

基于 Python 的机器视觉实验教学平台设计

韩志豪¹, 刘晓英²

(1. 浙江国际海运职业技术学院, 浙江 舟山 316000;

2. 浙江大学 舟山海洋研究中心, 浙江 舟山 316000)

摘要: 针对机器视觉领域的学习内容抽象、难以理解、相关的实验教学产品不足这些问题, 基于 Python 语言, 采用开源的 opencv-python 图像处理库与 TensorFlow 机器学习框架, 提出构建机器视觉实验教学平台; 该系统涵盖机器视觉的经典方法, 主要包括向量机、K 临近图像分类, 神经网络、卷积神经网络目标识别, 基于经典方法融合常用函数, 对系统分模块设计; 经过测试, 该系统具有较好的交互性与可扩展性, 可以适应机器视觉的实验要求, 训练数据、样本测试数据导入灵活, 机器视觉参数优化、代码迭代方便, 并且能够编译生成 .exe 可执行文件, 辅助学生学习机器视觉技术的真实应用场景, 提高学生实践解决问题能力和创新能力。

关键词: Python; 实验平台; 机器视觉

Design of Machine Vision Experiment Teaching Platform Based on Python

Han Zhihao¹, Liu Xiaoying²

(1. Zhejiang International Maritime College, Zhoushan 316000, China;

2. Ocean Research Center of Zhoushan, Zhejiang University, Zhoushan 316000, China)

Abstract: The learning content in the field of machine vision is abstract and difficult to understand, The related experimental teaching products are insufficient, Based on the Python language, the open-source openCV-python image processing library and the TensorFlow machine learning framework are used to construct a machine vision experimental teaching platform. The system covers the classical methods of machine vision, including vector machine, K proximity image classification, neural network, convolutional neural network target recognition, fusion of commonly used functions based on classical methods, and system sub-module design. After testing, the system has good interactivity and scalability, can adapt to the experimental requirements of machine vision, training data, sample test data import flexibility, machine vision parameter optimization, code iteration is convenient, and can compile and generate .exe executable. The document assists students in learning the real application scenarios of machine vision technology and improving students' ability to solve problems and innovate.

Keywords: Python ; experimental platform ; machine vision

0 引言

根据世界编程语言排行 TIOBE 统计, Python 荣获“2018 年度编程语言”称号。在统计领域、人工智能编程领域、脚本编写方面、系统测试方面, Python 均排名第一, 另外还在 Web 编程与科学计算方面也处于领先地位。^[1]

机器视觉 (Machine Vision) 是用计算机模拟生物的视觉功能, 在图像中提取、分辨、理解信息。目前机器视觉在人脸识别、同类商品推荐、目标跟踪、产品缺陷监测、医疗诊断、自动驾驶等领域已有非常广泛的应用。同时, 机器视觉也是人工智能机器学习的强有力的分支, 为机器学习的模型分析、数据调试产生重要的影响。

收稿日期: 2019-08-08; **修回日期:** 2019-09-16。

基金项目: 浙江省高等教育学会实验室工作研究项目 (YB201833), 浙江省科技厅重大科技专项 (2014C01018), 舟山市科技局科技项目 (2018C31074), 浙江省教育厅教学改革研究项目 (JG20180855)。

作者简介: 韩志豪 (1991-), 男, 浙江舟山人, 高级工程师, 主要从事电子海图、机器视觉方向的研究。

根据 2019 年 3 月, 教育部公布的 2018 年普通高等本科专业的备案和审批结果^[2], 增设人工智能专业 35 个, 但我国高等院校在实验平台的开发较薄弱, 与专业建设存在脱节的现象。为提高高等院校的机器视觉的专业教育水平, 提升学生的机器视觉的训练、研发、优化能力, 对机器视觉实验平台进行研发, 整合常见机器视觉所使用的模型。

1 机器视觉实验教学平台的总体设计

机器视觉实验教学平台的总体设计框架如图 1 所示。整个机器视觉实验教学平台分为图像分类、图像目标识别两类功能模块, 两个功能模块能满足机器视觉的教学需求。其中图像分类功能模块包括 K 临近 (k-NearestNeighbor, KNN) 图像分类模块、向量机 (support vector machine, SVM) 图像分类模块, 图像目标识别模块包括神经网络 (artificial neural network, ANN) 目标识别模块、卷积神经网络 (convolutional neural networks, CNN) 目标识别模块^[3-4], 涵盖真实产品在机器视觉方面的需求。

