



从 0 到 1 : 机器学习的探索与实践

陈爱珍



- **机器学习概念**
- 机器学习平台
- 人脸识别的实践

人工智能时代



1760

1860

1960

2020 ?



翻译



记者



助理



保安



司机



销售



客服



交易员



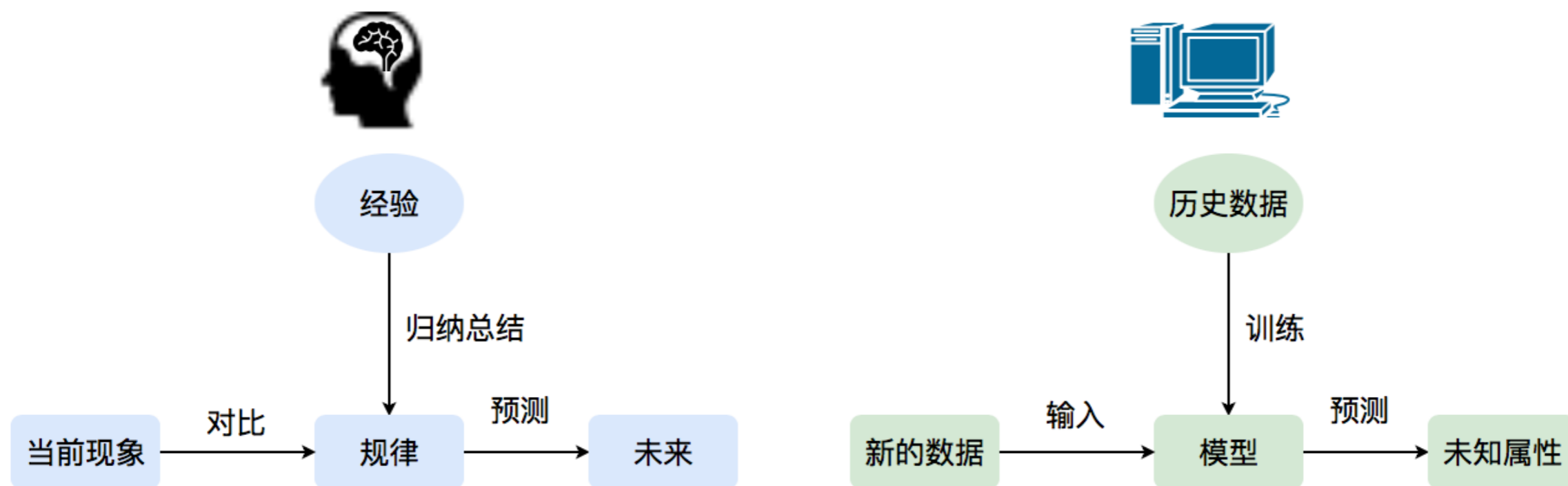
会计



保姆

未来人工智能将取代 90% 的职业

人脑学习 VS 机器学习



机器学习可以解决的问题

- **分类问题**

根据数据样本上抽取出的特征，判定其属于有限个类别中的哪一个。如图像内容识别识别。

- **回归问题**

根据数据样本上抽取出的特征，预测一个连续值的结果。如票房预测。

- **聚类问题**

根据数据样本上抽取出的特征，让样本抱抱团(相近/相关的样本在一团内)。如用户群体划分。

- **关联问题**

根据数据样本上抽取出的特征，发现数据背后存在的某种规则或者联系。如交叉销售或者捆绑销售。

机器学习的应用

计算机视觉

图片搜索
人脸识别
车牌识别
扫描文字识别
图片内容识别
