第 8 章　深度学习应用—人脸识别

教材： 王万良《人工智能导论》（第4版）

https://www.icourse163.org/course/ZJUT-1002694018

社区资源： https://github.com/Microsoft/ai-edu

参考MOOC： 人工智能：模型与算法（浙大吴飞）

提升卷积神经网络的技巧

- 1）数据增强；
- 2）图像预处理；
- 3）网络的初始化；
- 4）训练期间的小技巧；
- 5）激活函数的选择；
- 6）正则化策略；
- 7）从图中判断模型性能

提升卷积神经网络的技巧

- 数据增强
  - 深度网络一般需要大量的训练图像数据才能取得一个较好的性能
    - 水平翻转，随机修剪和光照、彩色变换
    - 多种不同处理的组合，例如，同时进行旋转和随机缩放
    - 将所有像素的饱和度和值（HSV彩色空间的S和V分量）提高到0.25和4倍之间的倍数（对于一个色块内的所有像素都相同），将这些值乘以系数0.7和1.4，并且向它们添加一个在[-0.1,0.1]范围内的值
    - 利用PCA改变训练图像中RGB通道的强度

提升卷积神经网络的技巧

- 图像预处理
  - 数据以0点为中心，然后做规范化操作，在Python中的操作为下面两行代码：
    - >>> X -= np.mean(X, axis = 0)# zero-center
    - >>> X /= np.std(X, axis = 0)# normalize
  - PCA白化（降低输入的冗余性）
    - 数据如第一种方式一样居中处理，然后计算数据结构中相关的协方差矩阵：
      - >>> X -= np.mean(X, axis = 0) # zero-center
      - >>> cov = np.dot(X.T, X) / X.shape[0] # compute the covariance matrix
    - 将原始（以零为中心）的数据投射到特征基中，将数据解相关
      - >>> U,S,V = np.linalg.svd(cov) # compute the SVD factorization of the data covariance matrix
      - >>> Xrot = np.dot(X, U) # decorrelate the data
    - 白化，它将数据放在特征基中，并通过在每个维度中除以特征值去规范化尺度：
      - >>> Xwhite = Xrot / np.sqrt(S + 1e-5) # divide by the eigenvalues (which are square roots of the singular values)

提升卷积神经网络的技巧

- 网络的初始化
  - 全零初始化
  - 小随机数初始化
    - 0.001*N（0,1），其中N（0,1）是均值为0，标准差为1的高斯分布。也可以使用均匀分布的初始化权值
  - 权值归一化
    - >>> w = np.random.randn(n) /sqrt(n)# calibratingthe variances with 1/sqrt(n)

提升卷积神经网络的技巧

- 训练期间的小技巧
  - 卷积核和池化窗口的大小
    - 在训练期间，输入图像的大小更倾向于2的倍数，例如32,64,224,384或512等
    - 采用小的卷积核（例如3×3）和小步长（例如1）进行填充为0的卷积计算，这不仅减少了参数数量，而且提高了整个深度网络的准确率。
      - 具有步幅1的3×3卷积核，可以保留图像或特征图的空间尺寸。
      - 对于池化层，常用的池化窗口大小为2×2。

提升卷积神经网络的技巧

- 训练期间的小技巧
  - 学习速率
    - 据微型批次（mini-batch-size）的大小来决定
    - 在训练开始时的学习率为0.1，在训练集上损失值不再下降时，然后将学习率除以2（或者5），然后继续进行训练，可能会取得不错的效果
  - 预训练模型的微调
    - 由于预先训练的深度模型良好的泛化能力，可以直接在其他数据集的训练时，采用这些预先训练的模型
    - 数据集的大小，以及它与原始数据集的相似度

提升卷积神经网络的技巧

- 激活函数的选择
  - 作用于卷积层和全连接层之后的非线性计算操作，目前主要的激活函数有tanh、Sigmoid、ReLU、PReLU等函数
    - S型非线性在误差的反向传播中会导致梯度消失的问题
    - 在实践中，tanh非线性函数要优于S型非线性函数
    - ReLU是一个非饱和的激活函数，大大加速了权值梯度下降的收敛速度

过拟合与欠拟合

一个有趣的小故事

一个非洲酋长到伦敦访问，一群记者在机场截住了他。

“早上好，酋长先生”，其中一人问道：“你的路途遥远吗？”

酋长发出了一连串刺耳的声音哄、哼、啊、吱、嘶嘶，然后用纯正的英语说道：是的，非常舒服。

“那么，您准备在这里待多久？”

他又发出了同样的一连串噪音，然后答道：“大约三星期，我想。”

“酋长，告诉我，你是哪里学的这样流利的英语？”迷惑不解的记者问。

又是一阵哄、哼、啊、吱、嘶嘶声，酋长说：“从短波收音机里。”

过拟合与欠拟合

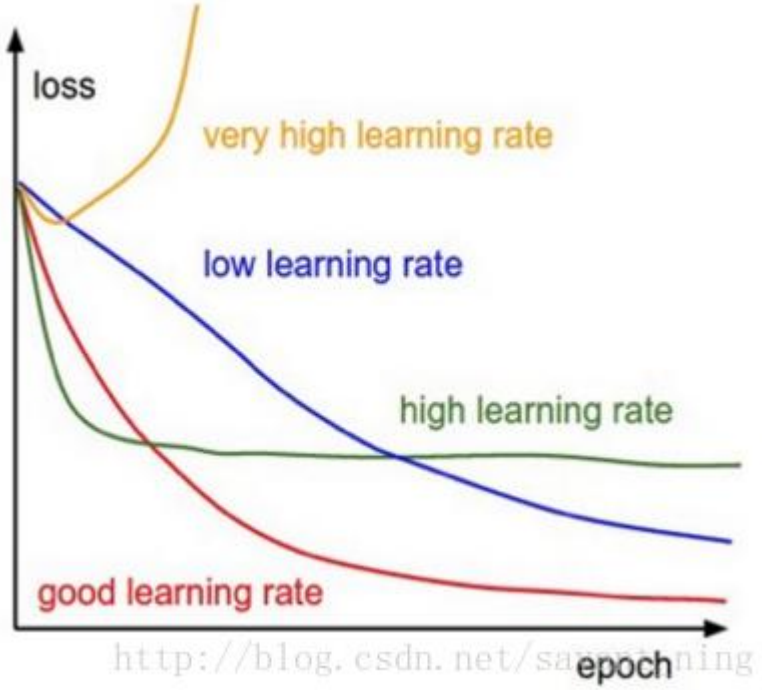
- 欠拟合
  - 原因：一般由于学习器学习能力低下造成
  - 表现：拟合训练集差，预测测试集效果差
  - 解决方案：增加模型的复杂度
- 过拟合
  - 原因：不仅学习了数据集中的有效信息，也学习了噪音数据
  - 表现：拟合训练集好，预测测试集效果差
  - 解决方案：降低参数复杂度、模型复杂度等
- 正则化
  - 正则项：为有效控制模型参数的复杂度，加入参数复杂度的惩罚项，以使得模型选择更加简单的参数

提升卷积神经网络的技巧

- 正则化策略
  - 神经网络防止过拟合的方法
    - L2正则化是正则化最常用的形式之一，可以直接在损失函数中加上带有乘性偏置的所有权值w的平方和来实现。
    - L1正则化是正则化中相对常见的形式，在原始的代价函数后面加上一个L1正则化项，即所有权重w的绝对值的和，乘以λ/n。
    - Dropout是一个很简单的正则化技术
      - 以一定的概率（dropout ratio）将隐层神经元的输入、输出设置为零
    - Batch Normalization
      - 在卷积计算和激活函数中间进行规范化计算，逐层尺度归一
        - 通过对相应的激活区域做规范化操作，使得输出信号各个维度服从均值为0，标准差为1的正态分布
        - 缩放和平移（scale and shift）
          - 规范化计算输出的结果还原最初的输入特征，从而保证网络的容纳能力