在机器视觉实验教学平台程序设计方面, 采用 Python 语言开发。机器视觉实验教学平台的图形用户界面 (graphical user interface, GUI) 设计采用 PyQt5 库, 机器学习

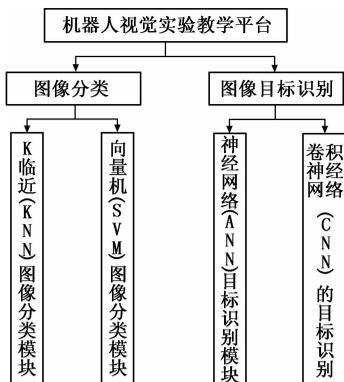


图 1 机器视觉实验教学平台的总体设计框架



图 3 菜单选择



图 4 选择样本目录

(Machine Learning) 采用 TensorFlow 框架, 图像处理部分采用 opencv-python 库。

在界面设计上, 首先让用户通过菜单栏选择以哪个模块进行试验, 此时选择样本目录、训练样本、数据测试、修改参数按钮均不可用。如图 2 所示。



图 2 待用户选择界面

2 机器视觉实验教学平台分块设计

机器视觉实验教学平台采用模块化的设计架构, 将整个系统分成四个模块, 分别为基于 K 临近分类 (KNN) 的图像分类模块、基于向量机 (SVM) 的图像分类模块、基于神经网络 (ANN) 的目标识别模块、基于卷积神经网络 (CNN) 的目标识别模块, 其中图像分类与目标识别为两个数据源, 通过将每个模块的类 (Class) 抽象, 构建两个工厂方法, 方便维护拓展。

2.1 基于 K 临近分类 (KNN) 的图像分类模块

K 临近分类算法, 通俗的讲是找 K 个最近的点, 核心思想是根据最近样本的类别确定样本所属的类别^[5]。使用欧几里德距离公式计算距离, 如式 (1) 所示。

$$d(A, B) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

计算向量点 A 与点 B 之间的距离, 表示 A (x₁, x₂, x₃, ..., x_n) 样本与 B (y₁, y₂, y₃, ..., y_n) 样本在 n 个特征上的距离。

本实验平台有 kaggle 的手写数字数据集, 通过训练可以达到对手写数字的分类。

用户在选择 K 临近分类 (KNN) 的图像分类菜单后, 选择样本目录按钮可用, 此时可以选择样本数据目录, 如图 3、图 4 所示。

用户选择样本目录后, 训练样本、修改参数按钮可用, 就可以根据样本目录文件训练样本。但在每个样本文件的命名中, 有一定的格式要求, 必须要有下划线, 且下划线

前为标签名, 下划线后为该图片的文件序号或文件名。其中标签名也就是最终的分类标签。

在完成样本训练之后, 就可以进行数据测试。当需要优化 K 临近分类 (KNN) 的算法参数时, 可以通过点击修改参数按钮, 链接到代码视图, 修改需要优化的代码及参数。其他几个模块的操作与 K 临近分类 (KNN) 的图像分类模块类似。其中, K 临近分类 (KNN) 的主要训练代码如下所示:

```
import numpy as np
# KNN 分类函数参数说明
# testData—待分类的数据
# trainData—用于训练的数据
# labels—训练数据的分类标签
# k—kNN 参数, 选择距离最小的 k 个点
def KNNclassify(testData, trainData, labels, k):
# 统计 trainData 的行数
trainDataSize = trainData.shape[0]
# 在列向量方向上(横向)重复 testData 共 1 次, 行向量方向上(纵向)重复 testData 共 trainDataSize 次
differMat = np.tile(testData, (trainDataSize, 1)) - trainData
# 二维特征相减后平方
sqdifferMat = differMat * * 2
# sum() 所有元素相加
tDistances = sqdifferMat.sum(axis=1)
# 欧几里德距离公式计算距离
distances = tDistances * * 0.5
# 返回 distances 中元素从小到大排序后的索引值
sortedDistances = distances.argsort()
# 定一个记录类别次数的字典
classNum = {}
for i in range(k):
# 取出前 k 个元素的类别
votedLabel = labels[sortedDistances[i]]
# 计算类别次
classNum[votedLabel] = classNum.get(votedLabel, 0) + 1
# 根据字典的值进行降序排列
```

```
sortedClassNum = sorted(classNum.items(), reverse = True,
key=operator.itemgetter(1))
# 返回出现次数最多的类别
return sortedClassNum[0][0]
# trainFileList 为待训练的文件列表, 由用户选择而来, count
为文件数
count = len(trainFileList)
# mLabels 是存放训练的标签
mLabels = []
# 将待训练矩阵初始化, 1024 是 32 * 32 像素的图像转成—
而来
trainMat = np.zeros((m, 1024))
# 通过下划线找到该样本的标签
for i in range(count):
# 获取文件的名字
fileName = trainFileList[i]
# 获得分类的数字标签
classNumber=int(fileName.split('_')[0])
# 将获得的类别添加到 mLabels 中
mLabels.append(classNumber)
# 将每一个样本的 1x1024 数据存储到已经初始化的 trainMat
矩阵中
trainMat[i, :] = pic2mat(fileName)
# kNNclassify 分类器
neigh=KNNclassify(myTest, trainMat, mLabels, 3)
# myTest 需要测试数据, trainMat 为待训练矩阵, mLabels 为
对应的标签
# 所有元素相加并用欧几里德公式计算距离
distances = tDistances * 0.5
```

