

# 小游戏及其背后的一些技术和应用

基于超大规模分类的人脸识别和以图搜图(ReID)

张德兵

算法部负责人&首席科学家

格灵深瞳

[debingzhang@deepglint.com](mailto:debingzhang@deepglint.com)

# 目录

## Part.1 (by 小美)

回顾前四期技术谜题

详解《The Second Chance》谜题

## Part.2 (by 张德兵博士)

第一名通关玩家Solution解法

大规模人脸训练及ReID的工业经验

# 很赞的图普技术谜题系列



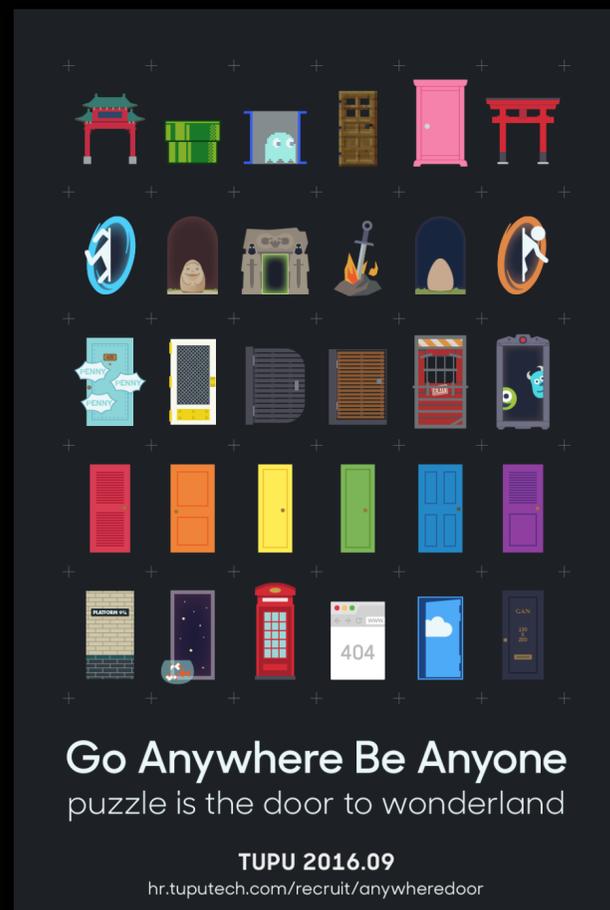
you can you up

```
import DL as gun
import Customers as metric

if __name__ == "__main__":
    result = gun.FCN("001100000111100000110001011000110011001000110000\
011110000110010000110010001100000111100000110011\
011001000110010100110000011110000011001001100010\
00110010")

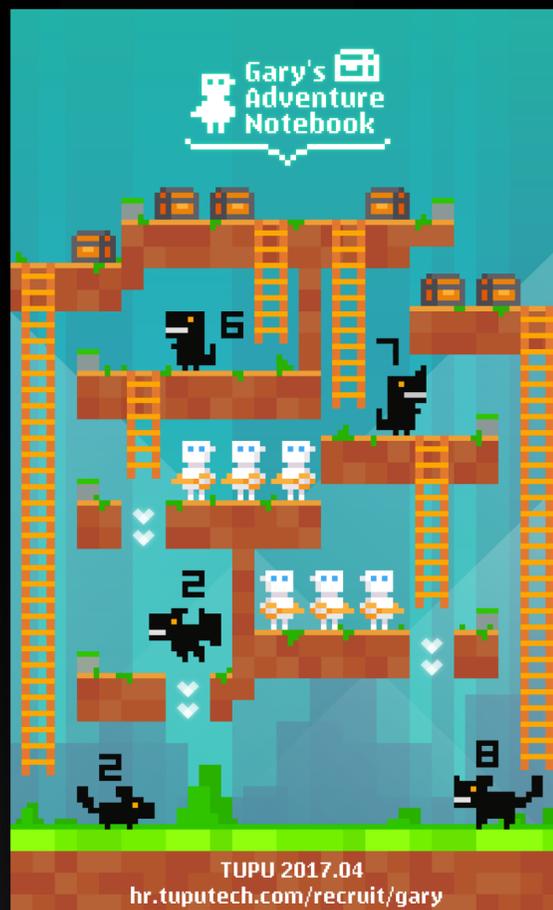
score = request("hr.tuputech.com/recruit/mario", result)
if metric.mean_IU(score) > threshold:
    print "Welcome"
```