视频内容识别

自然语言处理

搜索引擎智能匹配
文本内容理解
文本情绪判断
语音识别
输入法
机器翻译

精准营销

新闻推送
歌曲推荐
猜你喜欢
购买转化
商品推荐
针对性营销

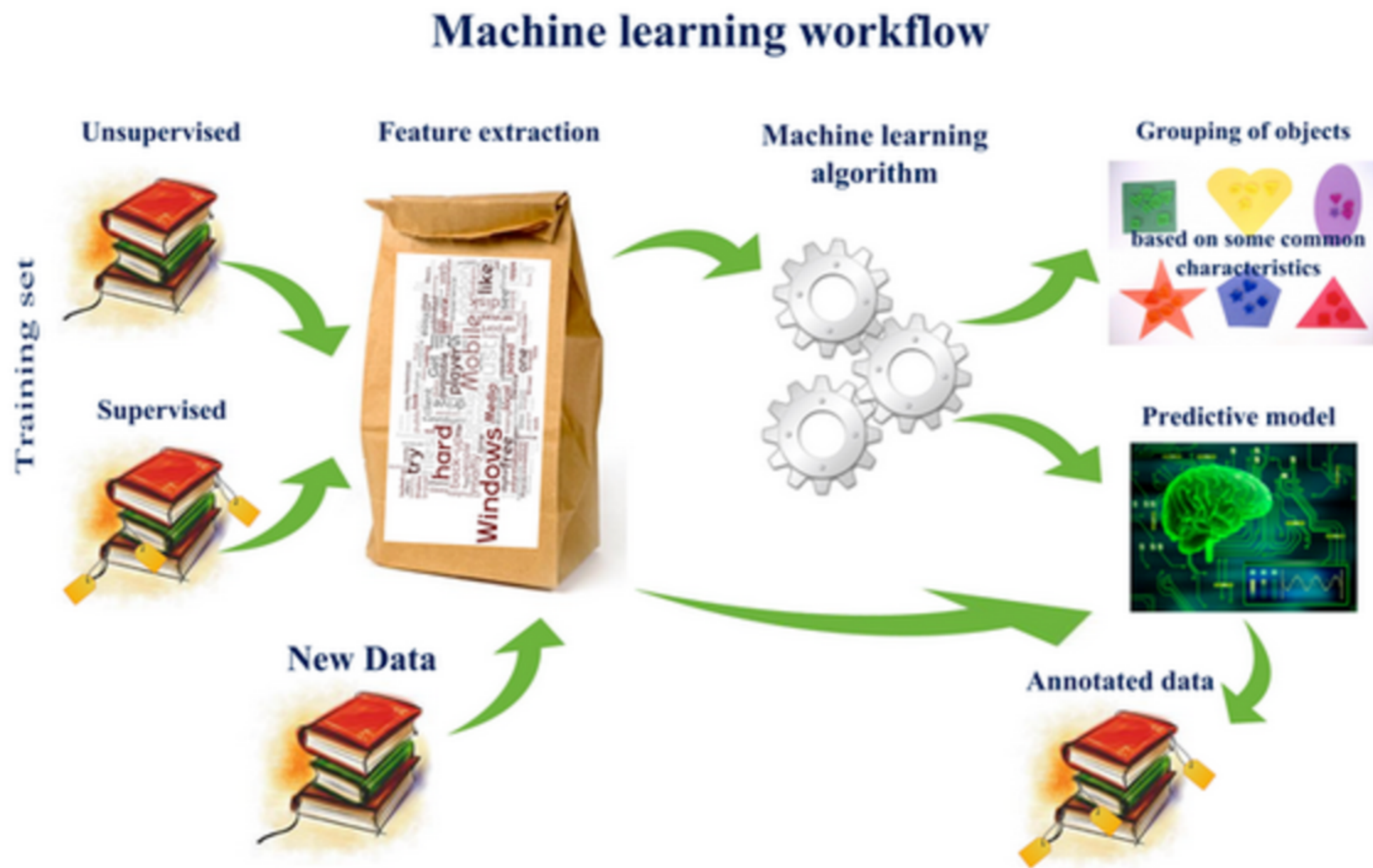
社会网络分析

用户画像
网络关联分析
欺诈作弊发现
热点发现

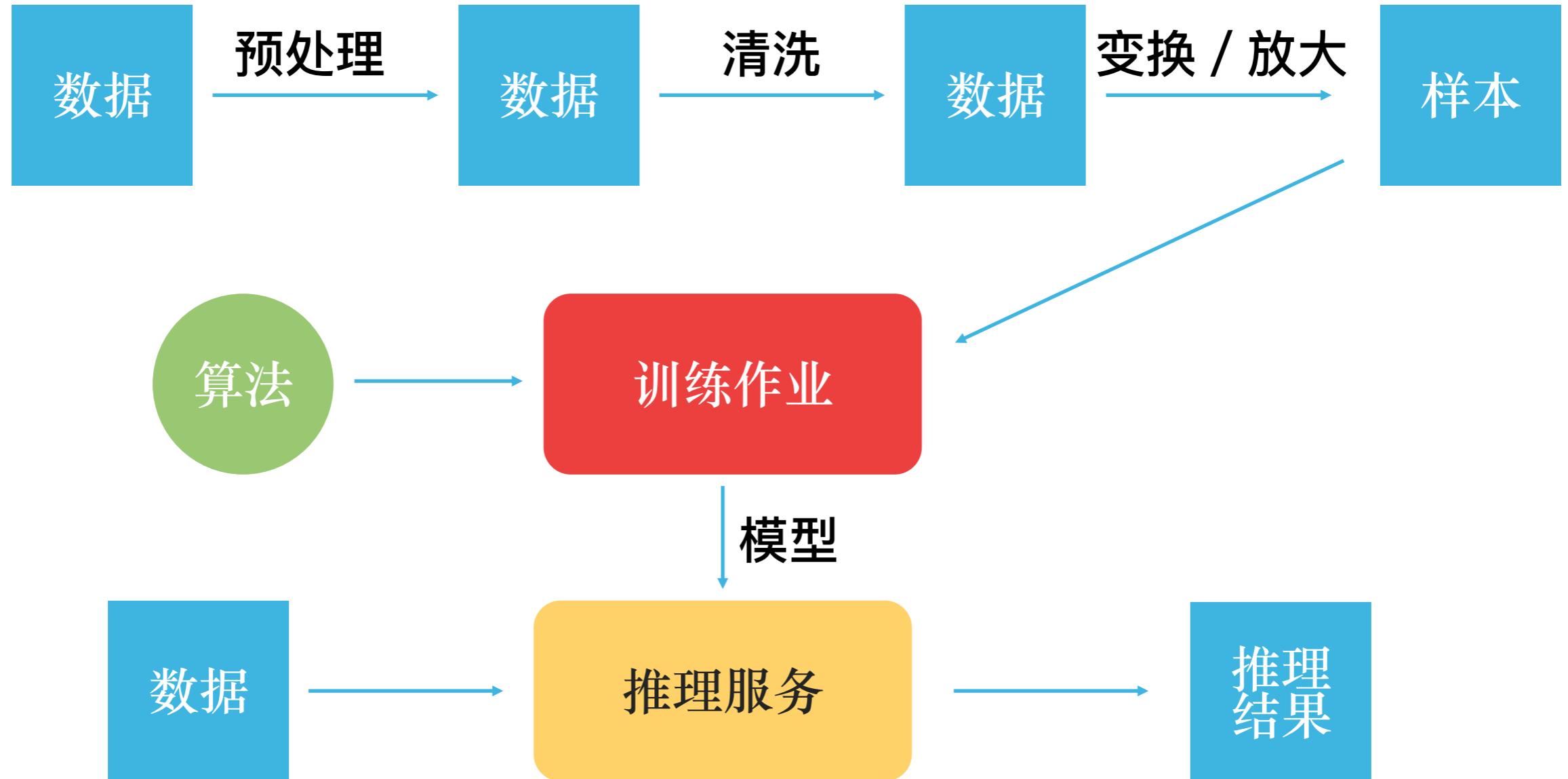


- 机器学习概念
- **机器学习平台**
- 人脸识别的实践

机器学习的工作流



机器学习的工作流



机器学习平台



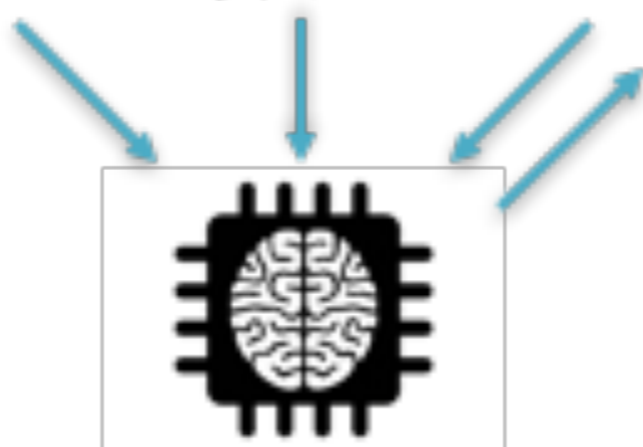
数据



算法



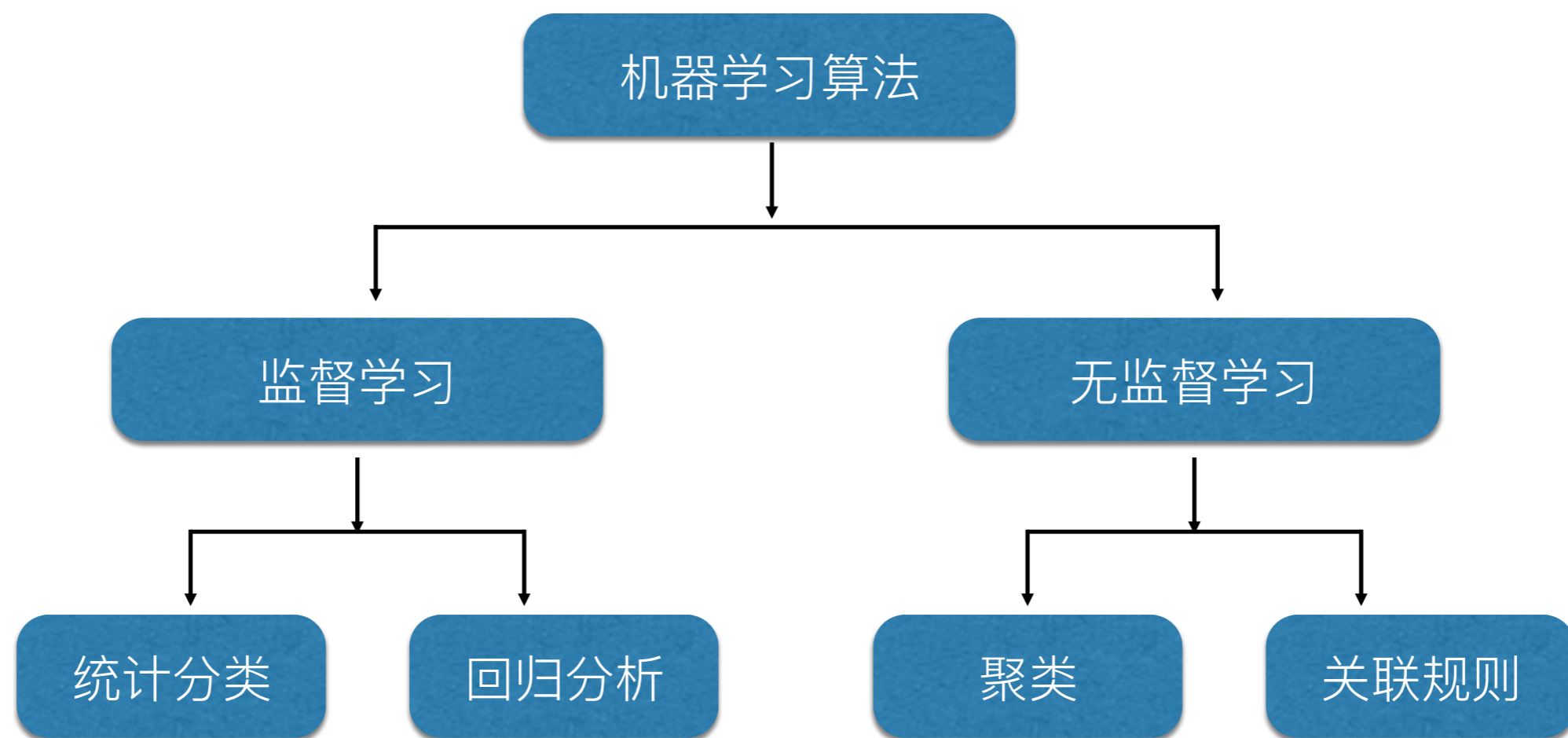
模型



机器学习计算平台

通过数据预处理，深度学习算法，提供预训练模型和高效分布式算法平台生成的定制化模型。

机器学习的算法



用已知结果的数据做训练

不需要已知标签

机器学习的算法

- 决策树
- 随机森林算法
- 逻辑回归
- SVM
- 朴素贝叶斯
- K最近邻算法
- K均值算法
- Adaboost 算法
- 神经网络
- 马尔可夫

深度学习主流框架



TensorFlow 是一个比较低层的库，支持 Python 和 C++，也允许在 CPU 和 GPU 上的计算分布，甚至支持使用 gRPC 进行水平扩展。TensorFlow 是一个非常好的框架，但是却非常低层。使用 TensorFlow 需要编写大量的代码，你必须一遍又一遍地重新发明轮子。

theano

Theano 是一个比较低层的库，也因此它并不适合深度学习，而更适合数值计算优化。它支持自动的函数梯度计算，带有 Python 接口并集成了 Numpy。Theano 不支持多 GPU 和水平扩展



Keras 是一个非常高层的库，可以工作在 Theano 和 TensorFlow（可以配置）之上，只需几行代码就能构建一个神经网络。适合初学者进行实验和测试。



Torch 是 Facebook 的人工智能研究所用的框架，编程语言是 Lua，Torch 的核心是流行的神经网络，它使用简单的优化库，同时具有最大的灵活性，实现复杂的神经网络的拓扑结构。你可以建立神经网络和并行意图，通过CPU和GPU等有效方式。