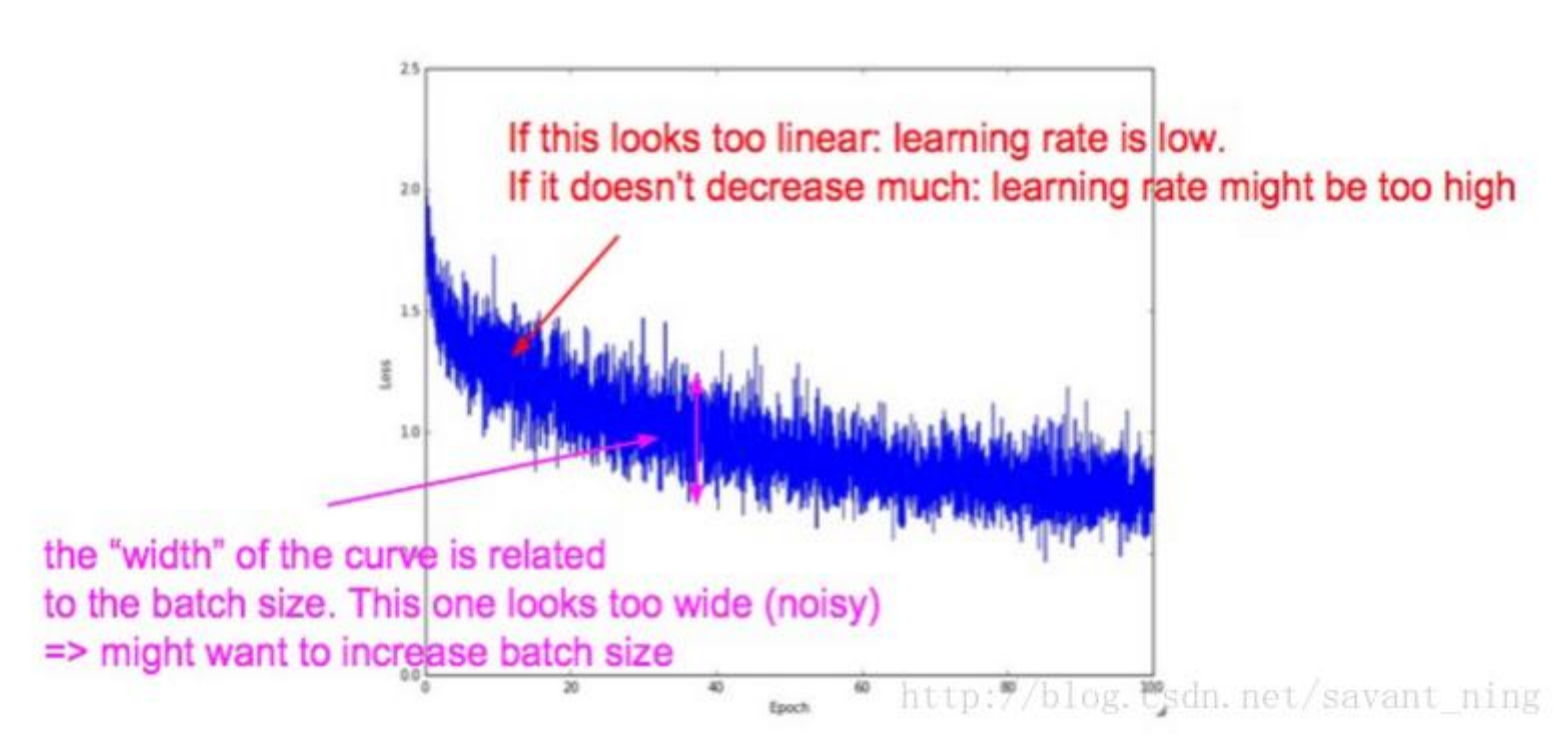
提升卷积神经网络的技巧

- 从图中判断模型性能
  - 学习率是一个非常敏感的参数
    - 非常高的学习率将导致相当奇怪的损失曲线，低学习率将使训练损失减少很慢
    - 良好的学习率，其损耗曲线能够逐渐递减，最终达到网络的最佳性能



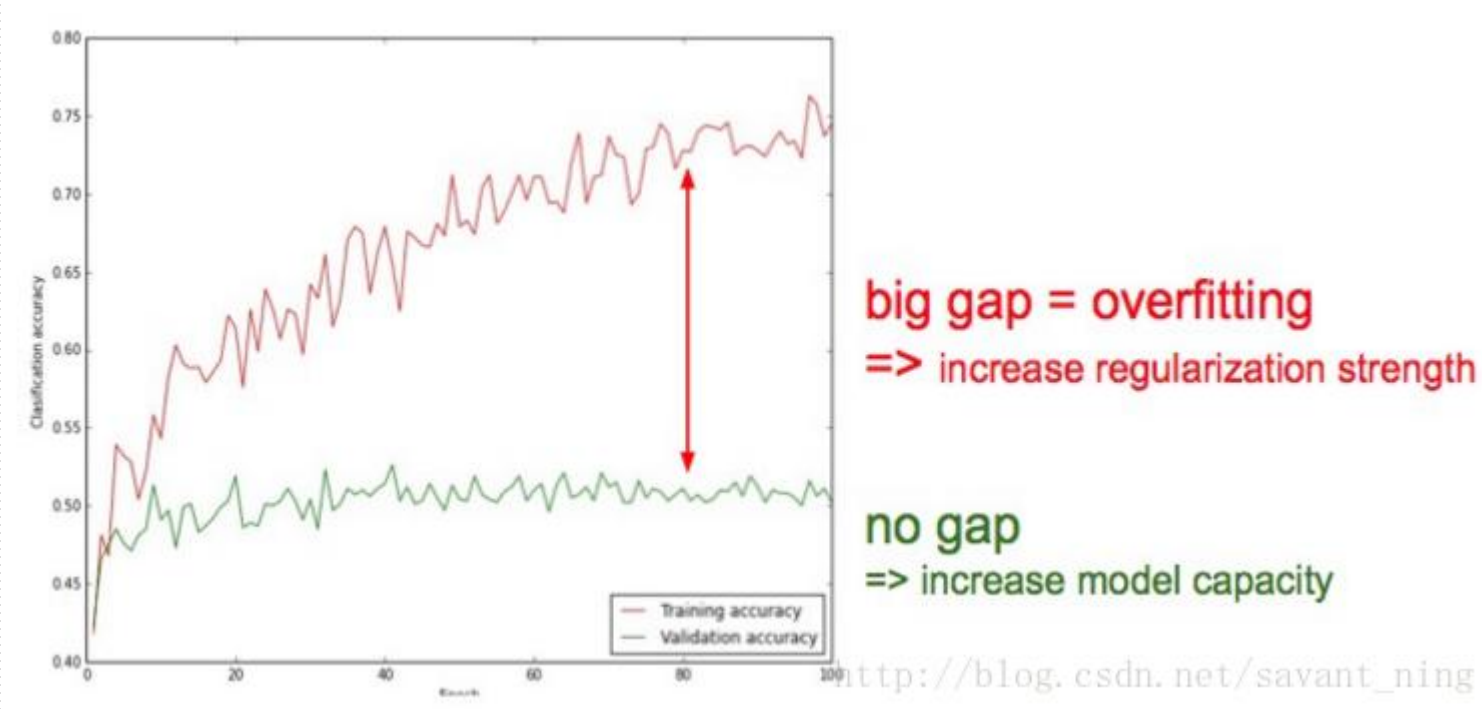
提升卷积神经网络的技巧

- 从图中判断模型性能
  - 损失曲线图



提升卷积神经网络的技巧

- 从图中判断模型性能
  - accuracy曲线图
    - 训练时的accuracy
    - 测试时的accuracy



人脸识别

- 何为人脸识别
  - （狭义）以分析与比较人脸视觉特征信息为手段，进行身份验证或查找的一项计算机视觉技术
  - （广义）在图片或视频流中识别出人脸，并对该人脸图像进行一系列相关操作的技术。
    - 图像采集、人脸检测、人脸定位、人脸提取、人脸预处理、人脸特征提取、人脸特征对比等

人脸识别

- 人脸识别的应用
  - 2016年2月，北京站开启“刷脸”进站模式
  - 手机解锁
    - Face ID

人脸识别

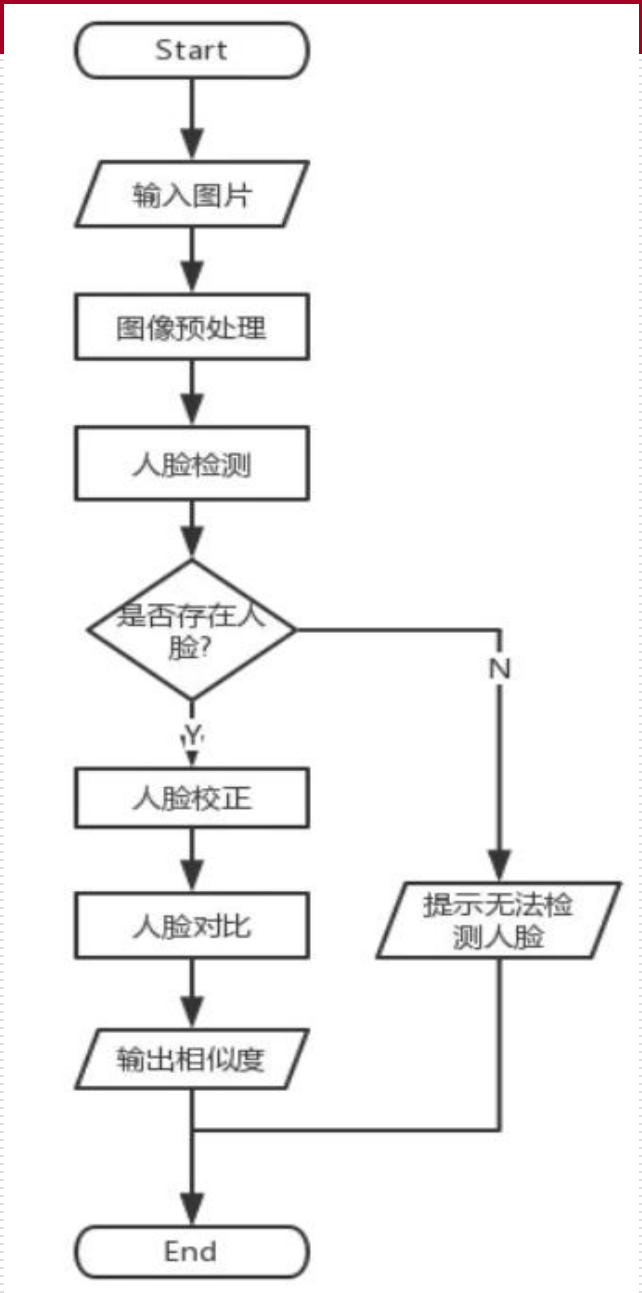
- 人脸识别技术的典型应用场景
  - 身份认证场景
    - 系统判断当前被检测人脸是否已经存在于系统内置的人脸数据库中。如果系统内没有该人的信息，则认证失败
  - 证件验证场景
    - 判断证件中的人脸图像与被识别人的人脸是否相同
    - 引入活体检测技术
  - 人脸检索场景
    - 对人脸图片“一对多”地对比
    - 在重要的交通关卡布置人脸检索探头，将行人的人脸图片在犯罪嫌疑人数据库中进行检索，从而比较高效地识别出犯罪嫌疑人

人脸识别

- 人脸识别技术的典型应用场景
  - 人脸分类场景
    - 社交类App可以通过用户上传的自拍图片来判断该用户的性别、年龄等特征，从而为用户有针对性地推荐一些可能感兴趣的人
  - 交互式应用场景
    - 美颜类自拍软件
  - 其他应用
    - 人脸图片的重建技术可以应用到通信工程领域，实现低比特率的图片与视频传输
    - “视频换脸”功能