**TUPU**  
2016.06



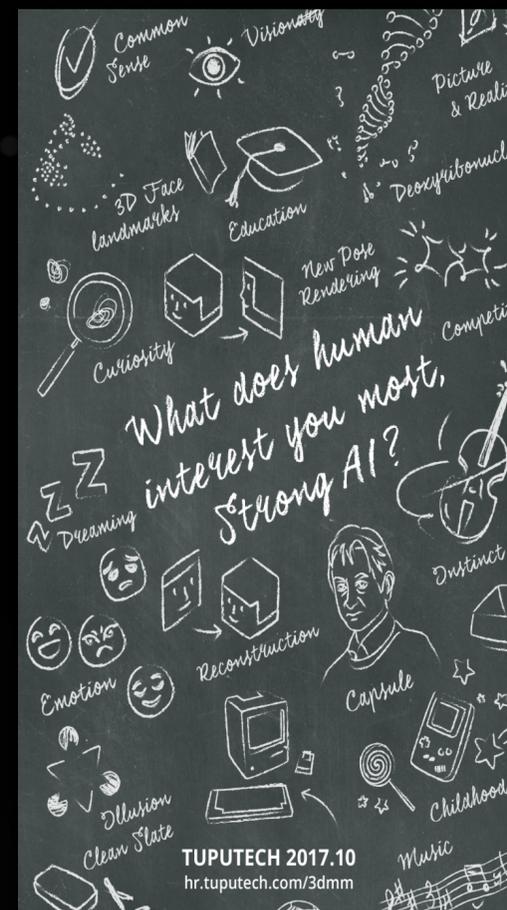
Go Anywhere Be Anyone  
puzzle is the door to wonderland

TUPU 2016.09  
[hr.tuputech.com/recruit/anywheredoor](http://hr.tuputech.com/recruit/anywheredoor)



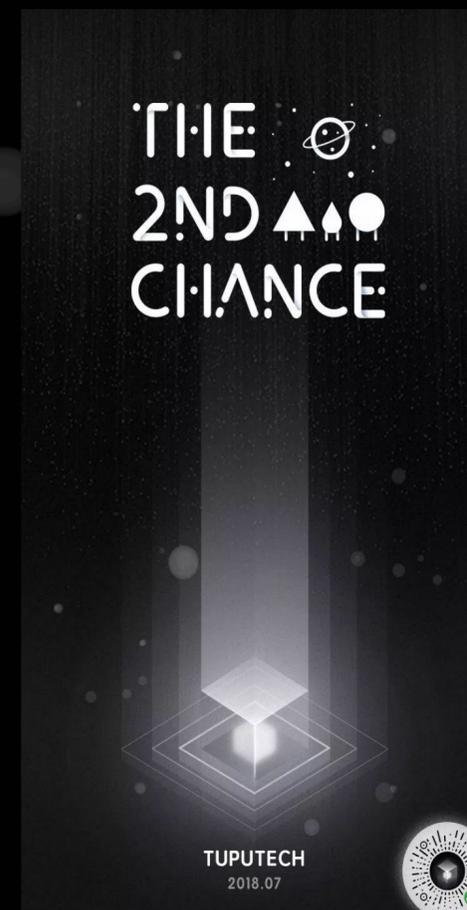
Gary's Adventure Notebook

TUPU 2017.04  
[hr.tuputech.com/recruit/gary](http://hr.tuputech.com/recruit/gary)



Common Sense Visionary  
3D Face Landmarks Education  
Curiosity New Pose Rendering  
Picture & Reality  
Deoxyribonucleic  
What does human interest you most, Strong AI?  
Dreaming  
Emotion  
Reconstruction  
Clean Slate  
Distinct  
Capsule  
Childhood  
Music

TUPUTECH 2017.10  
[hr.tuputech.com/3dmm](http://hr.tuputech.com/3dmm)



THE 2ND CHANCE

TUPUTECH  
2018.07

# 小游戏的通关过程

0. 发现图普又出新游戏了, 瞄了下是科幻风  $x$
1. 读题, 发现跟人脸识别有关  $x+10\text{min}$
2. 把可行路线遍历一遍, 这样出题真赞  $x+30\text{min}$
3. 仔细看看题目&查找e和pi  $x+1\text{h}$
4. 写一部分代码  $x+1.5\text{h}$
5. 回家  $x+2\text{h}$
6. coding, debug, 发现小trick  $x+3\text{h}$
7. 提交试试, 居然过了, 试了几个错误答案发现过不了...

