



TensorFlow简单应用

泰凡科技
徐大为

个人介绍

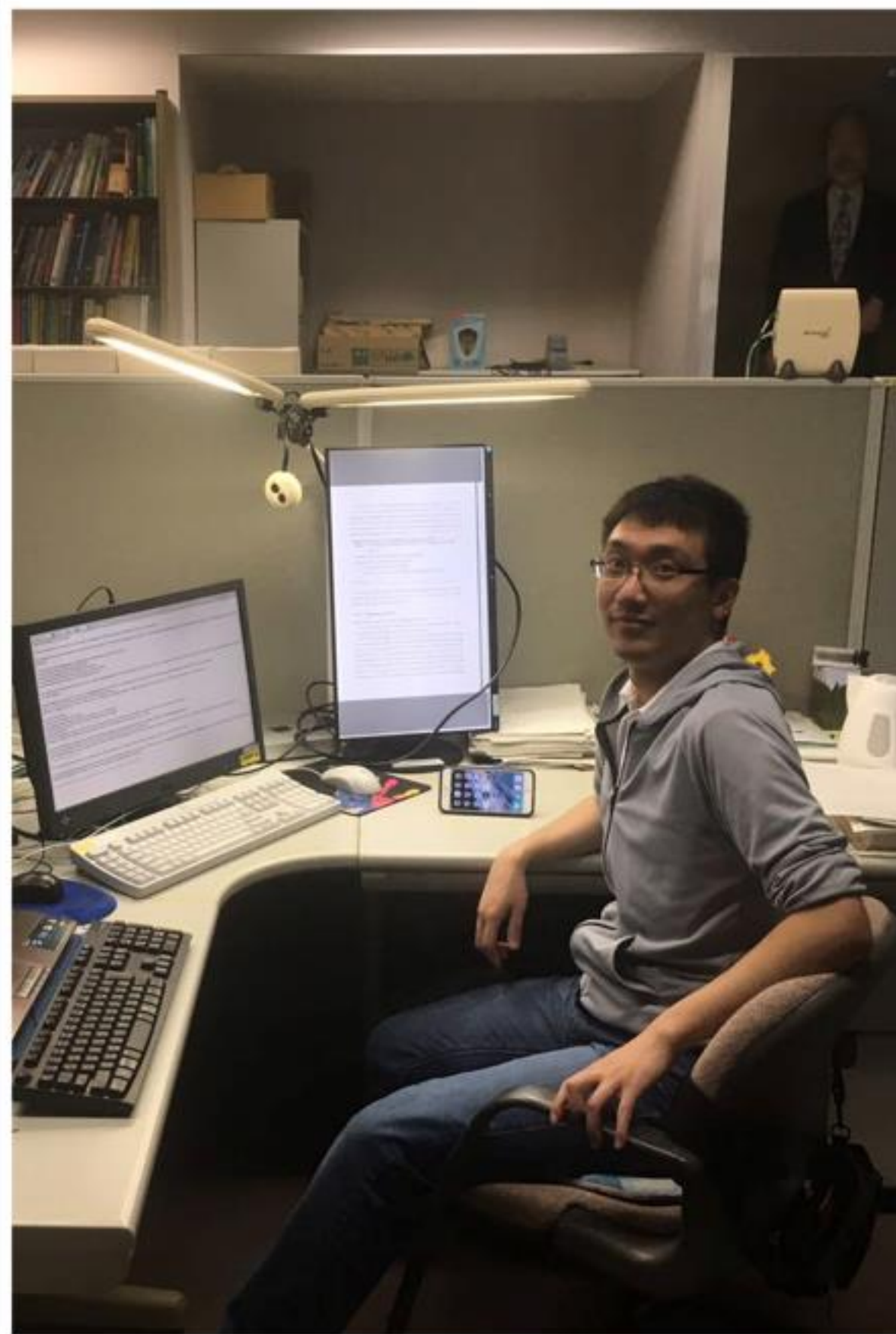
徐大为

2008~2012 天大软件工程本科

2012~2014 天大&JAIST硕士

2014~2017 JAIST 博士

2017~至今 天津泰凡科技



使用LSTM进行数据预测

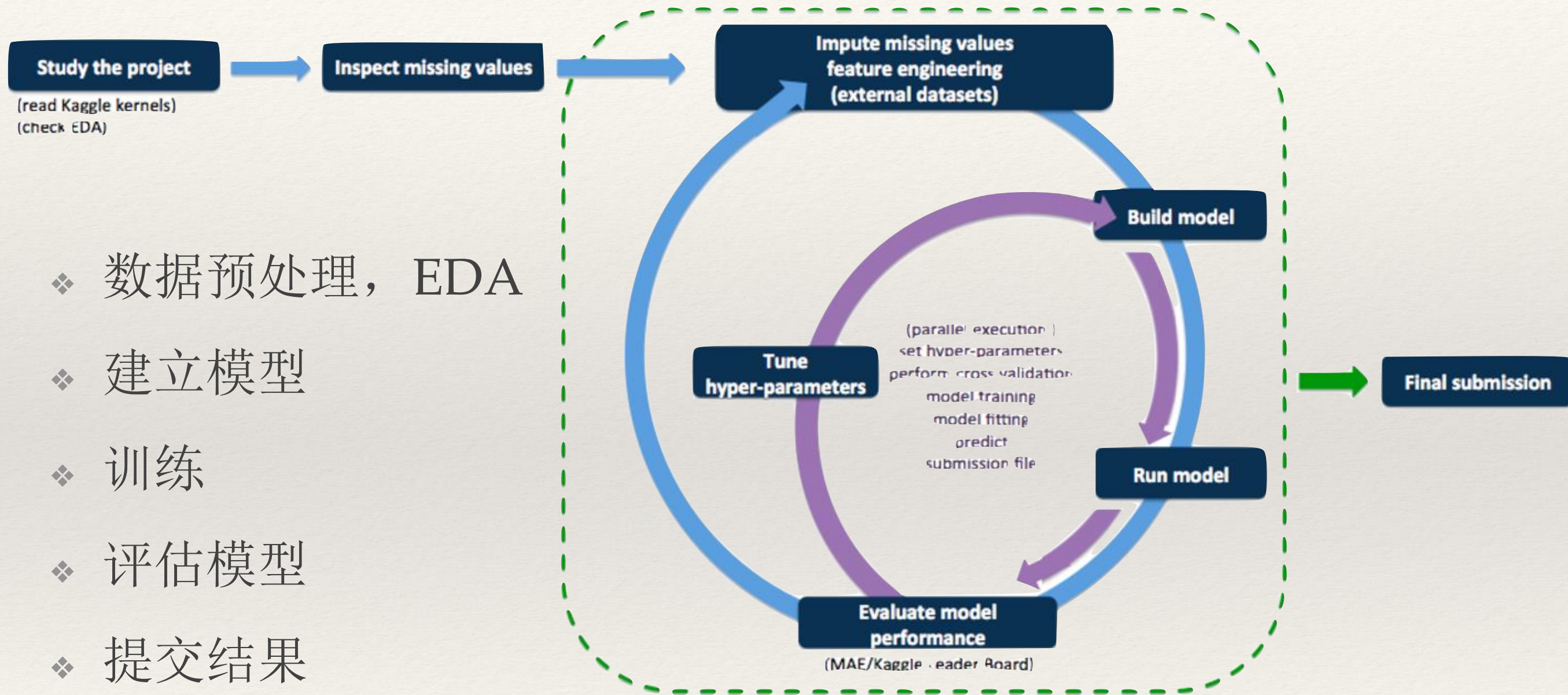
乘用车零售量预测

首页>天池大赛>[印象盐城]数创未来大数据竞赛 - 乘用车零售量预测

	状态	举办方	赛季2	奖金	参赛队伍
[印象盐城]数创未来大数据竞赛 - 乘用车零...	已结束	 	2018-03-15	¥ 460000	2500

- ❖ 对于政府、行业监管部门以及汽车企业需要有更细化粒度的销量预测解决方案。
- ❖ 提供2012年1月-2017年10月盐城分车型销量配置数据
- ❖ 预测2017年11月，12月盐城分车型销量数据。

工作流程



数据特征

- ❖ 一共32维特征,1个float,21个int以及10个object,不存在缺失值
- ❖ 20种类别特征， 10种度量特征， 最后的销量作为Label
- ❖ 评价指标：

$$\varepsilon = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - a_i)^2}$$

其中：

- (1) ε ：参赛者最终得分
- (2) n ：测试样本总量
- (3) p_i ：第*i*个测试样本的预测值
- (4) a_i ：第*i*个测试样本的真实值

字段	含义	示例
sale_date	销售日期	201201
province_id	销售省份ID	345
city_id	销售城市ID	3456
class_id	车型ID	234567
sale_quantity	销量	15
brand_id	品牌ID	234
compartment	厢数	3
type_id	车型类别ID	1
level_id	车型级别ID	1
department_id	车型系别ID	1
TR	变速器档位	6
gearbox_type	变速器形式	AT
displacement	排量	2.5
...

