

**专业与年级 2017级物联网工程**

**姓 名 黄家名**

**学　　号 1762910211**

**任课教师 周妍**

**成 绩**

**《人工智能原理大作业》**

**系统设计报告**

**（2019—2020学年第一学期）**

**河海大学常州校区 编制**

目录

[一 系统设计背景 3](#_Toc28329630)

[1.1技术背景 3](#_Toc28329631)

[1.2能力基础背景 3](#_Toc28329632)

[1.3 环境配置 3](#_Toc28329633)

[二 功能模块 3](#_Toc28329634)

[2.1选择图片 3](#_Toc28329635)

[2.2预测结果 4](#_Toc28329636)

[三 设计思路 4](#_Toc28329637)

[3.1模型训练 4](#_Toc28329638)

[3.2平台界面 4](#_Toc28329639)

[3.3预测结果 4](#_Toc28329640)

[四 实验过程演示 4](#_Toc28329641)

[4.1模型训练 4](#_Toc28329642)

[4.1.1第一次训练 4](#_Toc28329643)

[4.1.2第二次训练 5](#_Toc28329644)

[4.1.3第三次训练 6](#_Toc28329645)

[4.2平台展示 7](#_Toc28329646)

[4.2.1主界面 7](#_Toc28329647)

[4.2.2图片展示 7](#_Toc28329648)

[4.3结果预测 8](#_Toc28329649)

[六 心得体会 8](#_Toc28329650)

# 一 系统设计背景

## 1.1技术背景

在人工智能发展火热的当下，以卷积神经网络为基础的深度学习的应用不可谓不广泛，主要有两大类，数据预测和图片处理。数据预测自然不需要多说，图片处理主要包含有图像分类，检测，识别，以及分割方面的应用。其中图像分类包含场景分类，目标分类。

Keras 是一个用 Python 编写的高级神经网络 API，它能够以 TensorFlow、 CNTK或者Theano作为后端运行。Keras 的开发重点是支持快速的实验。能够以最小的时延把你的想法转换为实验结果，是做好研究的关键。

## 1.2能力基础背景

学习TensorFlow的基础教程一般都会接触到入门实验——手写数字识别（MNIST）。但当学习完这个实验后就会想着能不能自己去做一个CNN（卷积神经网络）模型来训练自己的图像集呢，于是基于此想法可以通过MNIST延伸加深TensorFlow的学习和理解。网上有很多图像分类的例子，由此选择了做猫狗的图像识别。

## 1.3 环境配置

->安装Python3.6

->根据电脑显卡找到对应版本的CUDA -> 安装CUDA v10.0.130

->根据CUDA找到对应版本的cudnn -> 安装cudnn v7.6.5.32

->根据cudnn找到对应版本的Tensorflow -> 安装Tensorflow-gpu v1.13.1。

->安装keras

**基于以上基础，决定设计一个基于flask框架的pythonweb，通过keras框架训练深度学习算法模型，实现猫狗大作战的小软件系统。**

# 二 功能模块

本系统采用了网页应用的形式，包含两个功能模块：选择图片和预测结果。

## 2.1选择图片

在网页界面上点击选择图片按钮，将图片展示在界面上。

## 2.2预测结果

在网页界面上点击识别图片按钮，将图片的base64传到后台，送入模型获得预测结果并显示。

# 三 设计思路

## 3.1模型训练

决定采用keras框架实现卷积神经网络的模型构建与训练。首先配置好模型训练的环境，安装python、jupyter notebook、cuda、cudnn、keras和tensorflow-gpu的相关版本等。对数据集进行分类分为训练集和验证集，配置神经网络模型，最后进行图片预处理，然后送入模型中进行训练。关注训练集和验证集的准确值和损失值，调整送入模型中的图片数量和训练轮数，防止过拟合。

## 3.2平台界面

使用HTML、CSS、JS等搭建网页的平台界面，完成页面的基本元素布局与样式，按钮的逻辑功能，实现图片的选择与展示、将图片传到后台并获取预测结果，最后将预测结果在网页中显示。

