# 智能计算系统实验

实验指导书



**安徽工业大学计算机科学与技术学院**

|  |  |
| --- | --- |
| 实验 三 | 基于TensorFlow实现风格迁移 |
| 实验 四 | 基于YOLOv3实现目标检测 |
| 专业班级 | 智能202 |
| 学 号 | 209074458 |
| 姓 名 | 张建才 |
| 指导教师 | 李雪 |

## 摘要：

1： 内含有主要代码。

2： 各个函数的数学表达形式及其实际意义。

3： TensorFlow模型的使用。

4： TensorFlow的函数说明。

5： 模型使用的方法。

## 实验3：基于TensorFlow实现风格迁移

### 实验目的

1. 神经网络的模块化实现；
2. 使用Tensor Flow实现风格迁移；
3. 使用内容损失函数和风格损失函数。

### 背景知识

1. TensorFlow函数：tf.keras.applications.VGG19部分参数说明。

**include\_top**: whether to include the 3 fully-connected layers at the top of the network.;

**weights:** one of None (random initialization), 'ImageNet' (pre-training on ImageNet), or the path to the weights file to be loaded.;

1. 在keras中,要想获取层的输出的各种信息,可以先获取层对象,再通过层对象的属性output或者output\_shape获取层输出的其他特性。def get\_layer(self, name=None, index=None):返回值:层实例(因为每个层都是一个类,所以返回的层本质上是一个类)。
2. 内容损失函数:

黑色的钟表

低可信度描述已自动生成

X是噪声图片的特征矩阵，P是内容图片的特征矩阵。M是P的长\*宽，N是信道数

1. 风格损失函数：

黑色的钟表

低可信度描述已自动生成

M是特征矩阵的长\*宽，N是特征矩阵的信道数。G为噪音图像特征的Gram矩阵，A为风格图片特征的GRAM矩阵。

1. 最终用于训练的损失函数为内容损失和风格损失的加权和：

徽标

描述已自动生成

1. 当训练开始时，我们根据内容图片和噪声，生成一张噪声图片。并将噪声图片喂给网络，计算loss，再根据loss调整噪声图片。将调整后的图片喂给网络，重新计算loss，再调整，再计算…直到达到指定迭代次数，此时，噪声图片已兼具内容图片的内容和风格图片的风格，进行保存即可。

### 实验环境

使用运算硬件：CPU

软件环境：TensorFlow==2.3.1；python==3.7；

使用数据集：imagenet数据集

### 实验步骤

1. 我们使用VGG中的一些层的输出来表示图片的内容特征和风格特征；分离内容特征层和风格特征层的输出，方便后续计算。
2. 将内容图片输入网络。
3. 计算内容损失。
4. 将风格图片输入网络。
5. 计算风格损失。
6. 最终用于训练的损失函数为内容损失和风格损失的加权和。
7. 提取特征层：

图形用户界面, 文本

描述已自动生成

1. 迭代函数：

文本

描述已自动生成

### 实验思考

1. 使用Adma优化器。
2. TensorFlow内含有比较多的函数和参数。在直接使用时，像使用VGG时，我们可以将VGG模型的参数固定，因为我们不需要训练，使用就行。

## 实验二：基于YOLOv3实现目标检测

### 实验目的

1. 使用YOLOv3。
2. 理解目标检测的方法。
3. 检测加上分类。

### 背景介绍

1. YOLOv3:包括划分单元格来做检测、使用Leaky ReLU[27] 作为激活 函数、使用 Batch Normalization[16] 避免过拟合、使用多尺度训练等，通过修改主干网络来提升精度与性能。
2. YOLOv3做目标检测时，首先用卷积神经网络提取特征，然后用非极大值抑制（NMS） 来筛选候选框。
3. 借鉴 Resnet 思想，引入残差（Rsidual）结构。

### 实验环境

硬件环境：CPU

软件环境：python==3.5.2；TensorFlow==2.2.0；N

### 实验步骤

1. Download YOLOv3 weights from [YOLO website](http://pjreddie.com/darknet/yolo/).
2. Convert the Darknet YOLO model to a Keras model.
3. Run YOLO detection.
4. 然后通过convert.py脚本构建模型，并将权重转成Keras版本。
5. 对于每一种class，分别进行非极大值抑制：

文本

描述已自动生成

### 实验思考

* 1. YOLO系列算法可以算是颇负盛名。目标检测相关的算法有不少，而YOLO因其识别速度快而出名，常被用于实时目标检测场景中。
  2. 使用默认定位点。如果使用自己的定位点，则可能需要进行一些更改。
  3. 推理结果与Darknet不完全相同，但差异很小。