

数据科学项目报告

##毕业项目 狗分类识别 丁道华 18188241@qq.com

2019 年 9 月 13 日

一、问题的定义

项目概述

该项目的目标是将狗狗的图片根据品种进行分类。该数据集来自sklearn，其中包含133个类别的8351张狗图像。Udacity已经提供模板代码及其他项目所需资源。熟悉图像识别分类流程，给定一张小狗的图像，算法将能够大致识别出小狗的品种，如果提供的是人脸，代表能识别出相似的小狗品种。

问题陈述

该项目创建图像中检测人脸和小狗后，需要预测品种。熟悉CNN模型，搭建模型，并了解如何使用迁移学习。在项目开始时尝试自行搭建CNN，效果极不佳，无法使用到真实环境，使用Xception，进行迁移学习。学习该项目能够具备CNN区分狗的种类，以及能够搭建简单的CNN模型，并了解令人惊喜的迁移学习方法，该方法能够最大的利用已经知识模型，并将其应用一个相关问题。使得个人用户或小微企业，节省大量的计算耗时，快速的得到最佳方式。也是为企业新任务时，可重复的利用之前已经训练好的模型。

评价指标

是评价分类指标的准确性，分类数据为133类，数据正常，种类平衡，使用简单的准确率即可。

通过迁移学习，选择VGG-19, ResNet-50, Inception, Xception其中一类，达到85%以上识别准确率。

二、分析

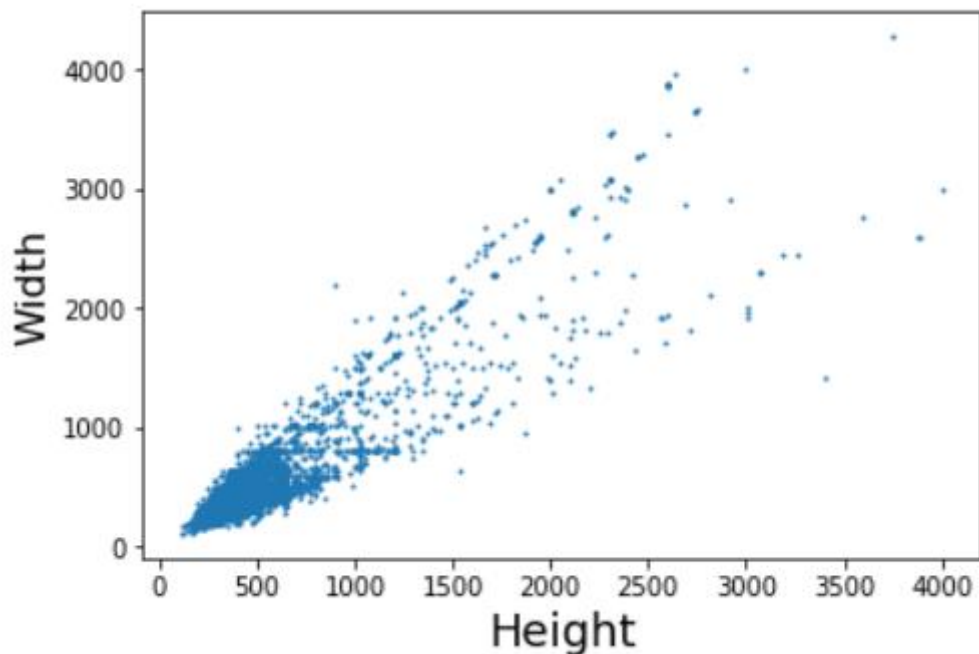
数据的获取

```
1 def load_dataset(path):
2     data = load_files(path)
3     dog_files = np.array(data['filenames'])
4     dog_targets = np_utils.to_categorical(np.array(data['target']), 133)
5     return dog_files, dog_targets
6
7 # load train, test, and validation datasets
8 train_files, train_targets = load_dataset('../../data/dog_images/train')
9 valid_files, valid_targets = load_dataset('../../data/dog_images/valid')
```