# 小游戏背后的技术

1. 超大规模的多分类问题  $400W(2^{**}22) \times 256 = 1G$  参数
2. 题目保证了即使暴力，单卡也可以解决
3. 但真实问题（比如1亿类）怎么处理呢？

# 小游戏背后的技术

区别对比	学术界	工业界
图片量	百万级	亿级
每个ID的平均人脸数量	几十	几十张/2张
人脸ID数	小于10万	百万级——亿级
显存占用（仅仅参数矩阵W） $Y = \text{Softmax}(W * F)$ 中，F为人脸特征，W是参数 W: NumClass*FeatDim(如512)	500M	50G(按一千万ID计算)
训练时间	与ImageNet相当	100X

# 小游戏背后的技术

1. 超大规模的多分类问题  $400W(2^{**22}) \times 256 = 1G$  参数
2. 真实问题（比如1亿类）怎么处理呢？
3. 肯定没有办法通过hack迭代次数或者用更少的类别了
4. 肯定是分布式了，但不是数据并行
5. 先一个精确做法

# 精确的多机分布式训练策略

N计算节点	每节点M卡	Batch数据并行	数据并行特征计算	特征汇聚	模型并行	模型并行FC计算	通信优化(FW+BP)	梯度汇聚
Machine_1	GPU_1	batch_1_1	$F_{1_1} = \text{CNN}(\text{batch}_{1_1})$	F	W_1_1	$F * W_{1_1}$	grad_F_1_1, grad_W_1_1	grad_F
	GPU_2	batch_1_2	$F_{1_2} = \text{CNN}(\text{batch}_{1_2})$		W_1_2	$F * W_{1_2}$	grad_F_1_2, grad_W_1_2	
	...	...	...		...	...	...	
	GPU_M	batch_1_M	$F_{1_M} = \text{CNN}(\text{batch}_{1_M})$		W_1_M	$F * W_{1_M}$	grad_F_1_M, grad_W_1_M	
Machine_2	GPU_1	batch_2_1	$F_{2_1} = \text{CNN}(\text{batch}_{2_1})$		W_2_1	$F * W_{2_1}$	grad_F_2_1, grad_W_2_1	
	GPU_2	batch_2_2	$F_{2_2} = \text{CNN}(\text{batch}_{2_2})$		W_2_2	$F * W_{2_2}$	grad_F_2_2, grad_W_2_2	
	...	...	...		...	...	...	
	GPU_M	batch_2_M	$F_{2_M} = \text{CNN}(\text{batch}_{2_M})$		W_2_M	$F * W_{2_M}$	grad_F_2_M, grad_W_2_M	
...	...	...	...		...	...	...	
Machine_N	GPU_1	batch_N_1	$F_{N_1} = \text{CNN}(\text{batch}_{N_1})$		W_N_1	$F * W_{N_1}$	grad_F_N_1, grad_W_N_1	
	GPU_2	batch_N_2	$F_{N_2} = \text{CNN}(\text{batch}_{N_2})$		W_N_2	$F * W_{N_2}$	grad_F_N_2, grad_W_N_2	
	...	...	...		...	...	...	
	GPU_M	batch_N_M	$F_{N_M} = \text{CNN}(\text{batch}_{N_M})$		W_N_M	$F * W_{N_M}$	grad_F_N_M, grad_W_N_M	

# 小游戏背后的技术

数据并行+模型并行的思路优势：

1. 实现相对简单，把模型的显存占用和计算量都均匀分散到了每个GPU
2. 不增加额外通信带宽(甚至降低了FC层的梯度更新所需带宽)，10G网络环境可以支持100卡以上的训练，高效支持几千万类的人脸识别（512维特征），甚至上亿类的人脸识别(128维特征)，接近线性加速
3. 支持大部分主流损失函数扩展(Margin, Normalization等等)

思考：数值稳定性、哪些地方需要通信？

# 大规模分类模型的近似

1. 特征维度的压缩(512->128), 参数精度的压缩(fp32, fp16, int8)
2. 梯度的稀疏化/相邻类中心的梯度优化/OHEM
3. 基于动态聚类结构/树形结构的类中心
4. [Open]参数的冗余度是非常高的, 那有没有更快、更精确的近似策略呢?

# 小游戏背后的应用

1. 人脸识别
2. 车辆以图搜图（不利用车牌）
3. 行人/非机动车/物品以图搜图
4. 关联性应用

可能会设计到的各种技术或问题：

高效的训练框架、大规模训练集制作、模型/损失函数设计、GAN、Domain Transfer、关键点/姿态、分割、遮挡、Attention等等