时间序列数据

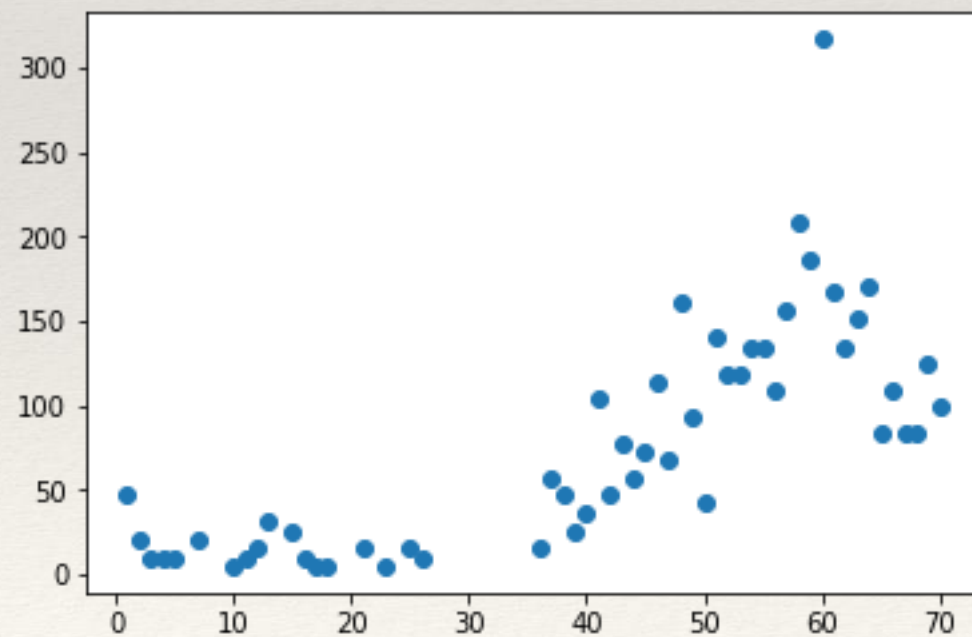
已知 $t-1$, $t-2$时刻城市某车型的销量, 预测 t 时刻的销量

❖ 经典的时序问题, 三点特征:

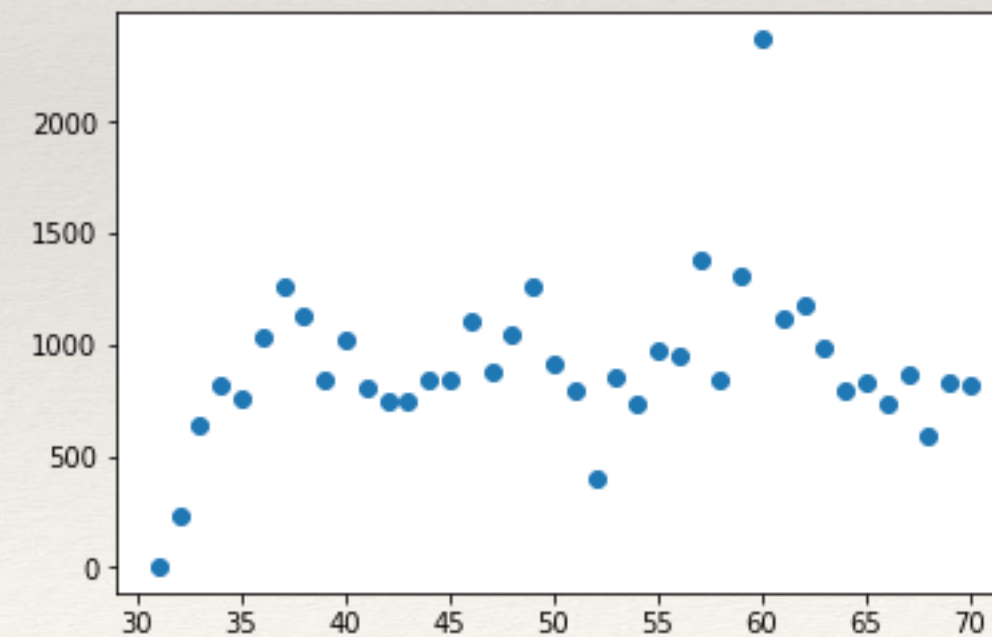
- ❖ 趋势性 (总体升/降)
- ❖ 周期性 (去年同一时间)
- ❖ 平滑性 (临近月份)

