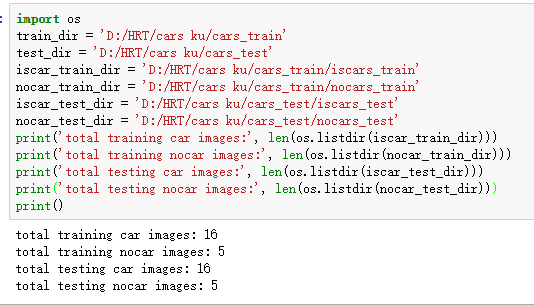
**目标识别技术**

**深度学习（卷积神经网络）**

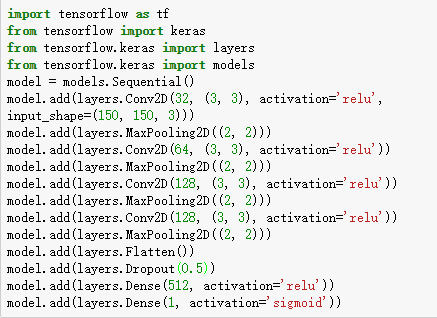
**技术报告**

1. **理论基础**
2. 卷积神经网络是Conv2D和MaxPooling2D层的堆叠，每个Conv2D层和MaxPooling2D层的输出都是一个形状为（height, width, channels）的3D张量。
3. 密集连接层和卷积层的根本区别在于，Dense 层从输入特征空间中学到的是全局模式，而卷积层学到的是局部模式。

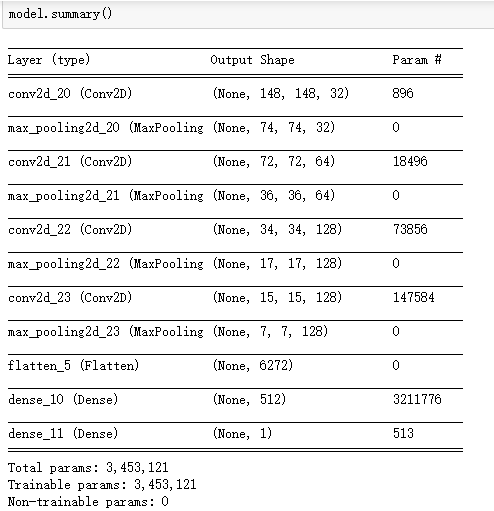
**二．主要思路**



6个文件目录的导入，采用了添加图片干扰项的方法，分别在训练集和测试集上划分出车和非车的照片



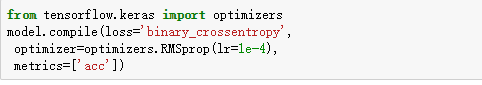
引入Sequential模型，构建神经网络，Con2D是卷积层，MaxPooling2D是池化层，激活函数为relu，因为Dense层的输入应该是1D张量，因此进入Dense层前需用Flatten层将3D张量转化为1D张量，Dropout层防止过拟合，为解决图片车和非车的问题，因此最后采用sigmoid层，



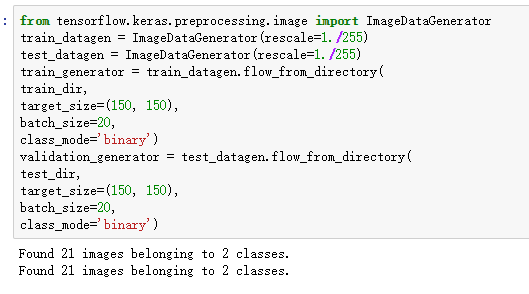
采用model.summary函数查看神经网络的架构，图片的输入是一个（150， 150， 3）格式的3D张量，分别对应高度，宽度和深度（RGB）.

进入Conv2D层后，卷积通常使用 3\*3 窗口和步幅 1，进行卷积运算，步幅为1，仍输出一个3D张量，但高度和宽度都会减小，这是由于边缘效应（如需高度宽度大小不变可用填充的方法），第三个轴上的张量维度为过滤器的数量，例如第一层卷积层后，有 32 个输出通道，每个通道都包含一个148\*148的特征图，是这个过滤器在输入上响应的二维空间图。