人脸识别

- 人脸识别的目标
  - 一种人脸对比解决方案的流程图



人脸对比场景

- 图像预处理
  - 如图片带有噪声，或者图片尺寸不符合系统要求等
  - 带有椒盐噪声的图片和经过中值滤波处理后的图片

人脸对比场景

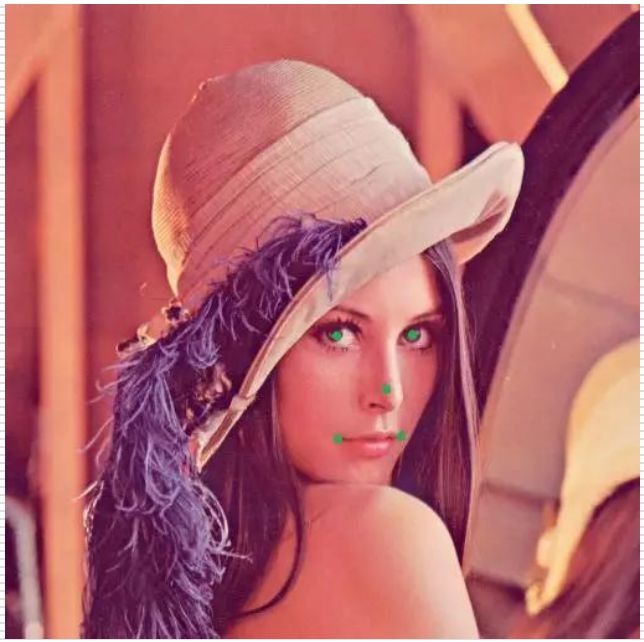
- 人脸检测
  - 用来判断一张图片中是否存在人脸的操作
    - 如果图片中存在人脸，则定位该人脸在图片中的位置；如果图片中不存在人脸，则返回图片中不存在人脸的提示信息

人脸对比场景

- 人脸校正
  - 对图片中人脸图像的一种几何变换，目的是减少倾斜角度等几何因素给系统带来的影响
  - 随着深度学习技术的广泛应用，人脸校正并不是被绝对要求存在于系统中

人脸对比场景

- 人脸特征点定位
  - 在检测到图片中人脸的位置之后，在图片中定位能够代表图片中人脸的关键位置的点。
  - 常用的人脸特征点是由左右眼、左右嘴角、鼻子这5个点组成的5点人脸特征点，以及包括人脸及嘴唇等轮廓构成的68点人脸特征点等



人脸对比场景

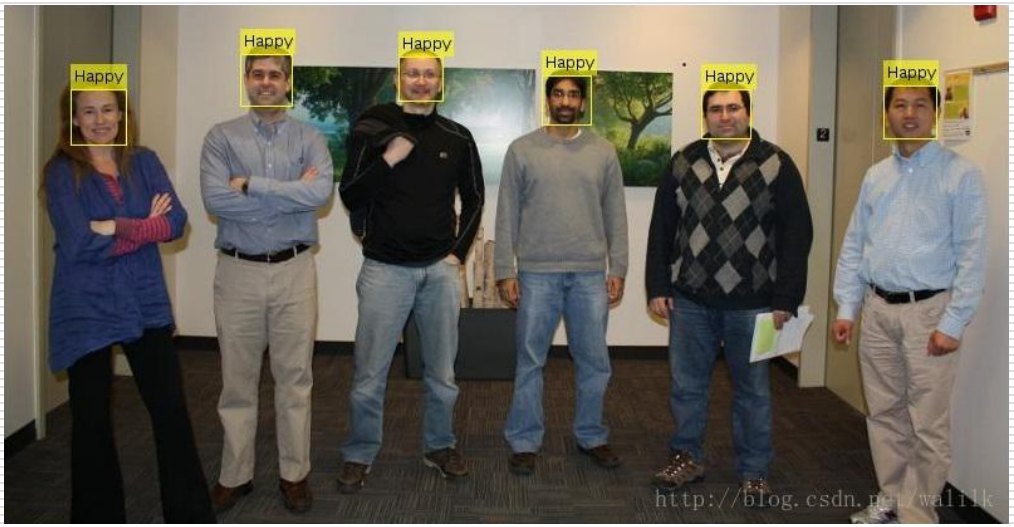
- 人脸特征提取
  - 提取到的人脸特征质量的优劣将直接影响输出结果正确与否
  - 提取到的特征往往是以特征向量的形式表示的
  - 特征提取过程可以看作一个数据抽取与压缩的过程

人脸对比场景

- 分类器
  - 一种分类算法

人脸识别之表情识别

- 20世纪Ekman等专家就通过跨文化调研提出了七类基础表情，分别是生气，害怕，厌恶，开心，悲伤，惊讶以及中立。
- Facial Expression Recognition 面部表情识别（FER）
- Automatic Facial Expression Analysis（AFEA）



人脸识别之表情识别

- 1971年，Ekman 和 Friesen 研究了人类的 6 种基本表情（即高兴、悲伤、惊讶、恐惧、愤怒、厌恶），并系统地建立了人脸表情图

表 1：表情脸的运动特征具体表现

表情	额头、眉毛	眼睛	脸的下半部
惊奇	①眉毛抬起，变高变湾 ②眉毛下的皮肤被拉伸 ③皱纹可能横跨额头	①眼睛睁大，上眼皮抬高，下眼皮下落 ②眼白可能在瞳孔的上边和/或下边露出来	下颌下落，嘴张开，唇和齿分开，但嘴部不紧张，也不拉伸
恐惧	①眉毛抬起并皱在一起 ②额头的皱纹只集中在中部，而不横跨整个额头	上眼睑抬起，下眼皮拉紧	嘴张，嘴唇或轻微紧张，向后拉；或拉长，同时向后拉
厌恶	眉毛压低，并压低上眼睑	在下眼皮下部出现横纹，脸颊推动其向上，当并不紧张	①上唇抬起 ②下唇与上唇紧闭，推动上唇向上，嘴角下拉，唇轻微凸起 ③鼻子皱起　④脸颊抬起
愤怒	①眉毛皱在一起，压低 ②在眉宇间出现竖直皱纹	①下眼皮拉紧，抬起或不抬起 ②上眼皮拉紧，眉毛压低 ③眼睛瞪大，可能鼓起	①唇有两种基本的位置：紧闭，唇角拉直或向下，张开，仿佛要喊 ②鼻孔可能张大
高兴	眉毛参考:稍微下弯	①下眼睑下边可能有皱纹，可能鼓起，但并不紧张 ②鱼尾纹从外眼角向外扩张	①唇角向后拉并抬高 ②嘴可能被张大，牙齿可能露出 ③一道皱纹从鼻子一直延伸到嘴角外部　④脸颊被抬起
悲伤	眉毛内角皱在一起，抬高，带动眉毛下的皮肤	眼内角的上眼皮抬高	①嘴角下拉 ②嘴角可能颤抖

人脸识别之表情识别

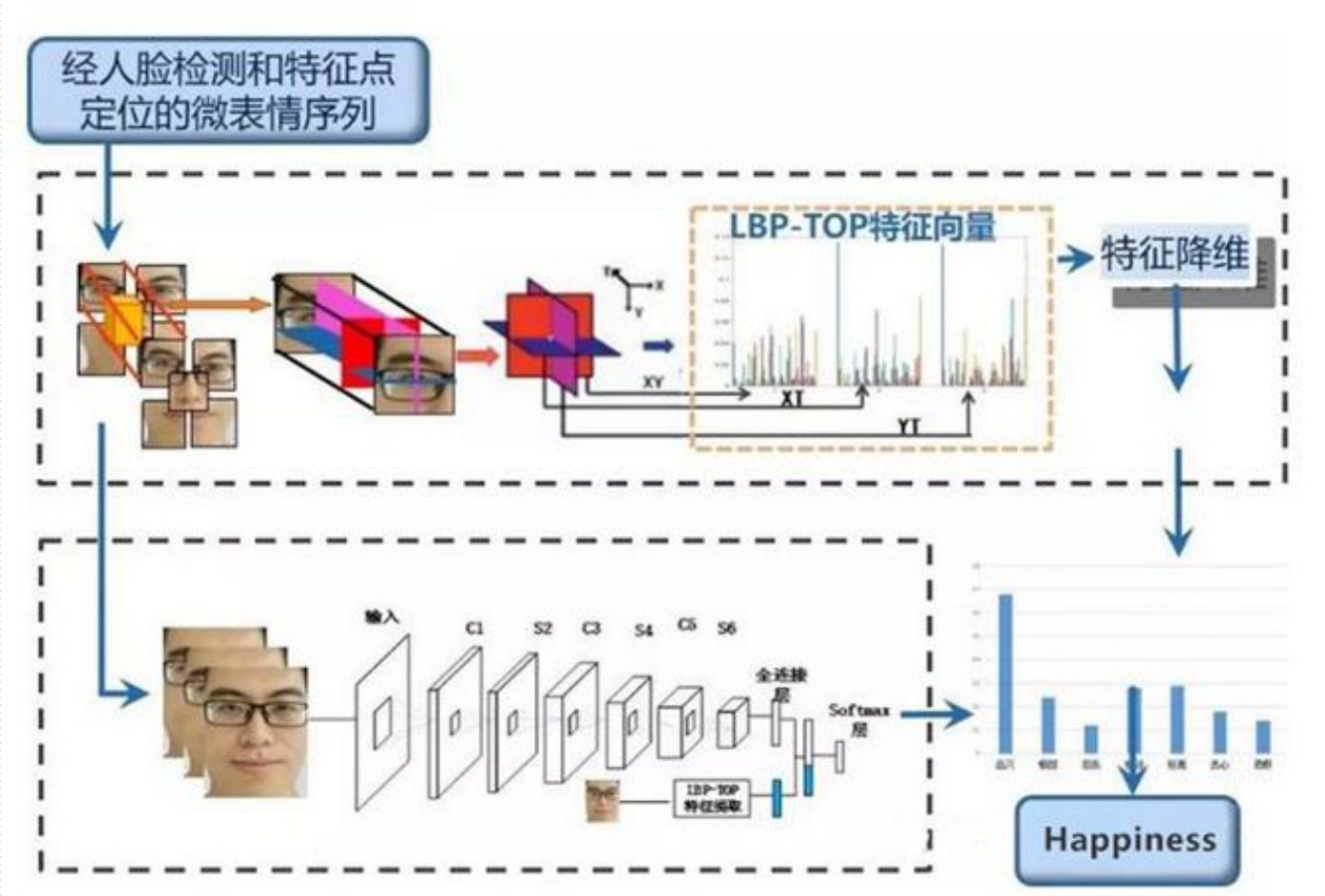
- 微表情（Micro-Expression）特指人类试图压抑或隐藏真实情感时泄露的非常短暂且不能自主控制的面部表情。
- 微表情的持续时间仅为 1/25 秒至 1/5 秒，表达的是一个人试图压抑与隐藏的真正情感。
- 微表情具有三个特点
  - 持续时间不超过1/5秒，能反映人的真实情感，在全人类是普遍存在的