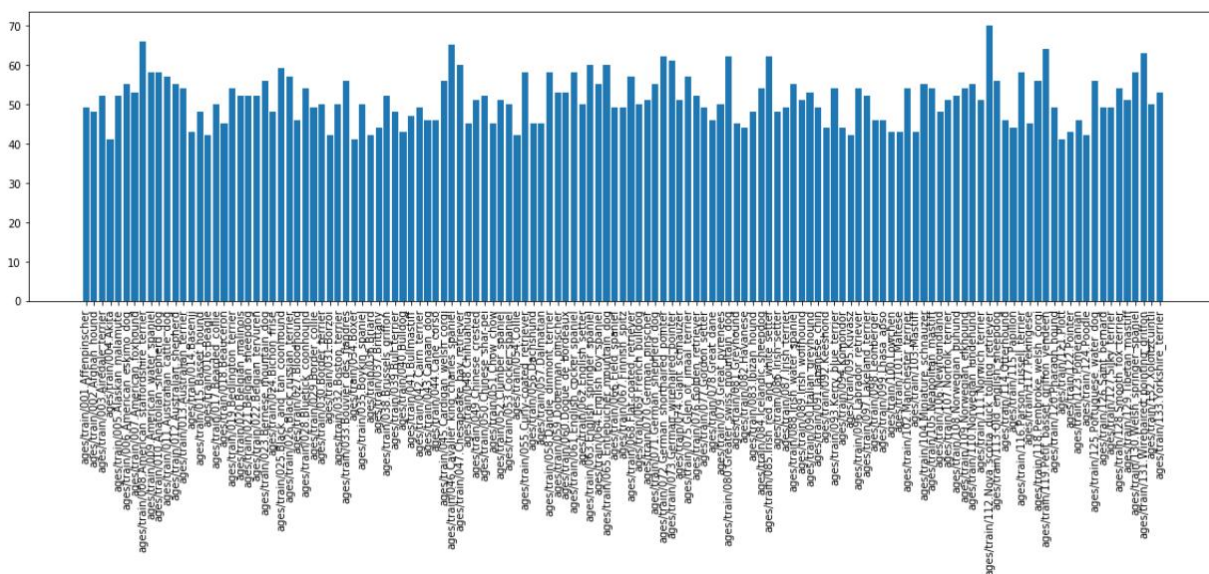
```
10 test_files, test_targets = load_dataset('.././../data/dog_images/test')
```

数据探索和可视化

数据文件有133个类别和8351个图像。



显示训练文件中图片大小的分布，不同种类狗的照片尺寸不一样，集中在400-500像素这间。



显示训练文件中，小狗种类的分布基本均匀。

数据预处理

当后端使用 TensorFlow 时，Keras CNN 要求输入的是一个 4D 的 numpy 数组（我们也称之为 4 维张量），其形状为：

(nb_samples, rows, columns, channels),

其中 `nb_samples` 表示图片 (或样本) 的数量, `rows`、`columns` 和 `channels` 表示每个图片各自的行数、列数和通道数。

`path_to_tensor` 函数接受一个彩色图片的文件路径字符串, 返回一个适用于 Keras CNN 训练的四维张量。这个函数首先加载图片, 然后将其重新调整为 224×224 像素的形状。接下来, 图片被转成 `numpy` 数组, 之后被整理成 4D 张量。在这个案例中, 因为我们处理的是彩色图片, 每个图片有 3 个通道。还有, 我们处理的是单张图片, 返回的张量的形状就是:

`(1,224,224,3)`。

`paths_to_tensor` 函数接受一个图片路径的字符串 `numpy` 数组, 返回一个四维张量。该张量的形状是:

`(nb_samples,224,224,3)`。

其中, `nb_samples` 是输入的图片路径中对应的样本数, 或者说图片的数目。建议你把 `nb_samples` 认为是 3 维张量的数量 (每个 3 维张量就对应数据集里的一张图片) 。

检测人脸

我们使用 OpenCV 实现的 [Haar 特征级联分类器 \(Haar feature-based cascade classifiers\)](#) 来检测图片中的人脸。

检测小狗

我们使用预训练的 [ResNet-50](#) 模型来检测图片中的狗狗。我们的第一行代码下载了 ResNet-50 模型, 以及在 [ImageNet](#) 预训练后的权重信息。ImageNet 是一个庞大且常用的数据集, 用于图像分类和其他视觉任务。ImageNet 包含 1000 万以上的 URL, 每个都链接到包含某个对象的图像, 这些对象分成了 [1000 个类别](#)。输入一个图片, 预训练的 ResNet-50 模型预测图片中包含了什么物体, 并返回对应的标签 (标签是 ImageNet 内置的一些类别) 。