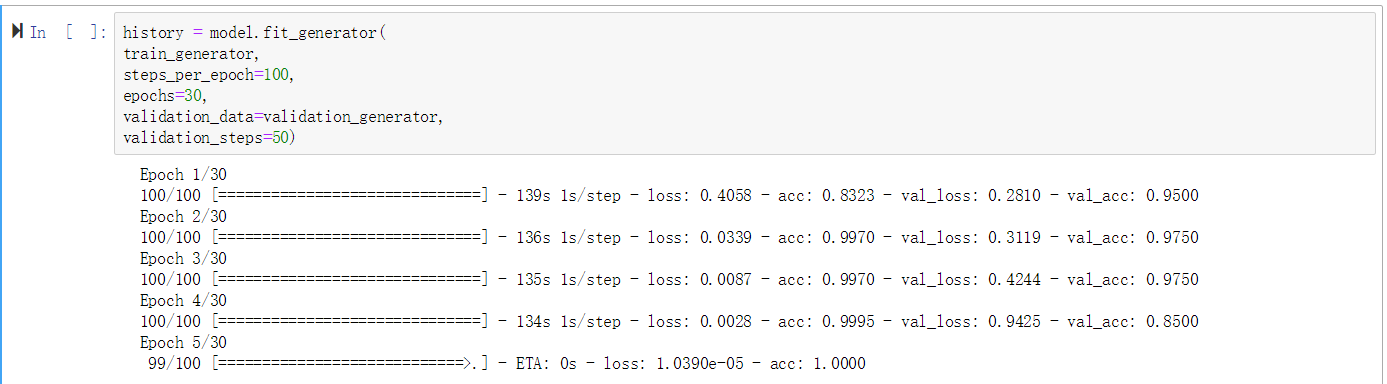
进入MaxPooling层后，最大池化采用用 2\*2 的窗口和步幅 2，每个过滤器输出其最大值，因此输出图片的高度和宽度为原来一半，最终效果是减少需要处理的特征图的元素个数，还可以通过让连续卷积层的观察窗口越来越大（即窗口覆盖原始输入的比例越来越大），从而引入空间过滤器的层级结构。



对于网络的编译，由于网络的最后一层为sigmoid层，因此采用二元交叉熵作为损失函数。采用RMprop作为优化器



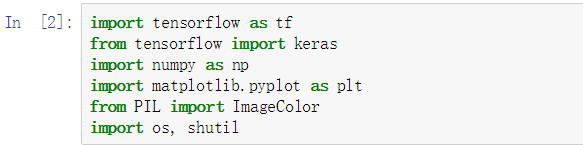
调用图像处理模块keras.preprocessing.image，包含ImageDataGenerator 类,首先将图像乘以1./255缩放，将像素值（0~255 范围内）缩放到 [0, 1] 区间，传入图片目录，将图片大小设置为150\*150，采用二进制标签，创建train\_generator生成器.



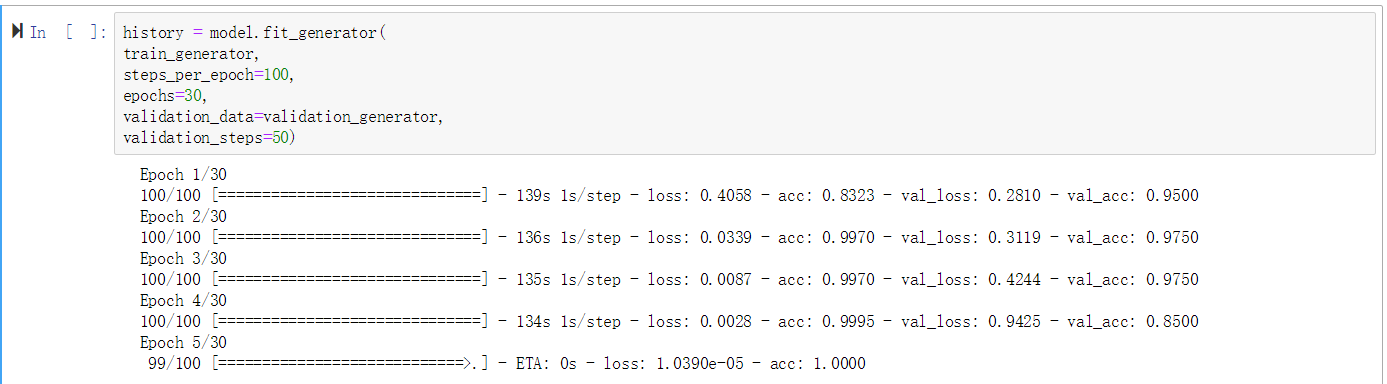
利用生成器生成的train\_generator,让模型对数据进行拟合,使用 fit\_generator 方法来拟合,它在数据生成器上的效果和 fit 相同。它的第一个参数是上一步生成的train\_generator生成器,可以不停地生成输入和目标组成的批量，同样可以通过validation\_generator导入validation\_data,并设置步长为50进行验证。

**三．代码**

**1.**代码使用python3.7编写，运行环境有以下：



**2.**运行截图



由于运算量过大，数据集仅包含：训练集40张图片，测试集40张图片。