人脸识别之表情识别

- 微表情的在自动谎言识别等众多领域有巨大的潜在应用价值
  - “无情感不智能”
    - 让电器拥有情感感知，让物体与人类产生情感的互动和交流。
      - 例如，当智能冰箱通过微表情或是动作读懂用户处于饥饿状态时，它将会自动打开冰箱等等。
    - 微表情识别系统“告状”：他上课溜号了
    - 屏幕上显示出约6款车的图片，通过识别顾客看图时的表情，成功锁定顾客最感兴趣的一辆车

人脸识别之表情识别

- 微表情自动分析可以分为检测和识别两个过程



人脸识别之表情识别

- 微表情持续时间短和动作幅度小两大识别难点，目前的识别率仍有很大的提升空间
- 基于深度学习的微表情识别工作流程
  - 1) 准备数据集：包含微表情的视频片段采集、视频图像归一化处理、训练/验证/测试集分割等；
  - 2) 设计学习模型：选择基本模型框架为卷积神经网络CNN+循环神经网络RNN、调整网络层数、确定损失函数、设计学习率等超参数；
  - 3) 训练模型：将模型输出误差通过BP算法反向传播，利用随机梯度下降SGD或Adam算法优化模型参数；
  - 4) 验证模型：利用未训练的数据验证模型的泛化能力，如果预测结果不理想，则需要重新设计模型，进行新一轮的训练

人脸识别之微表情识别

- Enriched Long-term Recurrent Convolutional Network for Facial Micro-Expression Recognition
  - https://github.com/IcedDoggie/Micro-Expression-with-Deep-Learning
  - 一个丰富的长期递归卷积网络（ELRCN）框架，首先通过CNN将每个微表情帧编码成特征向量，然后将特征向量通过长-短期记忆（LSTM）预测微表情。该框架包含两种不同的网络变体：
    - （1） 空间富集的输入数据的通道叠加
    - （2） 时间富集的特征的功能性叠加

人脸识别之表情识别

- 表情识别的应用场景
  - 驾驶员疲劳检测，医疗，谎言检测、人机交互、智能控制、安全、通信等领域
    - 商场门店的顾客情绪分析
    - 教育辅助机器人项目
      - 通过面部表情分析来判断机器人眼前的用户的情绪和心理
- 在线应用接口
  - 微软提供了表情识别的API接口，并通过JSON返回识别结果，如下：
    - Cognitive Services APIs - Emotion Recognization
  - Face++也提供了接口，并通过JSON返回识别结果，如下：
    - 人脸检测

人脸识别之表情识别

- 静态图像FER（图片表情识别）
- 动态序列FER（基于视频序列建模，如RNN方式等）
  - 传统的手工特征（LBP,LBP-TOP等），浅层学习（SVM，Adaboost等），深度学习（CNN，DBN，RNN）。
  - 从2013年开始，开始有了表情识别的比赛，如FER2013,EmotiW等
    - https://cs.anu.edu.au/f

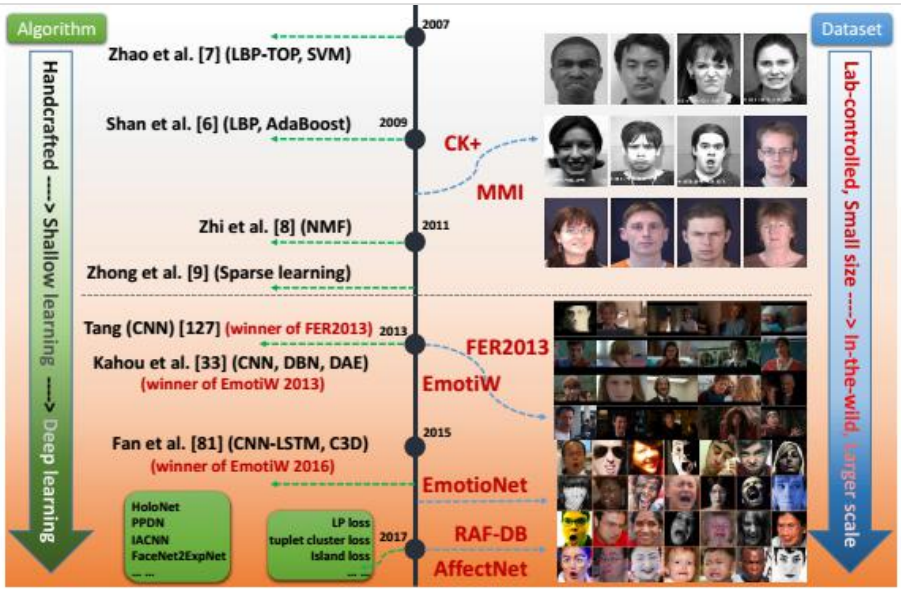


Fig. 1. The evolution of facial expression recognition in terms of datasets and methods.

基于深度学习的面部表情识别

- 预处理
- 深度特征学习
- 面部表示分类

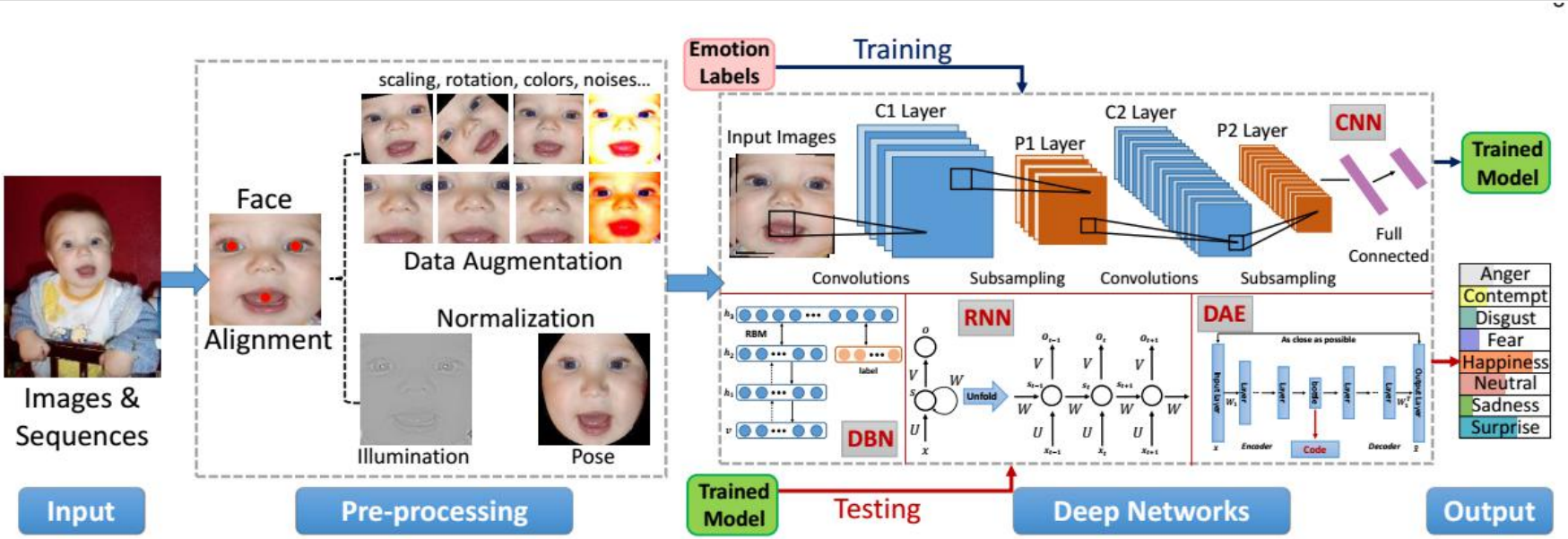
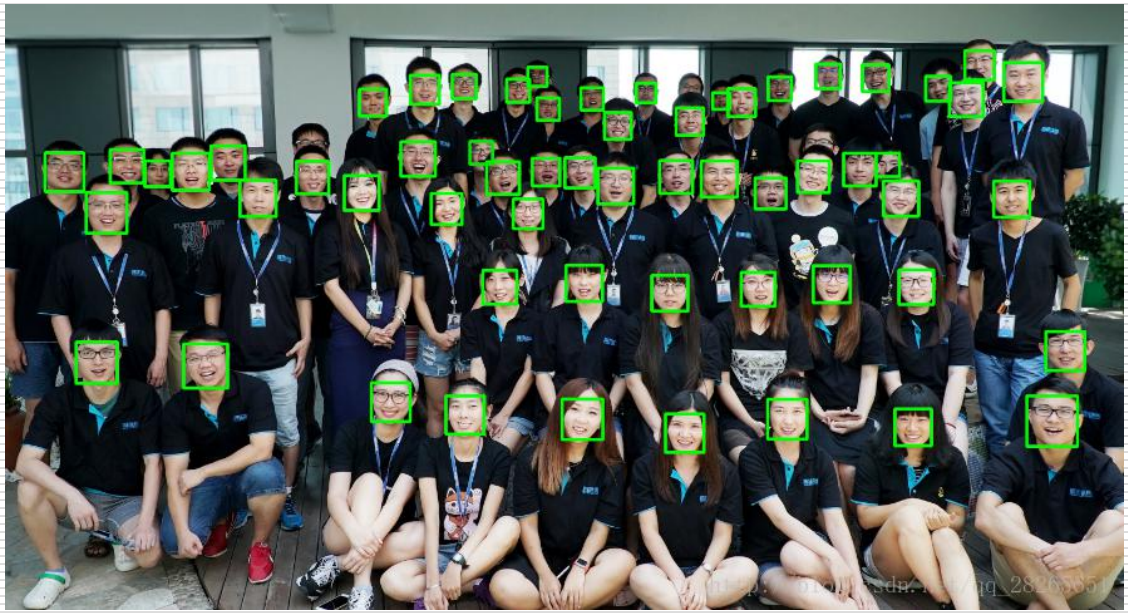


Fig. 2. The general pipeline of deep facial expression recognition systems.