Caffe

Caffe 非常稳健非常快速，适合生产的计算机视觉系统。缺点是不够灵活，文档非常贫乏，难安装，需要解决大量的依赖包。



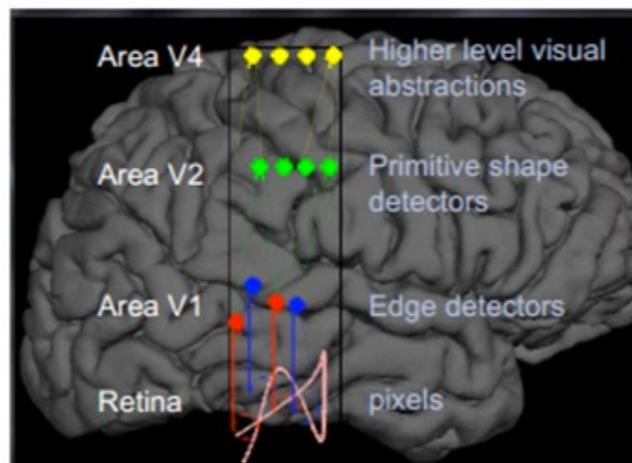
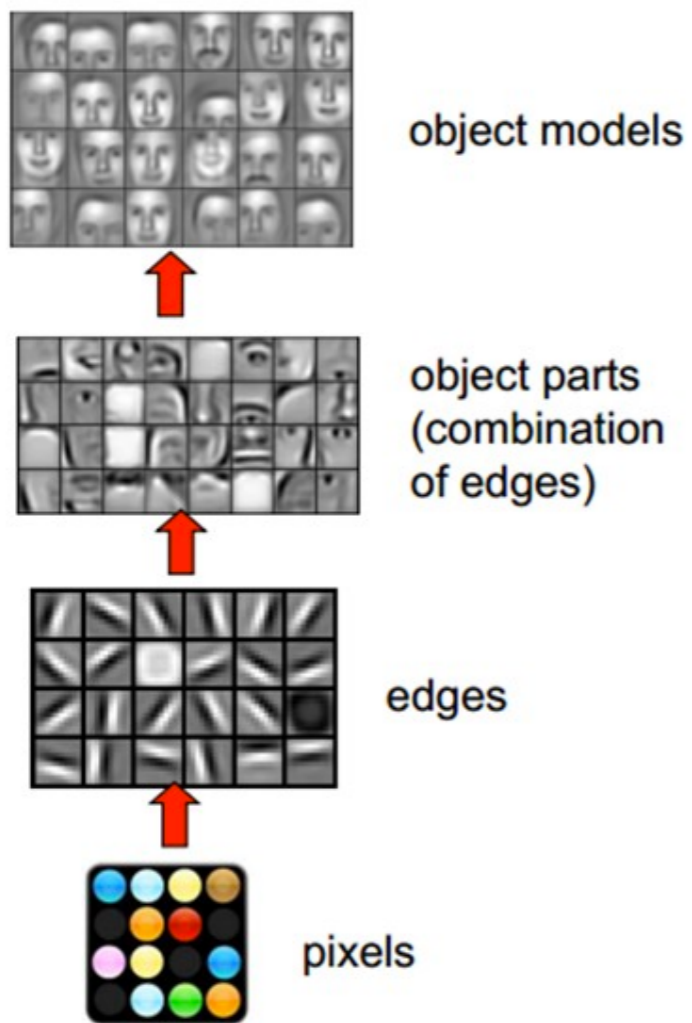
mxnet 支持的言语包括 Python, R, C++, Julia 等。亚马逊 AWS 宣布选择 mxnet 作为其深度学习