## 3.3预测结果

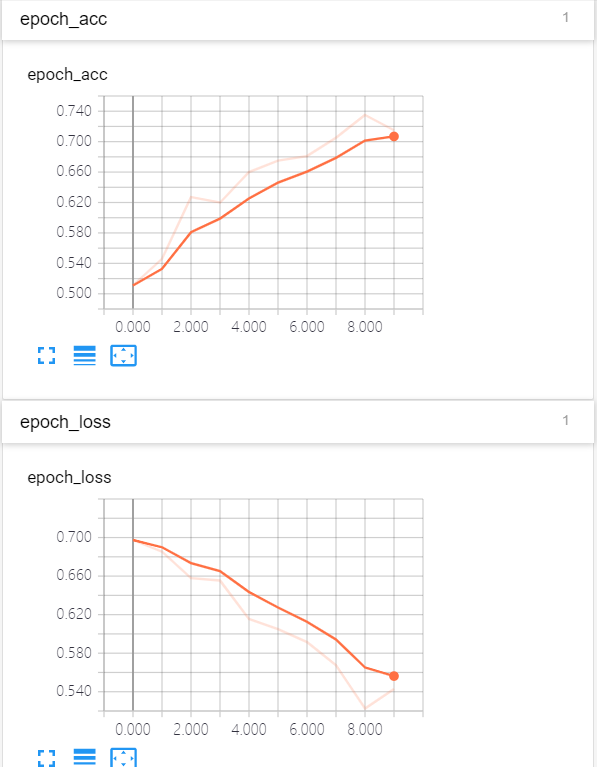
网页后端使用flask框架，前端将选择图片的base64值传给后端，后端将图片的base64值保存成本地图片后，加载模型将图片送入到模型中实现识别结果的预测。

# 四 实验过程演示

## 4.1模型训练

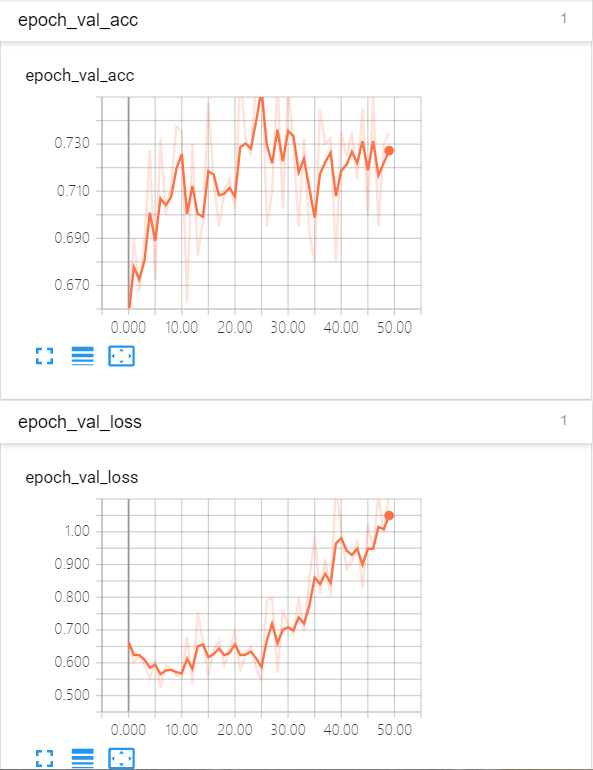
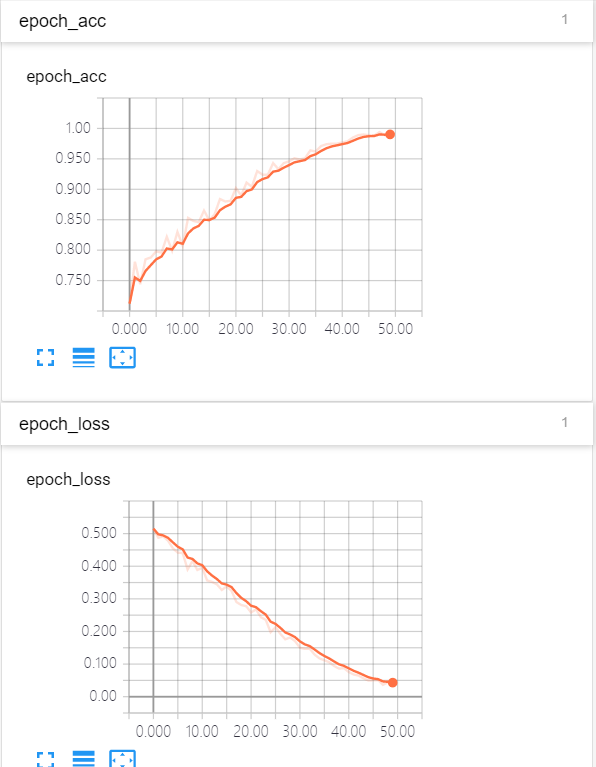
### 4.1.1第一次训练