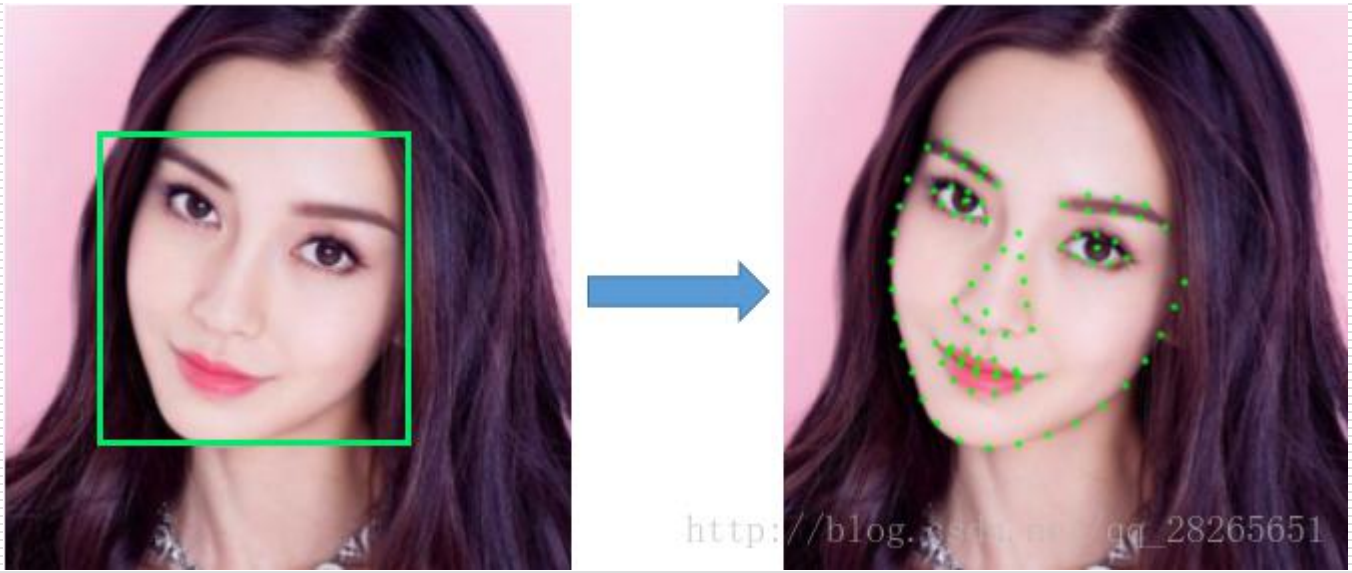
基于深度学习的面部表情识别

- 预处理
  - 人脸检测
    - 检测出图像中人脸所在位置的
    - “扫描”加“判别”



基于深度学习的面部表情识别

- 预处理
  - 人脸对齐
    - 根据人脸，检测出人脸定位点landmark.
      - 定位出人脸上五官关键点坐标
    - 输入是“一张人脸图片”加“人脸坐标框”，输出五官关键点的坐标序列
      - 主流的有IntraFace,采用SDM算法

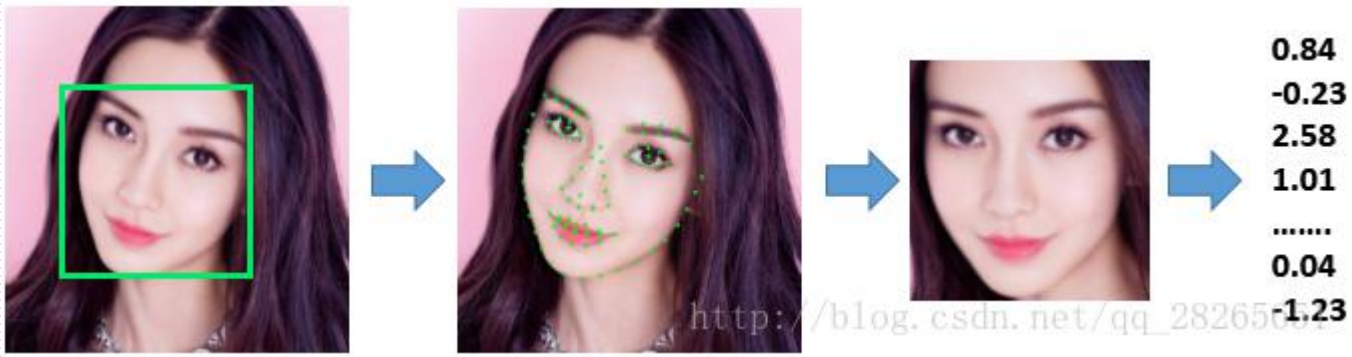


基于深度学习的面部表情识别

- 预处理
  - 数据增强
    - 离线数据增强，包含随机扰动，变换（旋转，平移，翻转，缩放，对齐），噪声添加如椒盐噪声，斑点噪声，以及亮度，饱和度变化，以及在眼睛之间添加2维高斯随机分布的噪声
    - 在线数据增强，包含Crop,水平翻转等
  - 人脸归一化
    - 亮度归一化和姿态归一化（就是人脸对齐拉正）

基于深度学习的面部表情识别

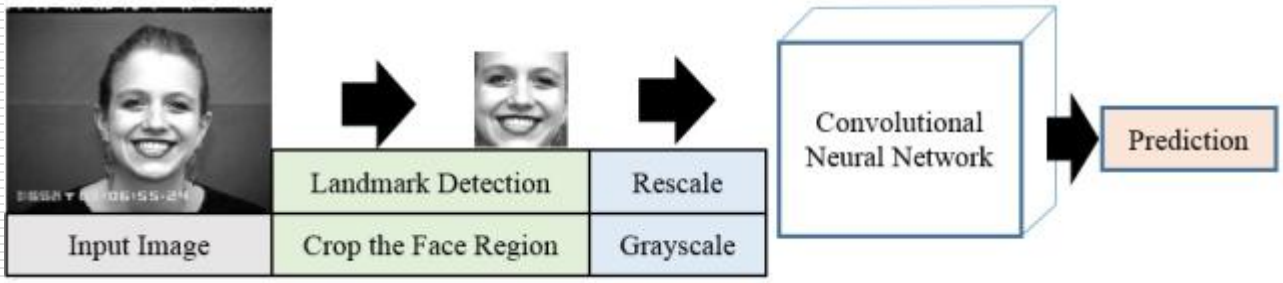
- 深度特征学习
  - “人脸提特征(Face Feature Extraction)”是将一张人脸图像转化为一串固定长度的数值的过程



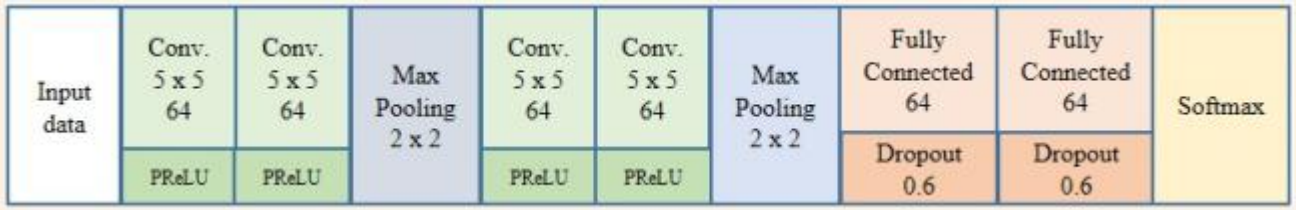
- 面部表示分类

基于深度学习的面部表情识别

- 2018 CVPR表情识别论文 A Compact Deep Learning Model for Robust Facial Expression Recognition
  - 首先通过IntraFace检测出的人脸关键点进行裁剪，然后将其resize成120x120，最后将其96x96的中心区域作为卷积网络的输入进行预测