用户在 K 临近分类 (KNN) 的图像分类模块, 也可以修改其中源代码, 将欧几里德距离改成马哈顿距离或者余弦距离等。

2.2 基于向量机 (SVM) 的图像分类模块

支持向量机 (SVM) 简单来讲, 去找一个超平面, 使得两种数据类型距离这个超平面的间隔最大化^[6]。两种数据类型, 如果能找到一个函数 $p(x) = \omega^T x + b$, 且 $p(x) > 0$ 表示一种数据; $p(x) < 0$ 表示另外一种数据, 那么就实现了 SVM。如图 5 所示。

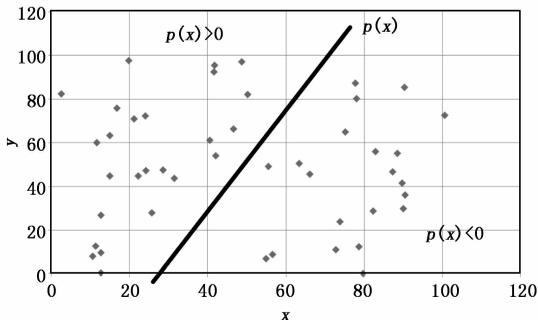


图 5 SVM 线性分割图

线性。因此, 为了让图像或视频样本特征属性可分割, 将特征量映射到多维空间, 进而通过多维切面将特征进行分割。而高维度直接求内积非常困难, 因此需要通过核函数 (kernel function) 将任意两个样本点在扩维后的空间的内积, 等于这两个样本点在原来空间经过一个函数后的输出。

本系统 SVM 可采用径向基核函数 (RBF, Radial Basis Function)、多项式核函数 (poly, Polynomial Kernel)、Sigmoid 核函数^[6]。

径向基核函数公式如下:

$$k(x_1, x_0) = \exp\left(-\frac{\|x_1 - x_0\|^2}{2\sigma^2}\right) \quad (2)$$

多项式核函数公式如下:

$$k(x, y) = (ax'y + c)^d \quad (3)$$

径向基核函数公式如下:

$$k(x, y) = \tanh(ax'y + c) \quad (4)$$

本模块可以对 kaggle 的手写数字数据集进行区分, 操作与 K 临近分类 (KNN) 的图像分类模块类似。

2.3 基于神经网络 (ANN) 的目标识别模块

神经网络是使用多组处理单元 (神经元) 将输入、隐含、输出, 组成一个网状结构, 结构示意图如图 6, 它有自学习、自组织、自适应、容错性、较强的非线性函数逼近的特点^[7]。其中在误差修正上, 一般利用误差反传播 (BP, Back Propagation) 训练算法, 提高权重 (Weight) 与偏差 (Biases) 的精确性^[8]。

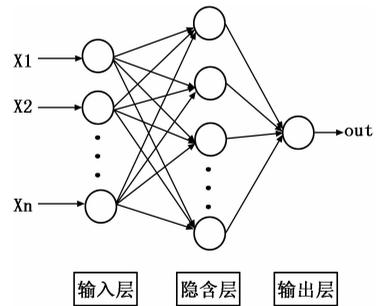


图 6 神经网络结构图

神经网络 (ANN) 在机器视觉方面, 能用于图像特征识别, 也能用于图像分类, 图像处理等, 是机器视觉常用的方法之一, 在本实验教学平台中, 将神经网络用于图像特征识别。