训练集为1000张猫图和1000张狗图，验证集为500张猫图和500张狗图进行了10轮训练。



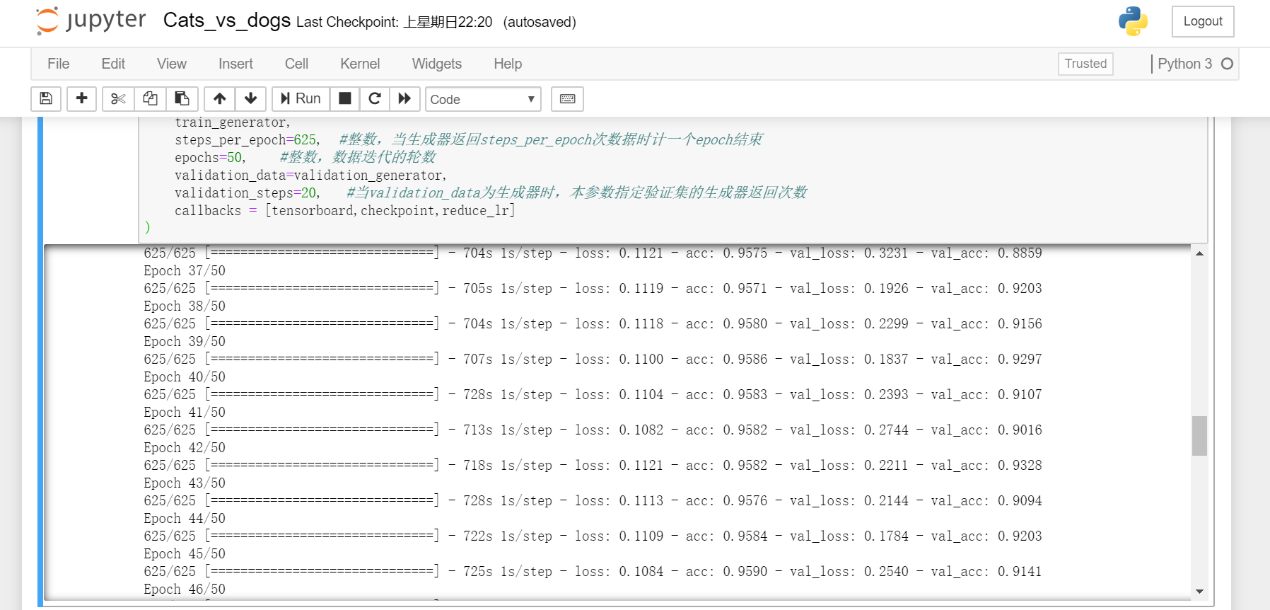
### 4.1.2第二次训练

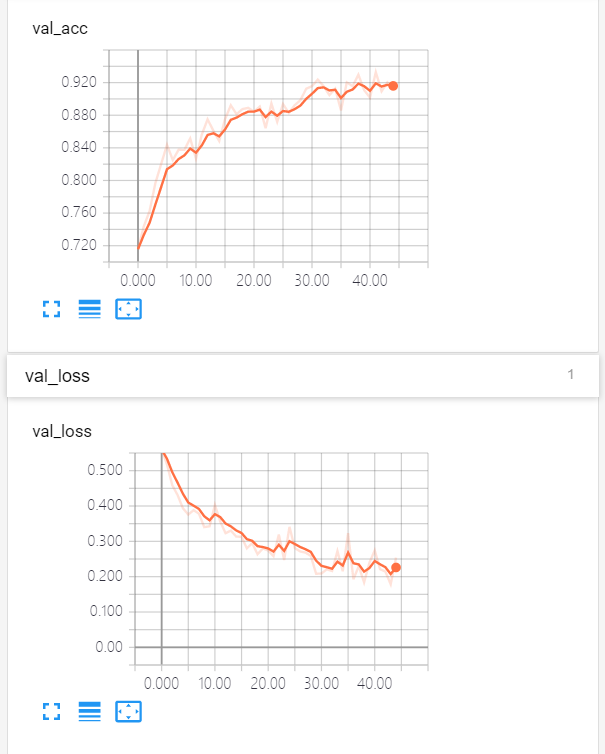
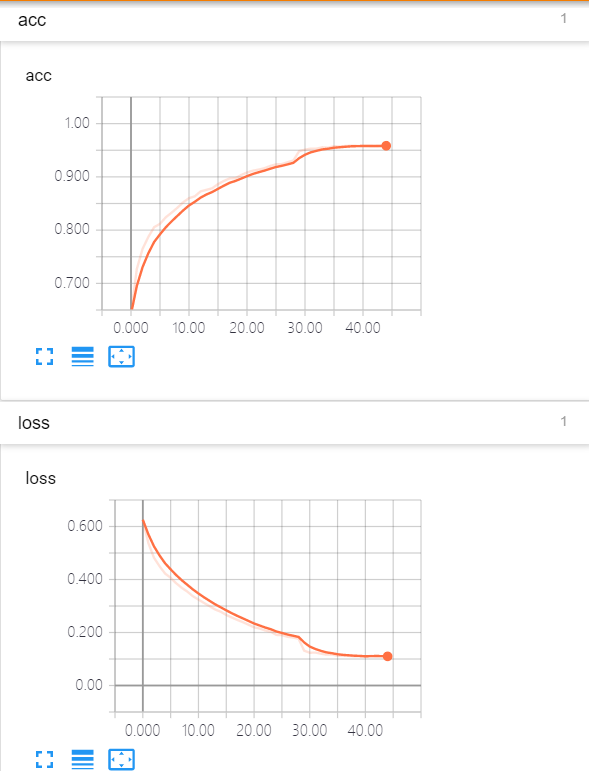
训练集为1000张猫图和1000张狗图，验证集为500张猫图和500张狗图进行了50轮训练，从图中可以看出出现了很严重的过拟合。



### 4.1.3第三次训练

训练集为10000张猫图和10000张狗图，验证集为2500张猫图和2500张狗图进行了50轮训练，在此之前进行了图像处理以增加数据集，同时设置了学习率自动下降的办法，防止过拟合。