AMI 的库，并宣称其巨大的横向扩展能力。



- 机器学习概念
- 机器学习平台
- **人脸识别的实践**

人的视觉系统

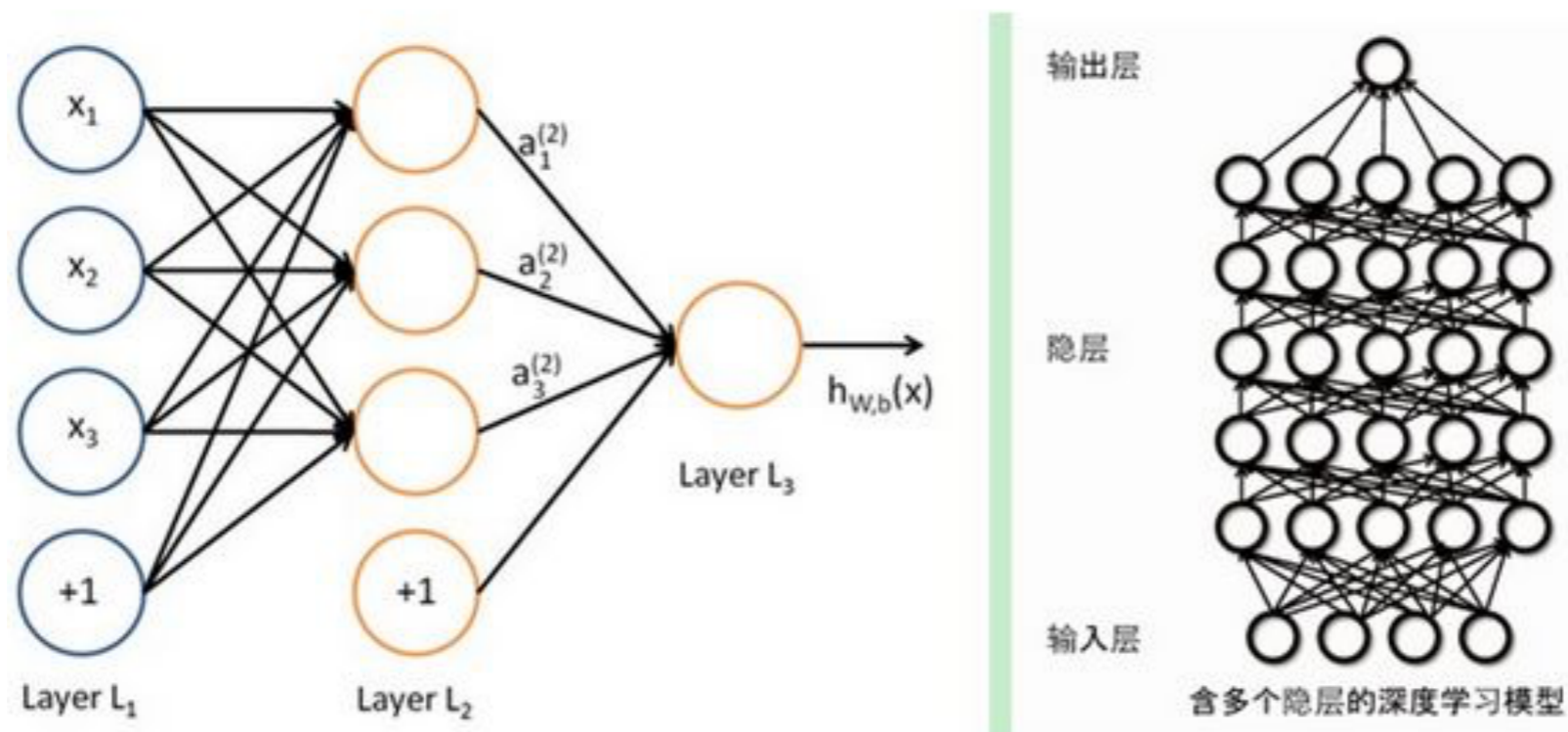


人脑中有150多亿个神经元

- 分级处理
- 高层的特征是低层特征的组合
- 层面越高越精确

神经网络

- 神经网络是一种模拟大脑的算法，是一类模式匹配算法。通常用于解决分类和回归问题。
- 是由具有适应性的简单单元组成的广泛并行互连的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应。
- 神经网络是机器学习的一个庞大的分支，有几百种不同的算法。
- 深度学习就是其中的一类算法，深度学习试图建立大得多也复杂得多的神经网络。