❖ EDA

- ❖ 销量随时间的变化等



class_id: 198427



class_id: 739296

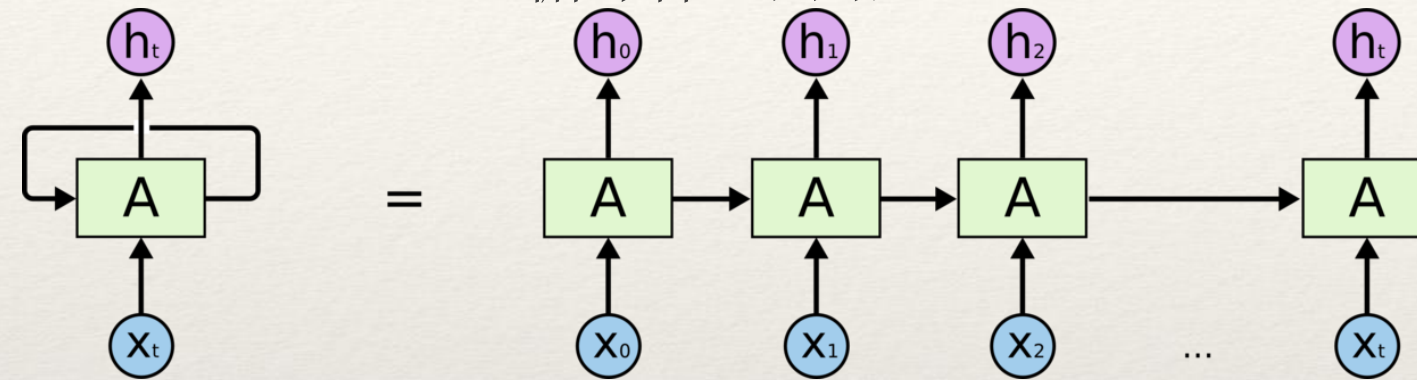


整体销量趋势与周期

可以发现，总销量随时间变化的趋势是走高的，但是会有周期性的峰值。

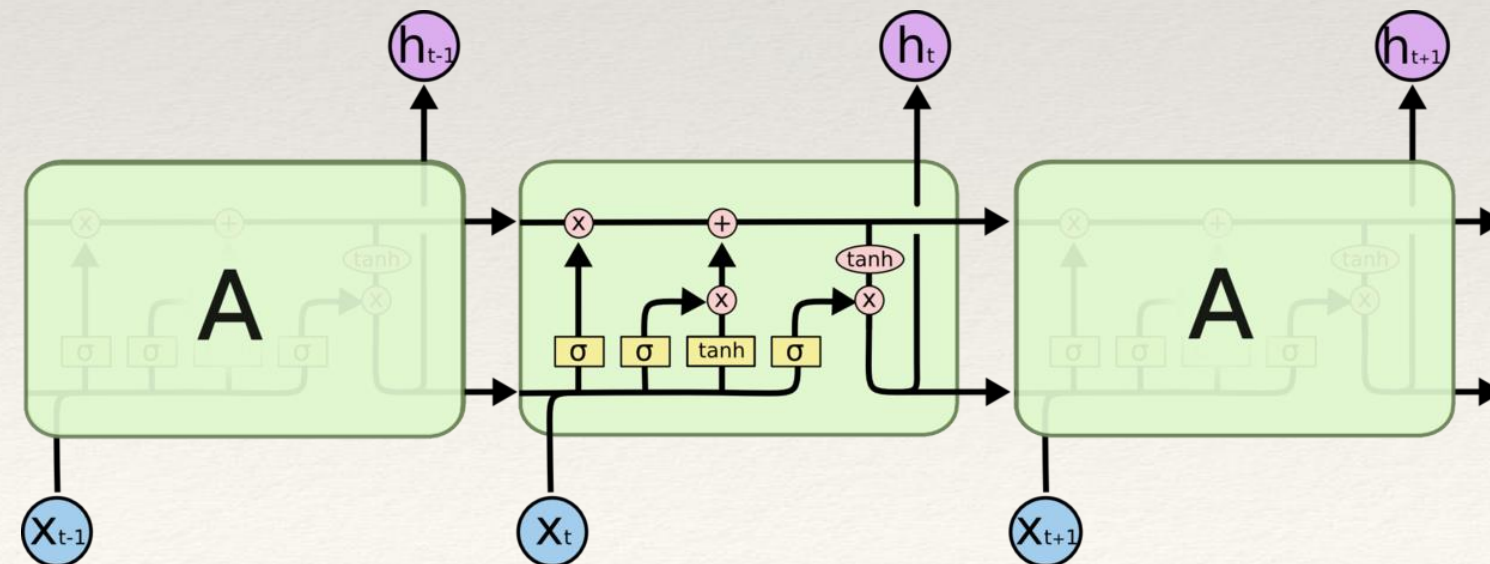
循环神经网络

- ❖ Recurrent Neural Networks 循环神经网络



- ❖ LSTM

- ❖ LSTM 是一种RNN，与最基本的RNN相比，它多了四个门(Gate)，它可以记忆不定时间长度的数值，区块中有一个gate能够决定input是否重要到能被记住及能不能被output。



过程

训练参数

duration = 20 #每次训练使用的
序列长度

n_hidden = 64 #lstm神经元个数

n_layer = 2 #lstm层数

test_datas = 2 #预测数据点个数

```
test_holder = tf.placeholder(tf.float32,[1,1,666]) #测试数据输入的  
place holder
```

```
output,_,_ = net(input_data,n_hidden,n_layer)#通过模型得到输出
```

```
test_output,test_istate,test_lstate =
```

```
net(test_holder,n_hidden,n_layer,True) #通过模型得到测试输出  
， 测试初始状态， 测试最终状态
```

```
loss_op =
```

```
tf.sqrt(tf.reduce_mean(tf.squared_difference(output,input_label)))  
#定义损失函数， 是输出和输入的差方
```

```
train_op = tf.train.AdamOptimizer(0.0001).minimize(loss_op) #  
定义训练， 使用Adam优化器
```

```
sess.run(tf.global_variables_initializer()) #初始化所有变量
```

```
tf.train.start_queue_runners(sess=sess) #开始运行输入队列
```

```
for step in range(1000000): #一个比较大的训练循环次数
```

```
    _loss_value = sess.run([train_op,loss_op])
```