2.4 基于卷积神经网络 (CNN) 的目标识别模块

由于神经网络 (ANN) 的权值太多, 计算非常困难, 因此 Yann Lecun 在 1988 年提出了卷积神经网络 (CNN)^[8]。卷积 (Convolution) 计算的主要方法是通过滤波器 (Filter), 将原始图像去除噪声, 并将特征提取出来, 常用的滤波器为高斯滤波器, 然后通过激活函数, 将数据正则化, 进而通过池化 (Pooling) 将特征图进行压缩, 以减少神经网络权值数^[8]。

在本实验教学平台中, 默认第一层、第二层采用 3×3 的卷积核, 步长为 1, 两端采用全填充方式补零, 使得宽卷

积具有交换性, 激活函数采用 ReLU 激活函数, 如式 (5) 所示。第三层采用最大值池化层, 以 2×2 矩阵筛选区域最大值。第四层使用随机失活, 丢弃部分神经网络, 防止过拟合。第五至八层与第一至四层一致。第九层将多维特征向量压至一维向量, 过渡进入全连接层。第十层采用全连接层, 与神经网络 (ANN) 类似, 但激活函数仍旧用 ReLU 激活函数。第十一层进一步采用随机失活, 随机丢弃一半的神经元, 防止过拟合。第十二层采用全连接的 SoftMax 多分类回归, 其中 SoftMax 函数如式 (6) 所示, 其中 W^T 为权重向量, 1 为全 1 向量。

$$\text{ReLU}(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (5)$$

$$\hat{y} = \text{softmax}(W^T x) = \frac{\exp(W^T x)}{1^T \exp(W^T x)} \quad (6)$$

在卷积神经网络学习过程中, 学习参数为卷积层中的权重与偏差, 与神经网络类似, 卷积神经网络也可以通过误差反向传播训练算法学习参数。

在本实验教学平台中, 将卷积神经网络用于图像目标识别, 测试数据采用 32×32 彩色牛马图, 因此包括长、宽、RGB 值的四维数据。

3 实验测试

基于 Python 的机器视觉实验教学平台在浙江国际海运职业技术学院进行了实验教学, 学生对数据集划分、函数特性、特征发现、过拟合与欠拟合等机器视觉实验问题上有更深入的认知, 大大缩短了他们训练机器视觉的实验周期。

通过 K 临近分类 (KNN) 的图像分类模块, 对 kaggle 的手写数字数据集进行学习, 并优化 KNN 参数后, 其识别准确率最高为 98.73%, 其中识别错误数据如图 7 所示。而通过向量机 (SVM) 的图像分类模块, 学习该数据集后, 并优化参数, 其识别准确率最高为 99.15%, 其中识别错误数据如图 8 所示。在图 7、图 8 中 Res 表示机器视觉的判断结果, True 表示真实结果。

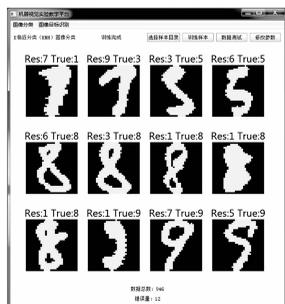


图 7 KNN 分类识别错误数据

kaggle 的手写数字数据集中, 测试数据样本共 946 个, KNN 与 SVM 对各个数字的错误判断情况图 9 所示。

其中 SVM 使用多项式核函数在区分手写数字时准确率最高, 图 10 是径向基核函数、多项式核函数、Sigmoid 核函数结合惩罚系数进行对比, 得出该结论。

通过卷积神经网络 (CNN) 的目标识别模块, 学习牛



图 8 SVM 分类识别错误数据

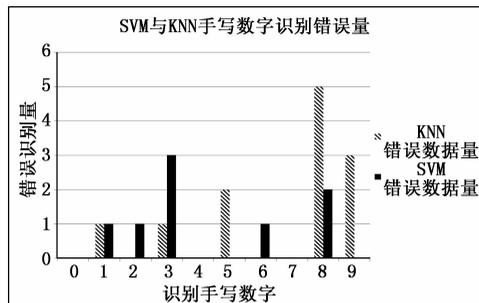


图 9 SVM 与 KNN 手写数字识别错误量

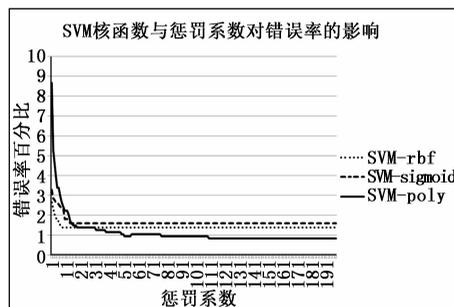


图 10 SVM 核函数与惩罚系数对错误率的影响

与马图片, 实现图片中的牛马识别区分。样本一共 1 000 张牛图、1 000 张马图, 测试数据一共 500 张牛图、500 张马图。初始的卷积神经网络模型如表 1 所示。