Google Tensorflow 游乐场

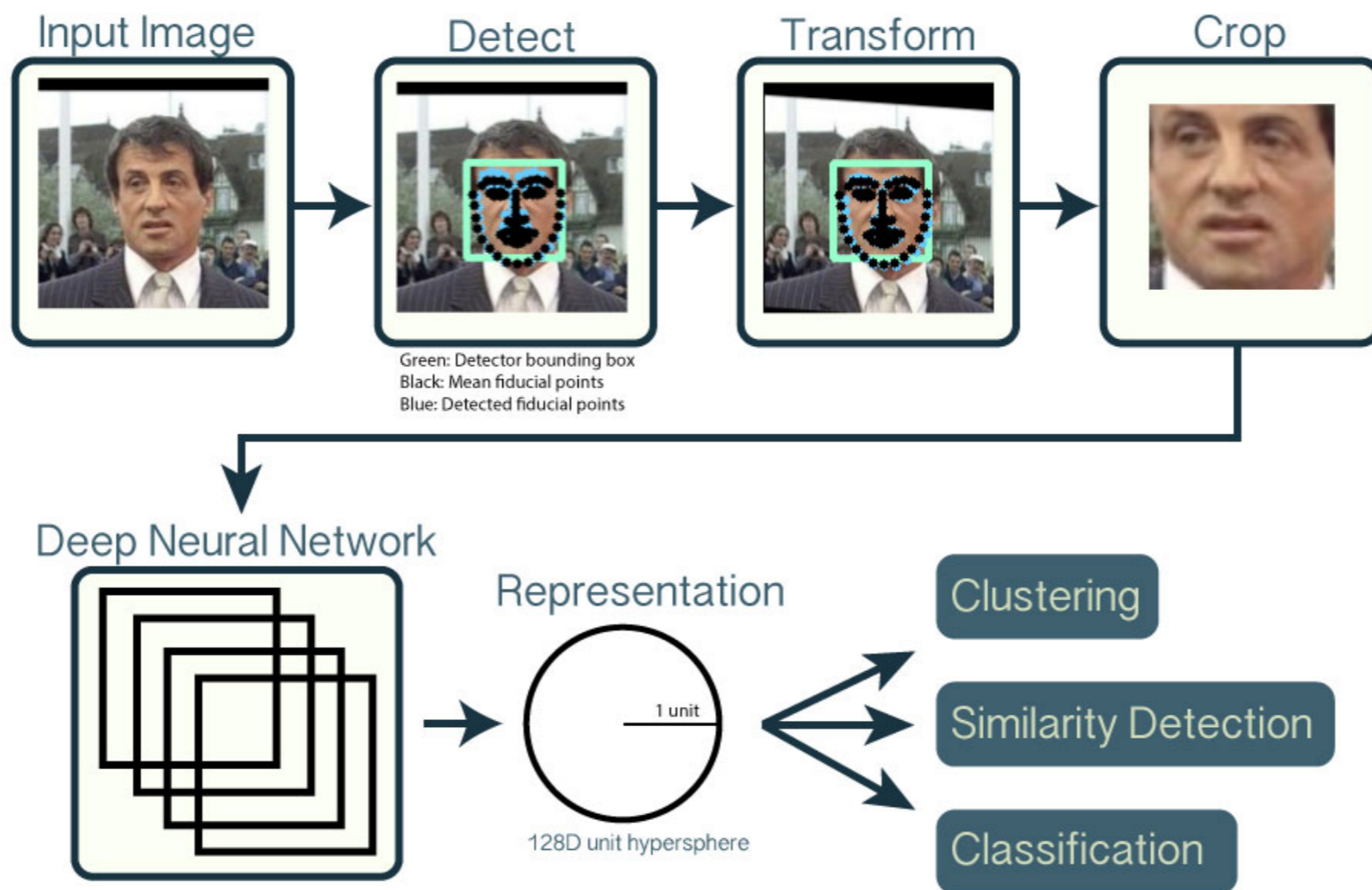
可以在浏览器中训练自己的神经网络，通过图像更直观地了解神经网络的工作原理：



<http://playground.tensorflow.org/>

开源项目 Openface 人脸识别实践

Openface 是用 Torch 和 Python 实现的一个基于深度神经网络的开源人脸识别系统。



<https://cmusatyalab.github.io/openface/>

四步完成人脸识别

1. 预处理：输入训练数据，裁剪脸部

```
./util/align-dlib.py ./training_images/ align outerEyesAndNose ./aligned-images/ --size 96
```

2. 特征提取：对预处理的数据提取特征数据

```
./batch-represent/main.lua -outDir ./generated-embeddings/ -data ./aligned-images/
```

3. 模型训练：根据特征生成分类器

```
./demos/classifier.py train ./generated-embeddings/
```

4. 模型预估：使用分类器对新数据做评估

```
./demos/classifier.py infer ./generated-embeddings/classifier.pkl test_image.jpg
```

使用 docker 使用 Openface

```
[localhost:tmp chenaizhen$ docker search openface
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
bamos/openface	OpenFace	28		[OK]
harshjv/openface	Face recognition with Google's FaceNet dee...	4		[OK]
elihar/openface	Modified version of bamos/openface	1		
nightseas/openface	CMU openface with CUDA.	1		[OK]
lilac/openface		0		
wzdf1982/openface	openface	0		[OK]
uoacer/openface-mass-compare	An openface script that runs a REST server	0		[OK]
askaditya619/openface	OpenFace tools and run over custom images	0		
idinteraction/openface		0		

```
root@37538e999c61:/# cd root
root@37538e999c61:~# cd openface/
root@37538e999c61:~/openface# ls
CONTRIBUTING.md  LICENSE  api-docs  build  data  docs  images  models  openface  run-tests.sh  tests  util
Dockerfile      README.md  batch-represent  cloc.sh  demos  evaluation  mkdocs.yml  opencv-dlib-torch.Dockerfile  requirements.txt  setup.py  training
```

1. 预处理：输入训练数据，裁剪脸部

```
[root@93aed3b1a312:~/openface/ML# /root/openface/util/align-dlib.py /root/openface/ML/training_images/ align outerEyesAndNose /root/openface/ML/aligned-images/ --size 96
=== /root/openface/ML/training_images/Pengmama/Pengmama20.jpg ===
=== /root/openface/ML/training_images/Pengmama/Pengmama18.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye15.jpg ===
=== /root/openface/ML/training_images/Pengmama/Pengmama8.jpg ===
=== /root/openface/ML/training_images/Pengmama/Pengmama3.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye1.jpg ===
=== /root/openface/ML/training_images/Trump/trump13.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye6.jpg ===
=== /root/openface/ML/training_images/Trump/trump2.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye10.jpg ===
=== /root/openface/ML/training_images/Hillary/Hillary19.jpg ===
=== /root/openface/ML/training_images/Mark/Mark10.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye14.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye4.jpg ===