总结

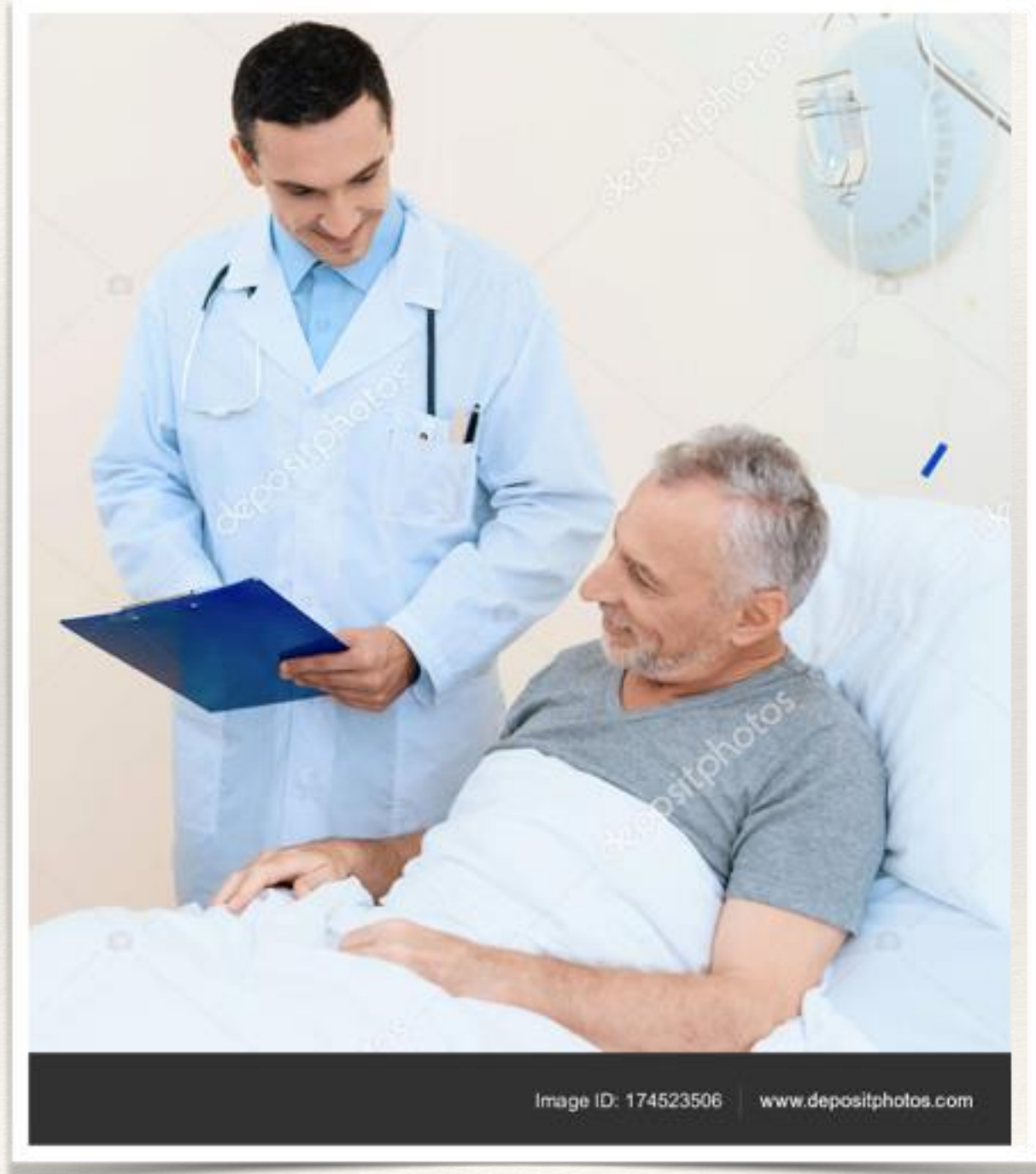
- ❖ 如果加入人工设计特征，会让序列数据更易于分析
- ❖ 对数据的可视化分析对于有效特征构建也很有帮助
- ❖ 对于序列问题，有这多种解决方法。LSTM只是其中之一。



使用Faster-RCNN进行图像识别

应用：离床检测

❖ 为了观测老人的活动状况，比如夜间或午休时间老人独自离床，照护人员需要经常巡视老人的房间和活动地点，以免发生意外情况。这种状况下，给养老机构带来了很大的负担，即使加强巡护也不能完全保证在出现意外的情况下，能够立即察觉。