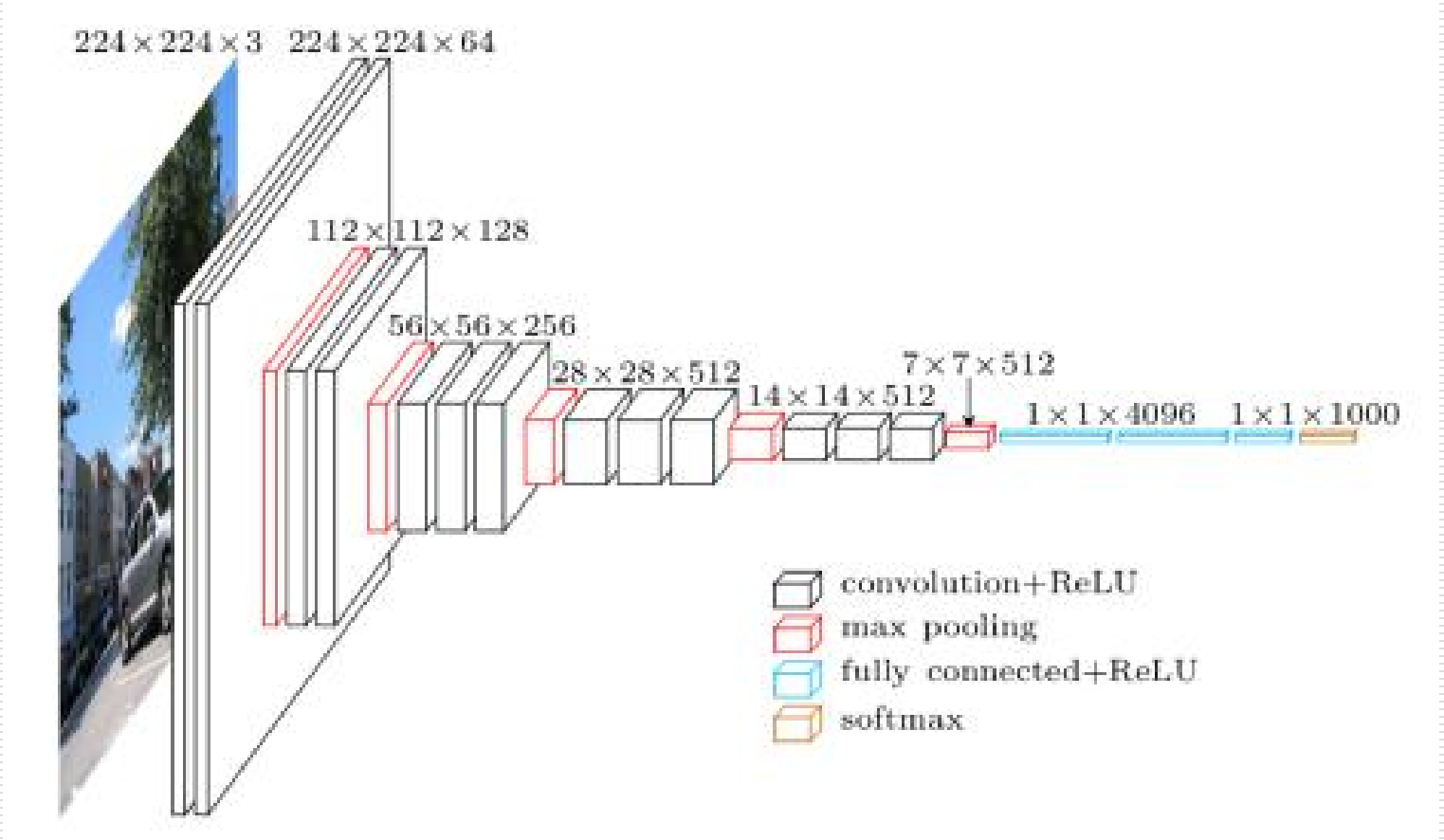


  - 网络结构



VGG模型

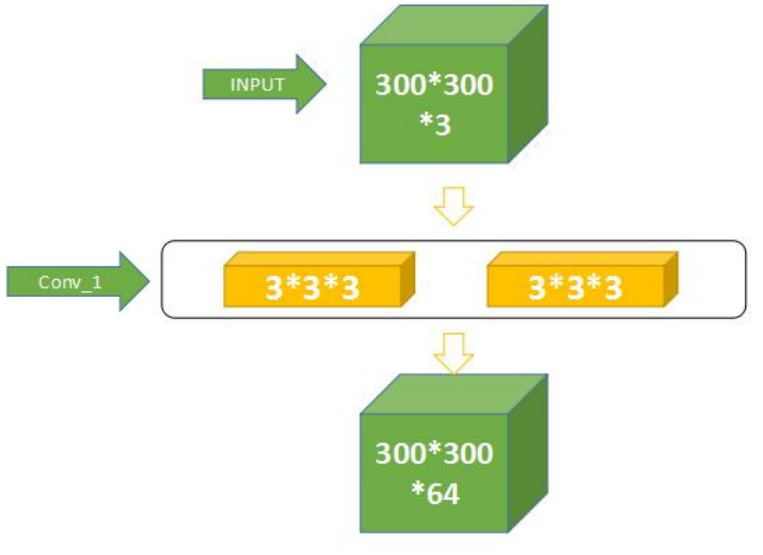
- VGG模型不仅能够在大规模数据集上的分类效果很好，其在其他数据集上的推广能力也非常出色

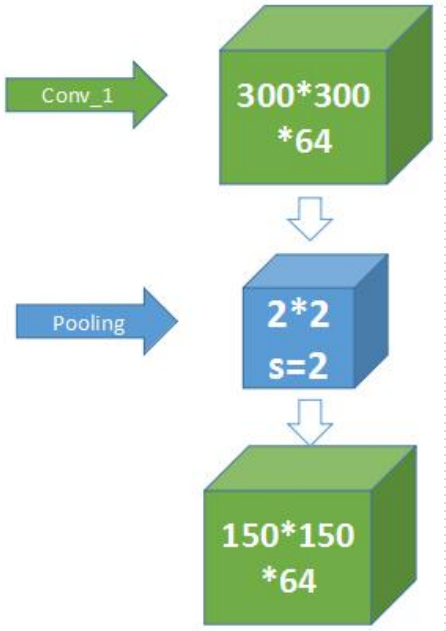


从keras看VGG16

- 从INPUT到Conv1：
- 从Conv1到Conv2之间的过渡

```
model = Sequential()
model.add(ZeroPadding2D((1,1),input_shape=(3,224,224)))
model.add(Convolution2D(64, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(64, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))
```

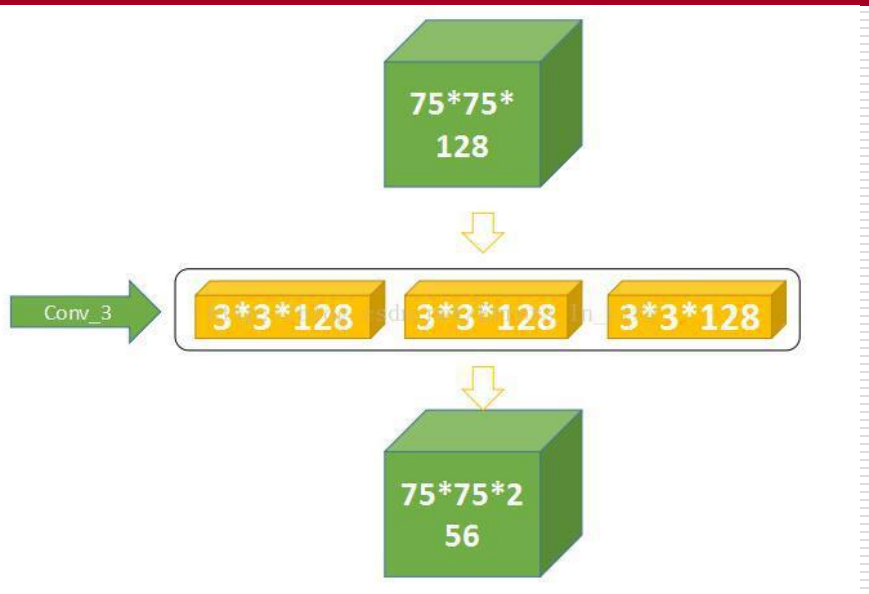


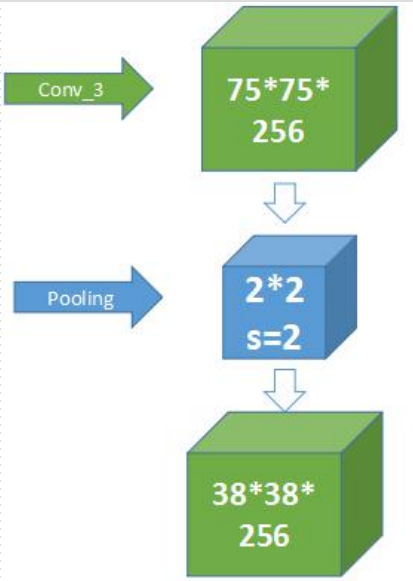


从keras看VGG16

- 进入Conv3的推演
- 从Conv3到Conv4之间的过渡

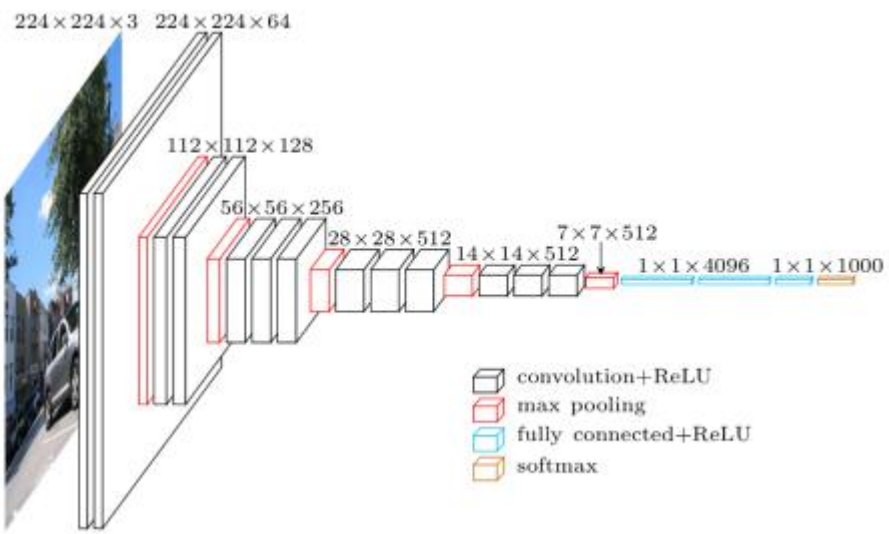
```
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(128, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(128, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))
```





从keras看VGG16

```
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(256, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(256, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(256, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))
```




```
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))

model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(ZeroPadding2D((1,1)))
model.add(Convolution2D(512, 3, 3, activation='relu'))
model.add(MaxPooling2D((2,2), strides=(2,2)))

model.add(Flatten())
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1000, activation='softmax'))
```