=== /root/openface/ML/training_images/Pengmama/Pengmama19.jpg ===
=== /root/openface/ML/training_images/Obama/obama14.jpg ===
=== /root/openface/ML/training_images/Trump/trump14.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye5.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye12.jpg ===
=== /root/openface/ML/training_images/Jay/Jay12.jpg ===
=== /root/openface/ML/training_images/Obama/obama10.jpg ===
=== /root/openface/ML/training_images/Xidada/Xidada3.jpg ===
=== /root/openface/ML/training_images/Mark/Mark20.jpg ===
=== /root/openface/ML/training_images/Hillary/Hillary18.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye13.jpg ===
=== /root/openface/ML/training_images/Xidada/Xidada15.jpg ===
=== /root/openface/ML/training_images/Mark/Mark12.jpg ===
=== /root/openface/ML/training_images/Hillary/Hillary8.jpg ===
=== /root/openface/ML/training_images/Xidada/Xidada9.jpg ===
=== /root/openface/ML/training_images/Jay/Jay1.jpg ===
=== /root/openface/ML/training_images/Xidada/Xidada4.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye18.jpg ===
=== /root/openface/ML/training_images/Maoyeye/Maoyeye11.jpg ===
=== /root/openface/ML/training_images/Mark/Mark14.jpg ===
=== /root/openface/ML/training_images/Trump/trump6.jpg ===
=== /root/openface/ML/training_images/Obama/obama4.png ===
=== /root/openface/ML/training_images/Hillary/Hillary16.jpg ===
=== /root/openface/ML/training_images/Xidada/Xidada19.jpg ===
=== /root/openface/ML/training_images/Xidada/Xidada12.jpg ===
=== /root/openface/ML/training_images/Trump/trump18.jpg ===
```



Hillary1



Hillary2



Hillary3



Hillary4



Hillary5



Hillary6



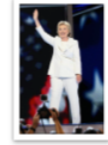
Hillary7



Hillary8



Hillary9



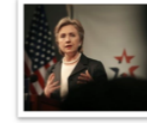
Hillary10



Hillary11



Hillary12



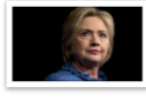
Hillary13



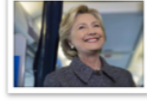
Hillary14



Hillary15



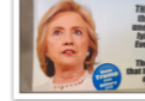
Hillary16



Hillary17



Hillary18



Hillary19



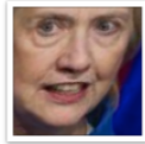
Hillary20



Hillary21



Hillary1.png



Hillary2.png



Hillary3.png



Hillary4.png



Hillary5.png



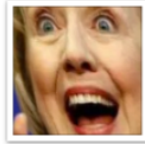
Hillary6.png



Hillary7.png



Hillary8.png



Hillary9.png



Hillary10.png



Hillary11.png



Hillary12.png



Hillary13.png



Hillary14.png



Hillary15.png



Hillary16.png



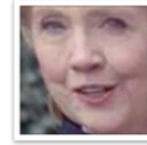
Hillary17.png



Hillary18.png



Hillary19.png



Hillary20.png



Hillary21.png

2.特征提取：对预处理的数据提取特征数据