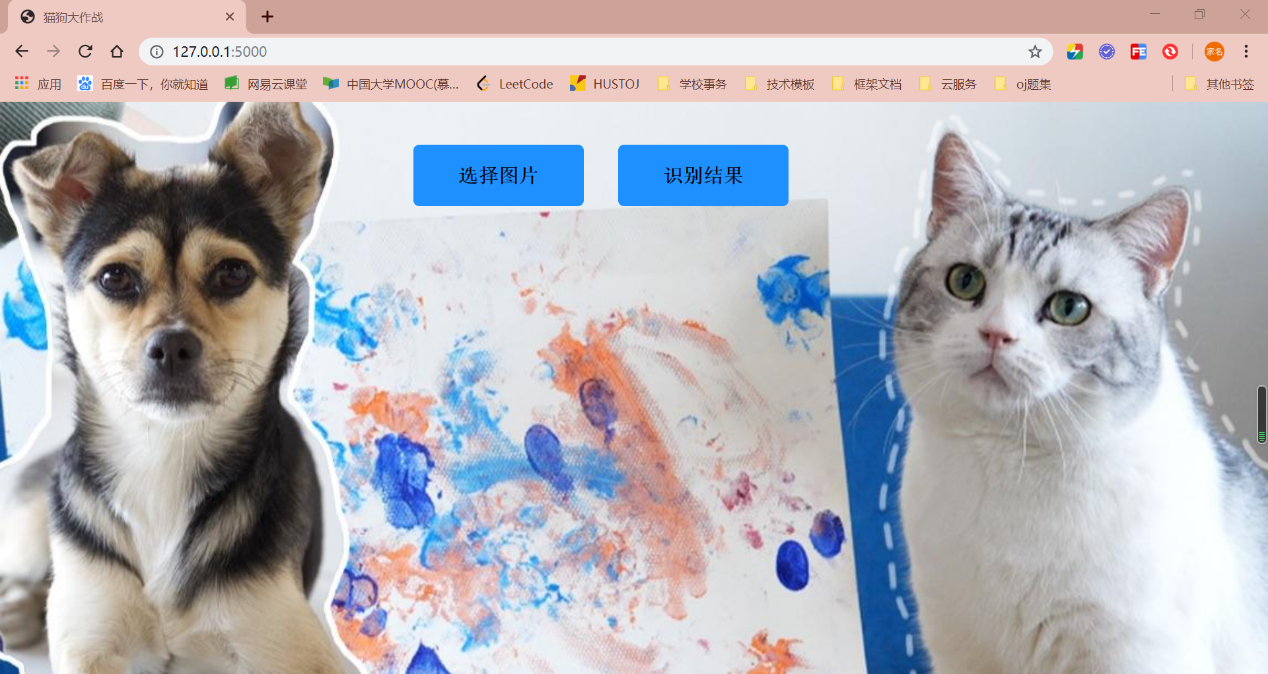




## 4.2平台展示

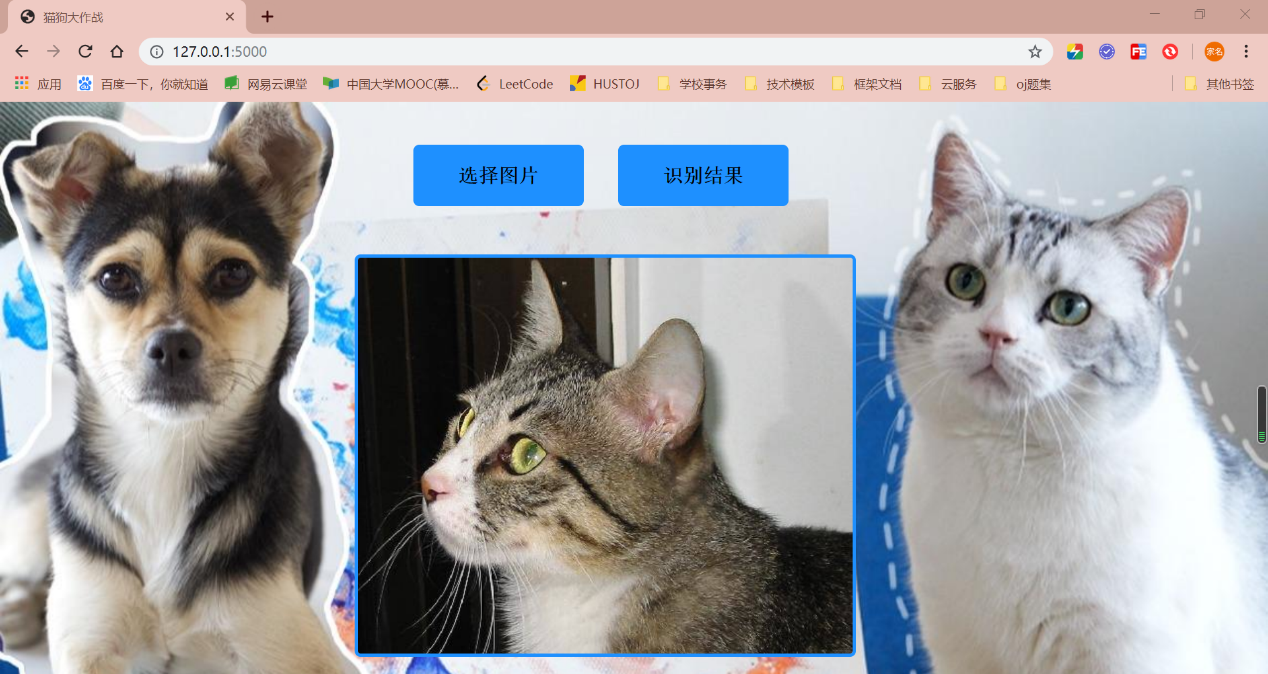
### 4.2.1主界面

主界面包含两个按钮，即选择图片和识别结果。



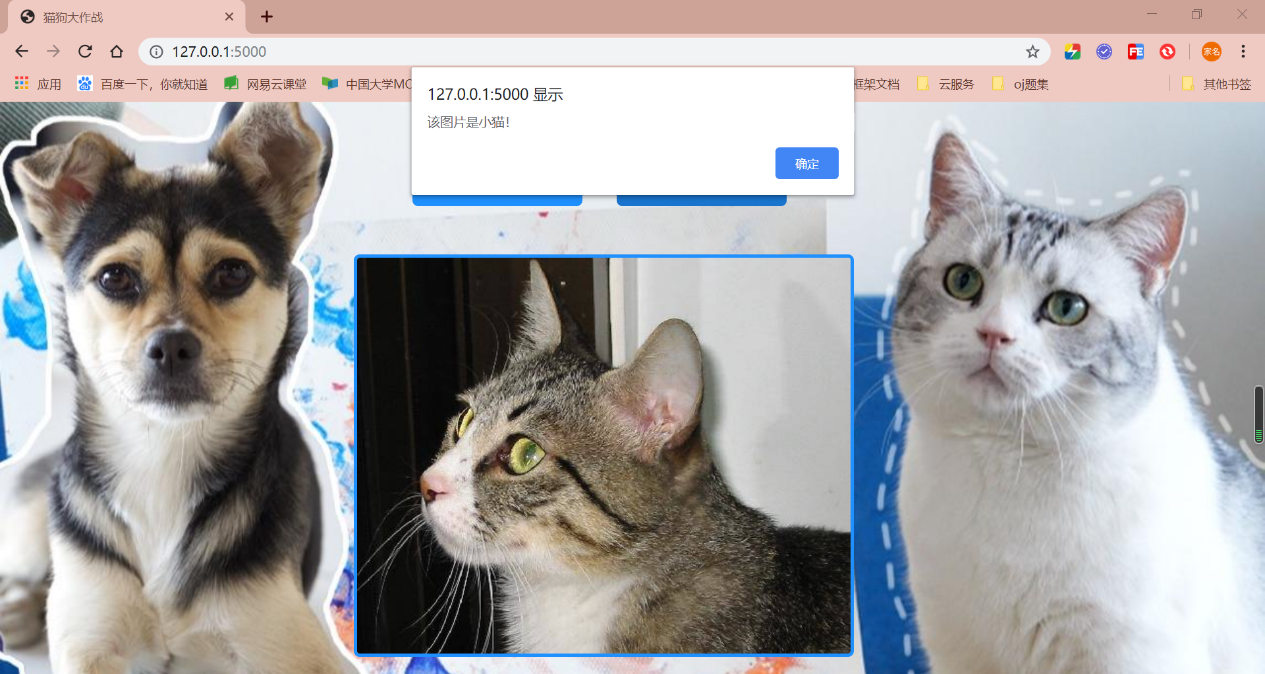
### 4.2.2图片展示

点击选择图片，将待识别的图片展示在网页上。



## 4.3结果预测

点击识别结果，将选择图片的base64值传给后端，后端将图片的base64值保存成本地图片后，加载模型将图片送入到模型中实现识别结果的预测。



# 六 心得体会

1. 环境的配置很关键。因为安装的是tensorflow-gpu，所以花了很长时间安装cuda和cudnn，然后tensorflow和cudnn的版本，cuda和电脑显卡版本都要对应。
2. 合理设计神经网络结构和训练轮次与每次训练的样本数。不然会导致模型过拟合，准确率不高。
3. 学会对数据进行预处理。可以使数据增强，数据集增多，同时实现所有像素值都归一化，方便训练。