思考题

- 今天的深度学习技术为什么会有如此大的成功？
- 它会持续的在其他各个领域都奏效吗？

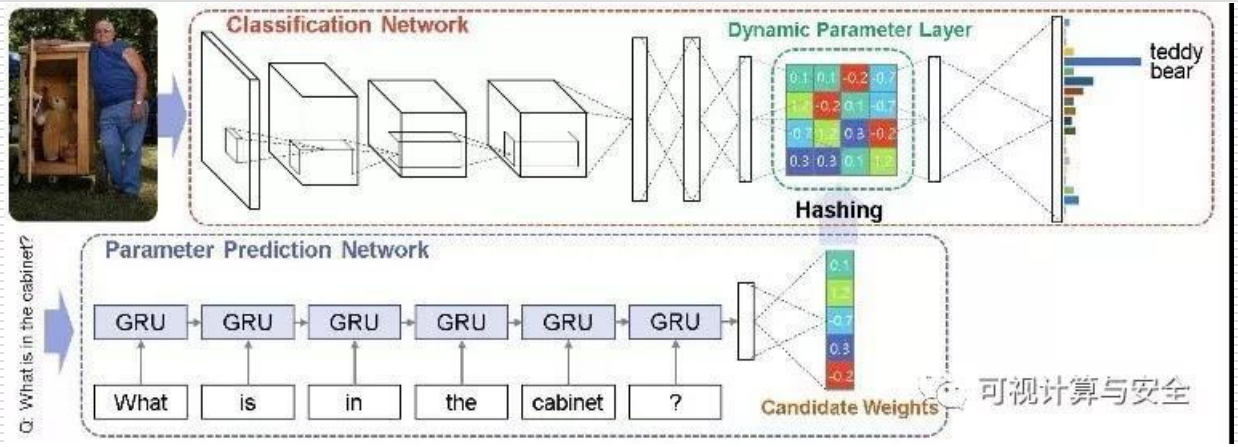
THE END

视觉问答技术

- 基线模型（Baseline Models）
  - 论文：H. Noh,P. H. Seo, and B. Han, “Image question answering using convolutional neuralnetwork with dynamic parameter prediction,” in The IEEE Conference on ComputerVision and Pattern Recognition (CVPR), 2016.
  - 工作：通过学习一个动态参数层的卷积神经网络(CNN)，进行网络权值的自适应预测。
  - 在自适应参数预测中，我们采用了一个单独的参数预测网络，该网络由以问题为输入的门控递归单元和生成一组候选权重作为输出的全连通层组成。通过使用哈希技术来减少这个问题的复杂性，使用预定义的哈希函数选择参数预测网络给出的候选权重，以确定动态参数层中的各个权重，进而提高基线模型的准确率。

视觉问答技术

- 基线模型（Baseline Models）
  - 代码：https://github.com/MarvinTeichmann/tensorflow-fcn

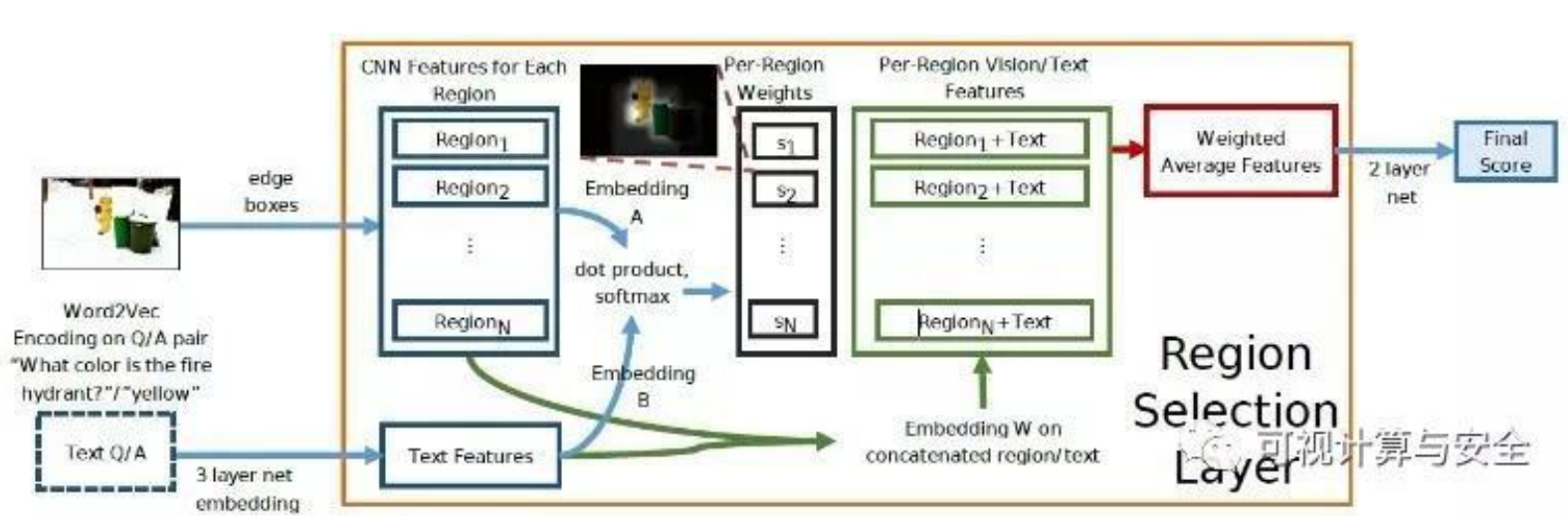


视觉问答技术

- 注意力模型（Attention BasedModels）
  - 论文：K. J.Shih, S. Singh, and D. Hoiem, “Where to look: Focus regions for visual questionanswering,” in The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016.
  - 工作：美国伊利诺伊大学的这项工作使用CNN来从这些盒子中提取特征。VQA系统的输入包括这些CNN特征，问题特征和一个选择性的答案。他们的系统被训练为每一个选择题的答案都能得到一个分数，并选出得分最高的答案。通过传递CNN区域特征点积以及问题，最后合并到一个全连接层中，可以简单地计算出权重的每个区域的加权平均得分。

视觉问答技术

- 注意力模型（Attention BasedModels）
  - 代码：https://github.com/kevjshih/wtl_vqa

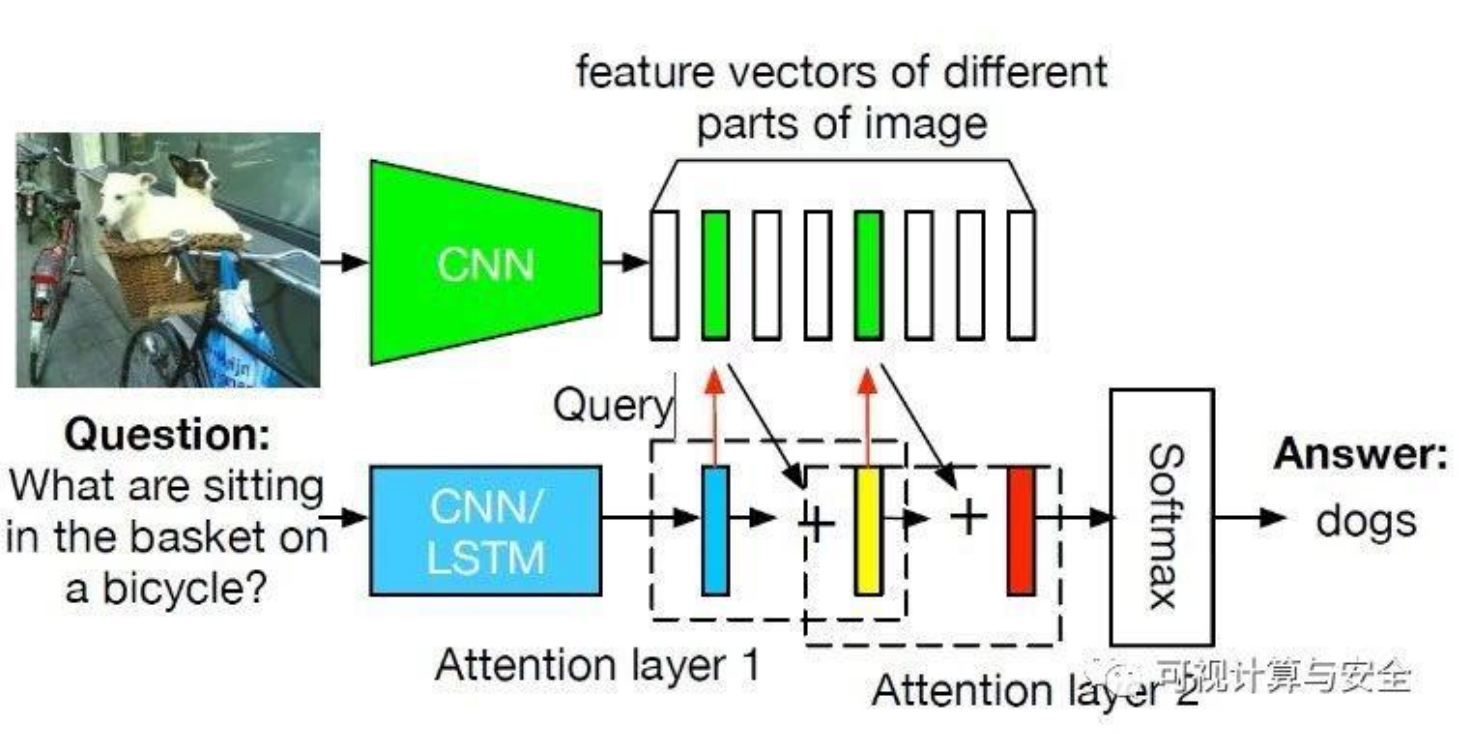


视觉问答技术

- 堆积注意力网络 Stacked Attention Network
  - 论文：Z.Yang, X. He, J. Gao, L. Deng, and A. J. Smola, “Stacked attention networks forimage question answering,” in The IEEE Conference on Computer Vision andPattern Recognition (CVPR), 2016.
  - 工作：美国卡耐基梅隆大学和微软提出的动态注意力网络。其注意层由一个单独的权重层确定，该权重层用问题和带激活函数的CNN的特征图计算图像位置上的注意力分布。然后将该分布应用到CNN 的特征层中，使用加权和在空间特征位置之间进行聚合，从而生成一个完整的图像，它比其他区域更强调某些空间的区域。将这个特征向量与问题特征向量结合，可得到预测答案。

