000-2000小时的训练。）

深度学习（Deep Learning）

人脸识别



22

☆ 收藏

📄 分享

🚩 举报



4 条评论

写下你的评论...

5 个月前



黄小贝

有源码么？

4 个月前



狗头山人七 (作者) 回复 **黄小贝**

 查看对话

兄弟，github上有。

4 个月前



陈勇

有研究github上的facenet的嘛？我想一起讨论哈问题（我的QQ:770762710）

2 天前