创建分类小狗品种CNN

创建好从图像中检测人脸和小狗的函数后, 我们需要预测图像中的小狗品种。

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 223, 223, 16)	208
max_pooling2d_1 (MaxPooling2D)	(None, 111, 111, 16)	0
conv2d_2 (Conv2D)	(None, 110, 110, 32)	2080
max_pooling2d_2 (MaxPooling2D)	(None, 55, 55, 32)	0
conv2d_3 (Conv2D)	(None, 54, 54, 64)	8256
max_pooling2d_3 (MaxPooling2D)	(None, 27, 27, 64)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 64)	0
dense_1 (Dense)	(None, 133)	8645
Total params: 19,189.0		
Trainable params: 19,189.0		
Non-trainable params: 0.0		

INPUT

CONV

POOL

CONV

POOL

CONV

POOL

GAP

DENSE

基准模型

识别率达到60%以上。

三、方法

迁移学习特征提取

```

1 bottleneck_features = np.load('/data/bottleneck_features/DogXceptionData.npz')
2 train_Xception= bottleneck_features['train']
3 valid_Xception = bottleneck_features['valid']
4 test_Xception = bottleneck_features['test']

```

模型架构

```

1 ### TODO: Define your architecture.
2 Xception_model = Sequential()
3 Xception_model.add(GlobalAveragePooling2D(input_shape=train_Xception.shape[1:]))
4 Xception_model.add(Dense(133, activation='softmax'))
5 Xception_model.summary()

```

训练模型

```

1 ### TODO: Train the model.
2 checkpointer = ModelCheckpoint(filepath='saved_models/weights.best.Xception.hdf5',
3 verbose=1, save_best_only=True)
4
5 Xception_model.fit(train_Xception, train_targets,
6 validation_data=(valid_Xception, valid_targets),

```

```
7 epochs=10, batch_size=20, callbacks=[checkpointer], verbose=1)
```

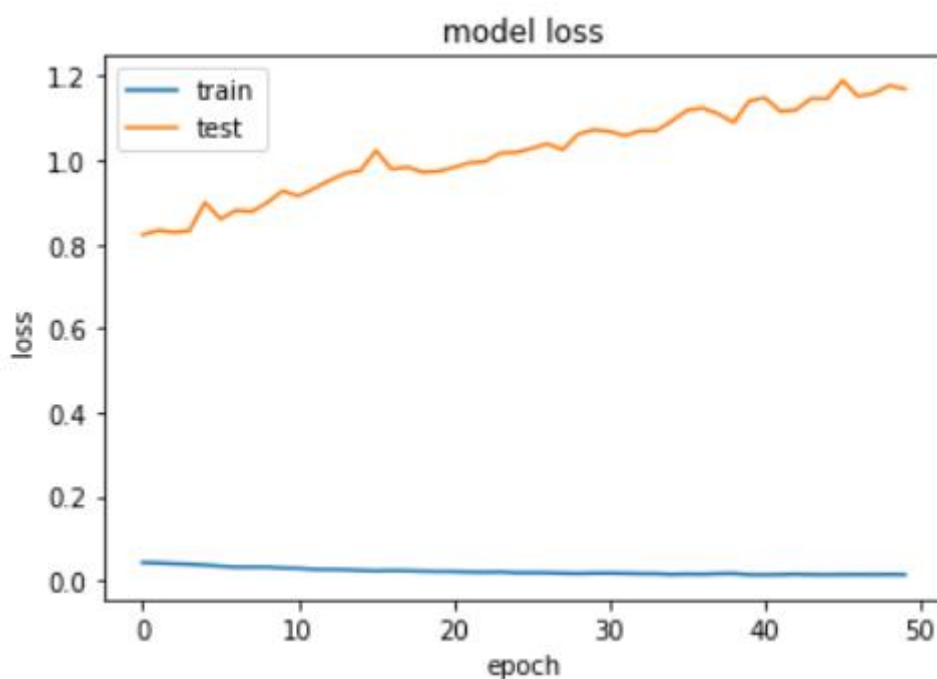
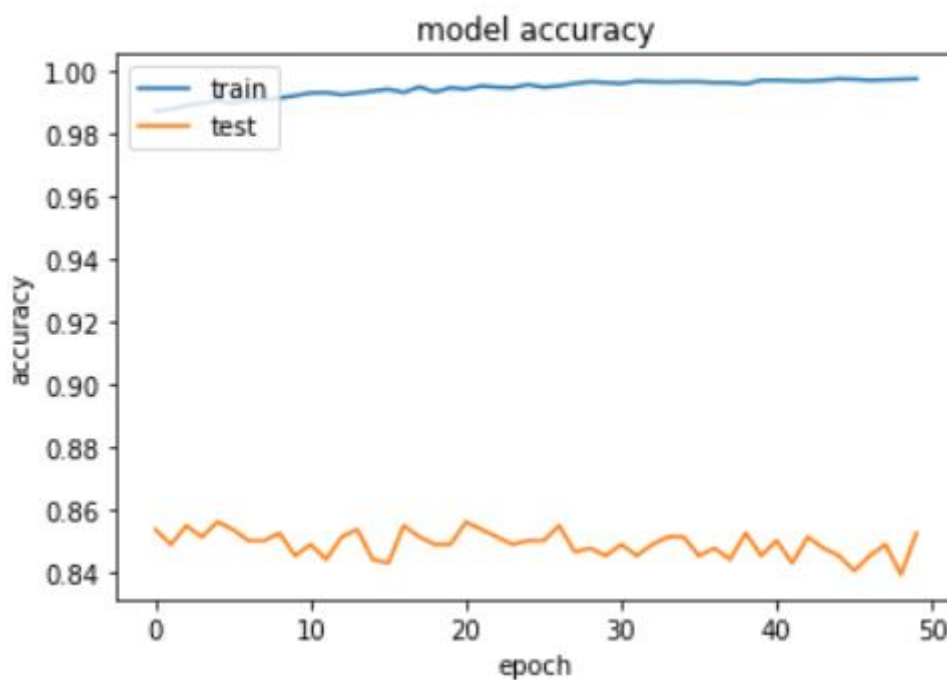
四、结果