```
[root@93aed3b1a312:~/openface/ML# /root/openface/batch-represent/main.lua -outDir /root/openface/ML/generated-embeddings/ -data /root/openface/ML/aligned-images/
{
  data : "/root/openface/ML/aligned-images/"
  imgDim : 96
  model : "/root/openface/models/openface/nn4.small2.v1.t7"
  device : 1
  outDir : "/root/openface/ML/generated-embeddings/"
  cache : false
  cuda : false
  batchSize : 50
}
/root/openface/ML/aligned-images/
cache location: /root/openface/ML/aligned-images/cache.t7
Creating metadata for cache.
{
  sampleSize :
  {
    1 : 3
    2 : 96
    3 : 96
  }
  split : 0
  verbose : true
  paths :
  {
    1 : "/root/openface/ML/aligned-images/"
  }
  samplingMode : "balanced"
  loadSize :
  {
    1 : 3
    2 : 96
    3 : 96
  }
}
running "find" on each class directory, and concatenate all those filenames into a single file containing all image paths for a given class
now combine all the files to a single large file
load the large concatenated list of sample paths to self.imagePath
154 samples found..... 0/154 .....] ETA: 0ms | Step: 0ms
Updating classList and imageClass appropriately
[===== 8/8 =====>] Tot: 204ms | Step: 25ms
Cleaning up temporary files
Splitting training and test sets to a ratio of 0/100
nImgs: 154
Represent: 50/154
Represent: 100/154
Represent: 150/154
```


3. 模型训练：根据特征生成分类器

```
[root@93aed3b1a312:~/openface/ML# /root/openface/demos/classifier.py train /root/openface/ML/generated-embeddings/
/usr/local/lib/python2.7/dist-packages/sklearn/lda.py:4: DeprecationWarning: lda.LDA has been moved to discriminant_analysis.LinearDiscriminantAnalysis in 0.17 and will be removed i
n 0.19
  "in 0.17 and will be removed in 0.19", DeprecationWarning)
Loading embeddings.
Training for 8 classes.
Saving classifier to '/root/openface/ML/generated-embeddings//classifier.pkl'
```

4. 模型预估：使用分类器对新数据做评估

```
[root@93aed3b1a312:~/openface/ML# /root/openface/demos/classifier.py infer /root/openface/ML/generated-embeddings/classifier.pkl /root/openface/ML/test_images/obama_test.jpg
/usr/local/lib/python2.7/dist-packages/sklearn/lda.py:4: DeprecationWarning: lda.LDA has been moved to discriminant_analysis.LinearDiscriminantAnalysis in 0.17 and will be removed i
n 0.19
  "in 0.17 and will be removed in 0.19", DeprecationWarning)

=== /root/openface/ML/test_images/obama_test.jpg ===
Predict Obama with 0.66 confidence.
```

在线预测系统

基于七牛自定义数据处理平台搭建一个简单的深度学习在线预测系统，帮助提供基本的在线预测服务。



```
{  
  who: "Trump",  
  confidence: 0.87  
}
```

Demo 演示