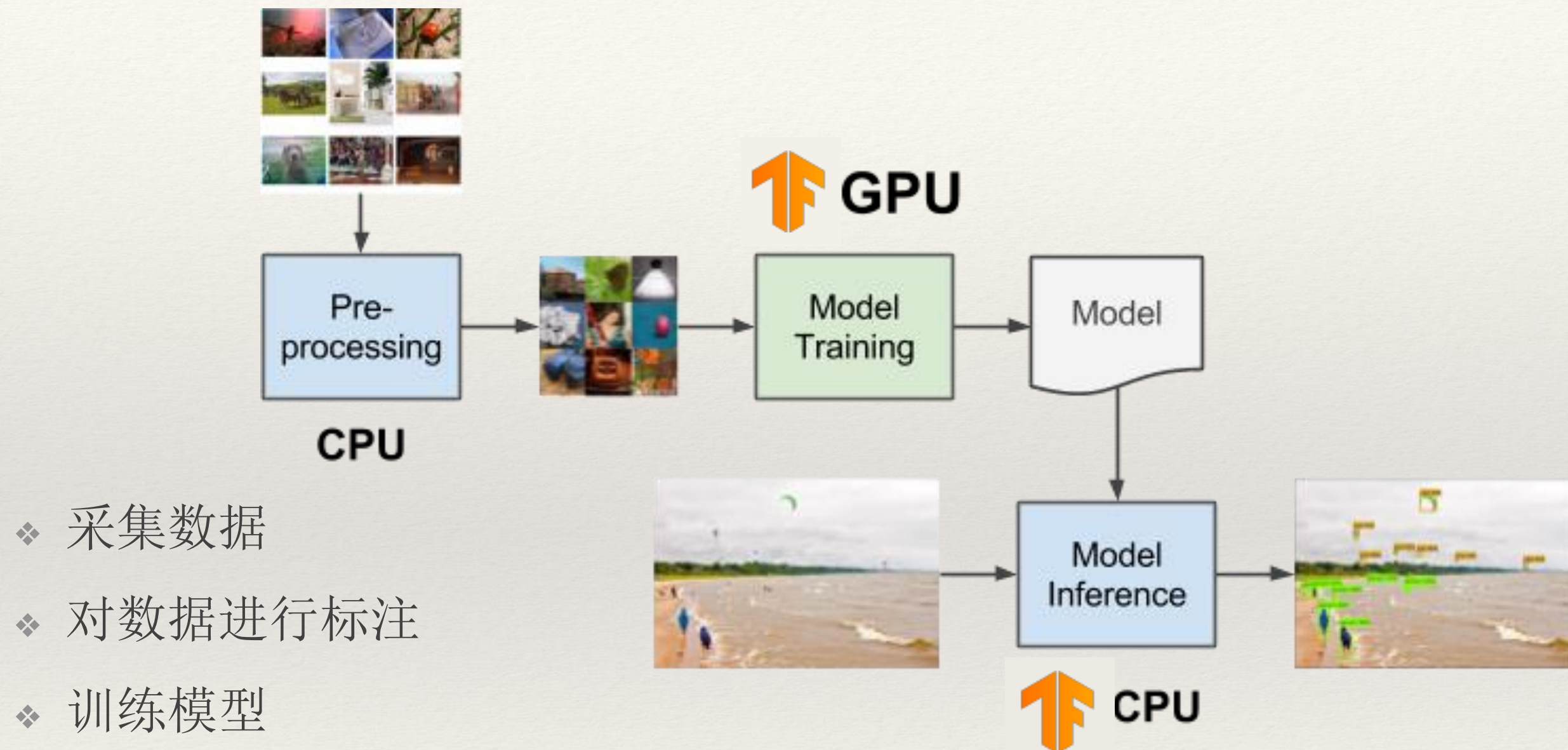




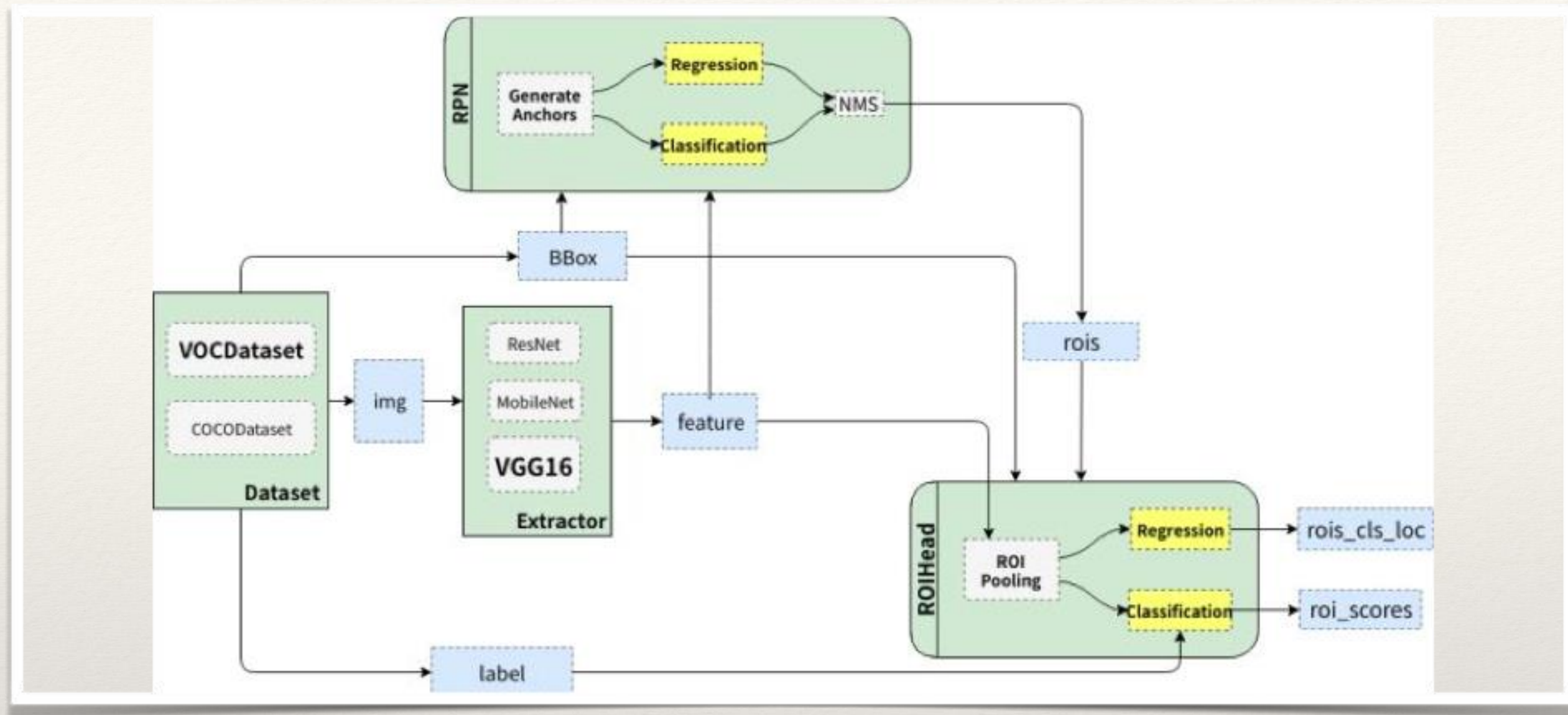
应用：离床检测

若老人在预警时间段离床，程序则会自动通知云端服务器，服务器会推送预警消息给对应的照护端平台，照护人员可及时发现老人的离床行为，避免发生危险情况。

工作流程



- ❖ 采集数据
- ❖ 对数据进行标注
- ❖ 训练模型
- ❖ 对测试数据应用模型