识别率达到85%以上。

使用简单的OpenCV人脸检测器，它的性能非常好。人脸检测器能够识别所有人类的100%，但是，它也发现11%的狗有人脸。

狗检测器能够确认100%的狗是狗，并正确地确认0%的人类是狗。

```
1 # list all data in history
2 print(history.history.keys())
3 # summarize history for accuracy
4 plt.plot(history.history['acc'])
5 plt.plot(history.history['val_acc'])
6 plt.title('model accuracy')
7 plt.ylabel('accuracy')
8 plt.xlabel('epoch')
9 plt.legend(['train', 'test'], loc='upper left')
10 plt.show()
11 # summarize history for loss
12 plt.plot(history.history['loss'])
13 plt.plot(history.history['val_loss'])
14 plt.title('model loss')
15 plt.ylabel('loss')
16 plt.xlabel('epoch')
17 plt.legend(['train', 'test'], loc='upper left')
18 plt.show()
```



使用狗的品种分类模型模型，训练准确性，由于xception性能一流，直接达到98%以上。验证准确性要低得多（85%左右），表明该模型过于适合训练集，过拟合，但是由于xception的一流。

合理性分析

在与基准模型进行对比，该结果合理，已经预训练过的模型。

Xception是Google出品，属于2017年左右。它在Google家的MobileNet v1之后，MobileNet v2之前。模型在ImageNet等数据集上都取得了相比Inception v3与Resnet-152更好的结果。其模型大小与计算效率相对Inception v3也取得了较大提高。

四、项目结论

该报告，只是根据Udacity学城的已构建框架，对相关流程，CNN搭建，迁移学习的了解等，对整个图像识别和分类做了了解，在未来遇到实际问题时，可以找到解决方案的方向。

此项目的流程为：1、导入数据集。2、检测人脸。3、检测小狗。4、（从头开始）创建分类小狗品种的 CNN。5、（使用迁移学习）使用分类小狗品种的 CNN。6、（使用迁移学习）创建分类小狗品种的 CNN。7、编写算法。8、测试算法

解决方案的整体流程：从项目开始前，回顾了 udacity 的学习过程，了解到对该项目以及对数据科学的理解：一、初步的探索，业务的理解。二、数据准备，预处理，可视化的分析。三、特征选择、数据清洗。四、算法选择，迁移学习。五、模型训练。六、评估和调优。七、合并所有训练数据按最优参数进行训练。八、提交报告或应用。

需要作出的改进

由于对 python和深度学习的不熟悉，代码能力需要进一步的提升，并且可以尝试使用 Pytorch等进行尝试。

项目：<https://github.com/ddhgy/DogBreedClassifier>

最后，感谢 udacity，感谢审阅论文的老师。