视觉问答技术

- 堆积注意力网络 Stacked Attention Network
  - 代码：https://github.com/zcyang/imageqa-san

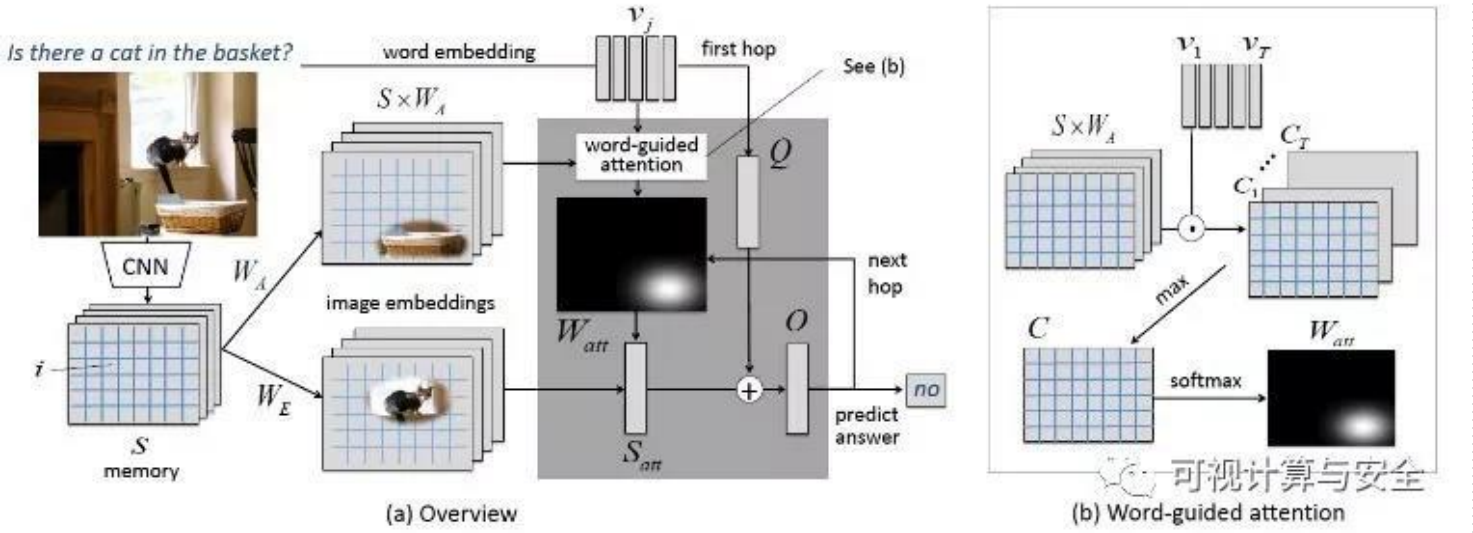


视觉问答技术

- 空间记忆网络Spatial Memory Network
  - 论文：H. Xu and K.Saenko, “Ask, attend and answer: Exploring question-guided spatial attentionfor visual question answering,” in European Conference on Computer Vision(ECCV), 2016.
  - 工作：美国麻省理工学院提出的空间记忆网络，该模型通过估计图像斑块与问题中单个单词的相关性来产生空间关联性。这种文字引导的注意力被用来预测注意力分布，然后用来计算图像视觉特征的加权和。然后探索了两种不同的模型。在单跳模型中，将编码整个问题的特征与加权视觉特征相结合并预测答案。在两跳模型中，将视觉特征和问题特征的组合循环回到关注机制中，用于细化注意力分配。

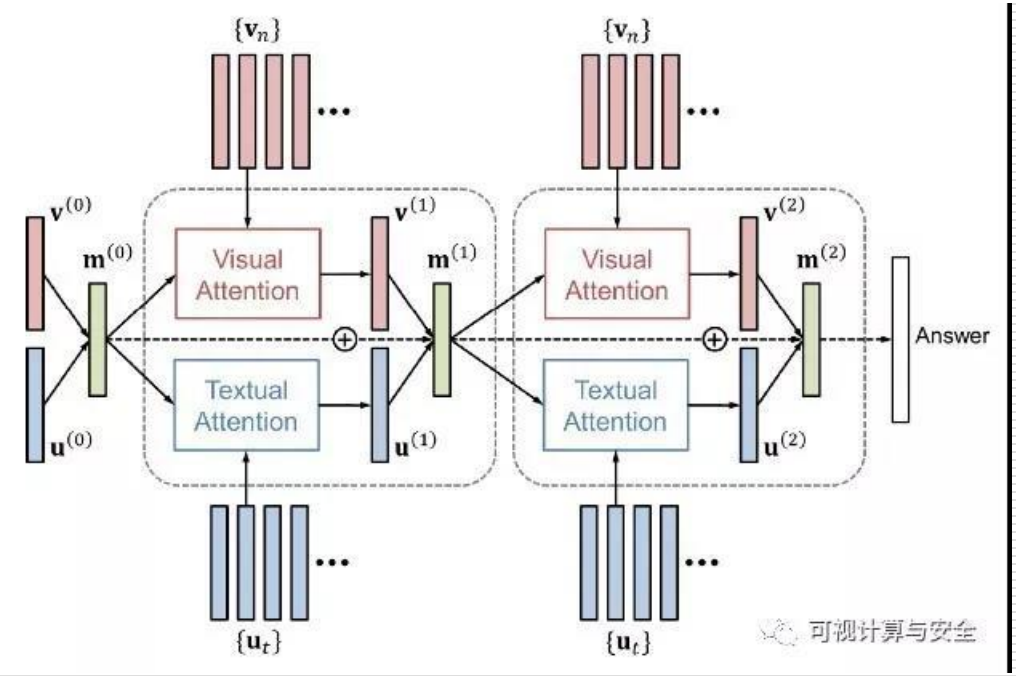
视觉问答技术

- 空间记忆网络Spatial Memory Network
  - 代码：https://github.com/VisionLearningGroup/Ask_Attend_and_Answer



视觉问答技术

- DAN
  - 论文：H. Nam,J. Ha, and J. Kim. Dual attention networks for multimodal reasoning andmatching. In The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2017.
  - 代码：https://github.com/iammrhelo/pytorch-vqa-dan



视觉问答技术

- 双线性融合（Bilinear PoolingMethods）
  - MLB
    - 论文：J.-H.Kim, K.-W. On, J. Kim, J.-W. Ha, and B.-T. Zhang, “Hadamard product forlow-rank bilinear pooling,” arXiv preprint arXiv:1610.04325, 2016.
    - 工作：首尔国立大学的工作者们使用多模态低秩双线性池(MLB)方案，使用Hadamard乘积和线性映射来实现近似双线性池。当与空间视觉注意机制一起使用时，MLB可以与VQA中的MCB相媲美，但计算复杂度较低，并且使用的神经网络参数较少。
    - 代码：https://github.com/jnhwkim/MulLowBiVQA

视觉问答技术

- 双线性融合（Bilinear PoolingMethods）
  - MLB
    - 论文：J.-H.Kim, K.-W. On, J. Kim, J.-W. Ha, and B.-T. Zhang, “Hadamard product forlow-rank bilinear pooling,” arXiv preprint arXiv:1610.04325, 2016.
    - 工作：首尔国立大学的工作者们使用多模态低秩双线性池(MLB)方案，使用Hadamard乘积和线性映射来实现近似双线性池。当与空间视觉注意机制一起使用时，MLB可以与VQA中的MCB相媲美，但计算复杂度较低，并且使用的神经网络参数较少。
    - 代码：https://github.com/jnhwkim/MulLowBiVQA

视觉问答技术

- 双线性融合（Bilinear PoolingMethods）
  - MCB
    - 论文：A.Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, “Multimodal compact bilinear pooling for visual question answering and visualgrounding,” in Conference on Empirical Methods on Natural Language Processing(EMNLP), 2016.
    - 工作：美国加州大学和日本索尼公司在众多的基线模型中发现，由于在双线性融合的过程中，计算高维双线性外积通常是不可行的，而使用多模态双线性池(MCB)来高效地表达多模态特征。在视觉问答方面对MCB进行了范围较广的评价，取得了目前所有基线模型中效果最好的结果。
    - 代码：https://github.com/akirafukui/vqa-mcb

视觉问答技术

- 双线性融合（Bilinear PoolingMethods）
  - MFB
    - 论文：Z. Yu,J. Yu, J. Fan, and D. Tao. Multi-modal Factorized Bilinear Pooling withCo-Attention Learning for Visual Question Answering. ArXiv e-prints, Aug. 2017.
    - 工作：本文提出了一种多模态分解双线性池化方法，有效地结合了多模态特征，使VQA性能优于其他双线性融合方法。对于细粒度的图像和问题表示，我们开发了一种“共享注意力”机制，使用端到端深度网络架构来共同学习图像和问题注意力。
    - 代码：https://github.com/yuzcccc/vqa-mfb

视觉问答技术

- 双线性融合（Bilinear PoolingMethods）
  - MFH
    - 论文：Z. Yu,J. Yu, C. Xiang, J. Fan, and D. Tao. Beyond bilinear: Generalized multimodalfactorized high-order pooling for visual question answering. TNNLS, 2018.
    - 工作：针对多模态特征融合问题，作者提出了一种广义多模态因子分解的高阶池化方法(MFH)，通过充分利用多模态特征之间的相关性，实现多模态特征的更有效融合，进一步提高VQA性能。在答案预测中，利用KL (Kullback-Leibler)散度作为损失函数，可以更准确地表征具有相同或相似含义的多个不同答案之间的复杂关联，从而使我们能够获得更快的收敛速度，并在答案预测中获得略好的准确性。
    - 代码：https://github.com/yuzcccc/vqa-mfb