Faster-RCNN的Tensorflow实现

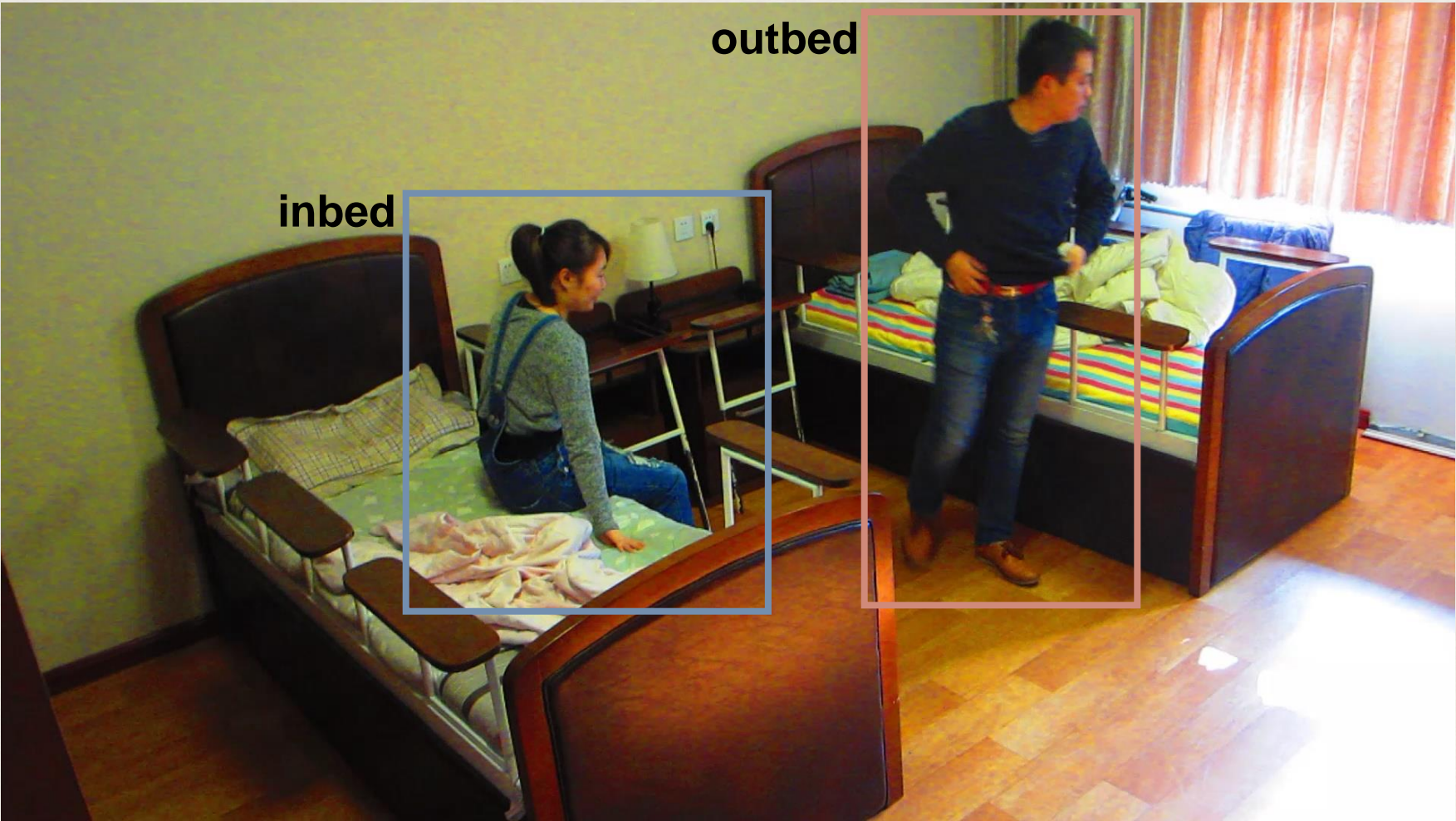
目标检测、定位、识别

在目标检测领域, Faster R-CNN是许多目标检测算法的基础

*Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
by Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun.

建立并标注数据集

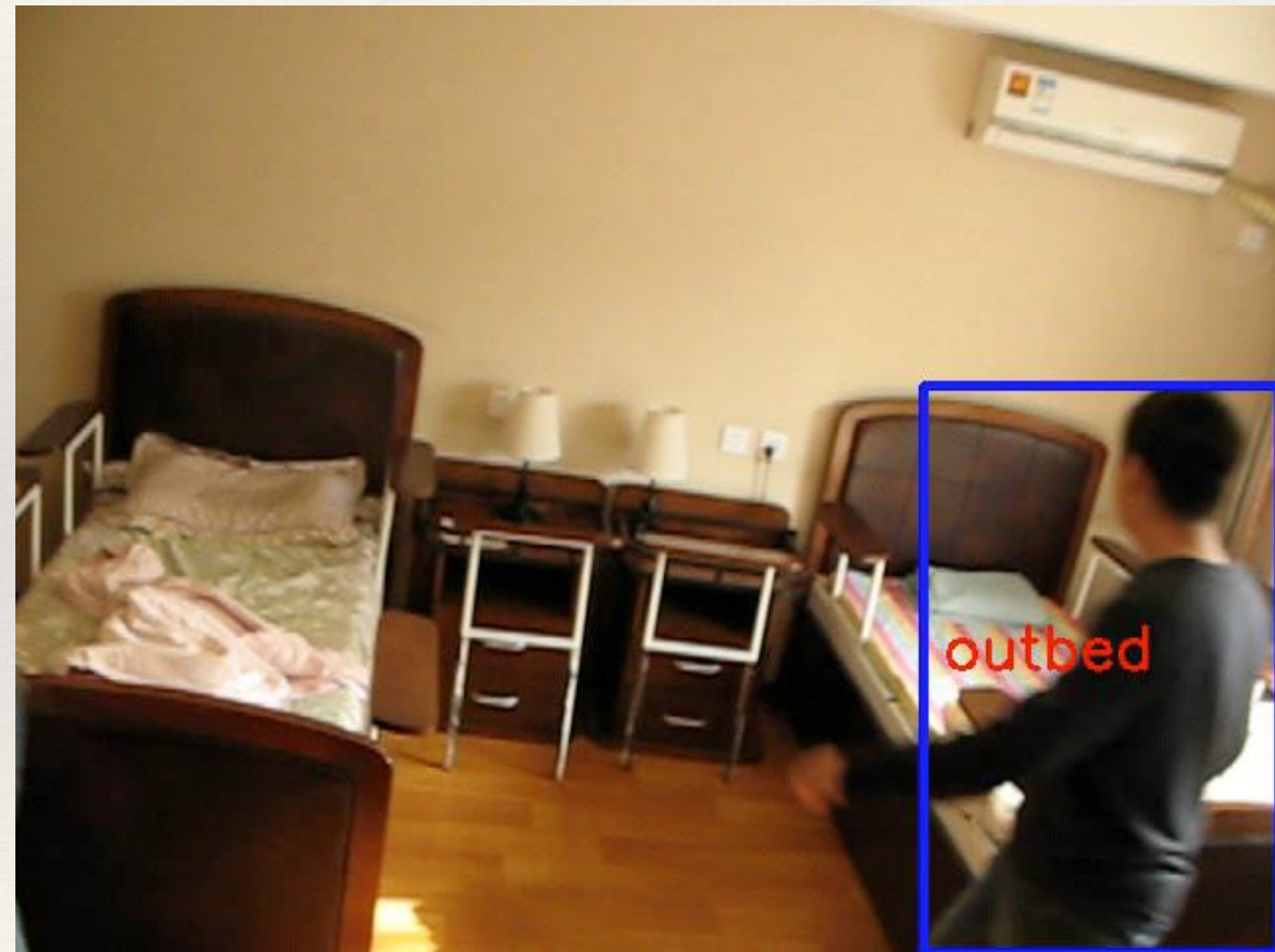
- ❖ 训练数据集基于采集的约2700张动作数据，按照VOC格式手动标注Ground Truth Bounding Box:
- ❖ 对于每个人物，要确定该人物边界框的左上角和右下角坐标。



```
<annotation>
...
<object>
  <name>outbed</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  <bndbox>
    <xmin>1222</xmin>
    <ymin>17</ymin>
    <xmax>1457</xmax>
    <ymax>719</ymax>
  </bndbox>
</object>
<object>
  <name>inbed</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  <bndbox>
    <xmin>113</xmin>
    <ymin>280</ymin>
    <xmax>1185</xmax>
    <ymax>1075</ymax>
  </bndbox>
</object>
</annotation>
```


训练和测试结果

- ❖ Faster R-CNN的训练，是在已经训练好的model(VGG16)的基础上继续进行训练。
- ❖ 更多时间花在调试上。
- ❖ 右边是训练70万次的效果
- ❖ 使用的硬件设备：
 - ❖ Intel® Core™ i7-6700K CPU @ 4.00GHz × 8
 - ❖ GeForce GTX 1080Ti
 - ❖ 16G RAM



Thank you for watching!