表 1 初始的卷积神经网络模型

Layer (type)	Output Shape	Param
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
conv2d_2 (Conv2D)	(None, 30, 30, 32)	9248
max_pooling2d_1	(None, 15, 15, 32)	0
dropout_1 (Dropout)	(None, 15, 15, 32)	0
conv2d_3 (Conv2D)	(None, 15, 15, 64)	18496
conv2d_4 (Conv2D)	(None, 13, 13, 64)	36928
max_pooling2d_2	(None, 6, 6, 64)	0
dropout_2 (Dropout)	(None, 6, 6, 64)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 512)	1180160
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130

经过 100 次的迭代，初始的卷积神经网络最低训练误差值为 0.317 183，最低测试误差值为 0.370 668，最高训练准确率为 86.212 3%，最高测试准确率为 84.342 8%。

对该神经网络结构进行优化，优化后的卷积网络模型如表 2 所示，开始仍旧采用 3 × 3 的卷积核，两端采用全填充方式补零，激活函数采用 ReLU，经过 5 次卷积与激活函数后，第十一层采用最大值池化层，以 2 × 2 矩阵筛选区域最大值；第十二层采用随机失活，丢弃 25% 的神经网络数据。并以上述方式进行三轮卷积，共卷积 15 层，如表 2 的

表 2 优化后的卷积神经网络模型

Layer (type)	Output Shape	Param
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
activation_1 (Activation)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 32)	9248
activation_2 (Activation)	(None, 32, 32, 32)	0
conv2d_3 (Conv2D)	(None, 32, 32, 32)	9248
activation_3 (Activation)	(None, 32, 32, 32)	0
conv2d_4 (Conv2D)	(None, 32, 32, 48)	13872
activation_4 (Activation)	(None, 32, 32, 48)	0
conv2d_5 (Conv2D)	(None, 32, 32, 48)	20784
activation_5 (Activation)	(None, 32, 32, 48)	0
max_pooling2d_1	(None, 16, 16, 48)	0
dropout_1 (Dropout)	(None, 16, 16, 48)	0
conv2d_6 (Conv2D)	(None, 16, 16, 80)	34640
activation_6 (Activation)	(None, 16, 16, 80)	0
conv2d_7 (Conv2D)	(None, 16, 16, 80)	57680
activation_7 (Activation)	(None, 16, 16, 80)	0
conv2d_8 (Conv2D)	(None, 16, 16, 80)	57680
activation_8 (Activation)	(None, 16, 16, 80)	0
conv2d_9 (Conv2D)	(None, 16, 16, 80)	57680
activation_9 (Activation)	(None, 16, 16, 80)	0
conv2d_10 (Conv2D)	(None, 16, 16, 80)	57680
activation_10 (Activation)	(None, 16, 16, 80)	0
max_pooling2d_2	(None, 8, 8, 80)	0
dropout_2 (Dropout)	(None, 8, 8, 80)	0
conv2d_11 (Conv2D)	(None, 8, 8, 128)	92288
activation_11 (Activation)	(None, 8, 8, 128)	0
conv2d_12 (Conv2D)	(None, 8, 8, 128)	147584
activation_12 (Activation)	(None, 8, 8, 128)	0
conv2d_13 (Conv2D)	(None, 8, 8, 128)	147584
activation_13 (Activation)	(None, 8, 8, 128)	0
conv2d_14 (Conv2D)	(None, 8, 8, 128)	147584
activation_14 (Activation)	(None, 8, 8, 128)	0
conv2d_15 (Conv2D)	(None, 8, 8, 128)	147584
activation_15 (Activation)	(None, 8, 8, 128)	0
global_max_pooling2d_1	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 500)	64500
activation_16 (Activation)	(None, 500)	0
dropout_4 (Dropout)	(None, 500)	0
dense_2 (Dense)	(None, 10)	5010
activation_17 (Activation)	(None, 10)	0

conv2d_1 至 dropout_3，但在最后一次池化时，采用全域最大值池化，即 global_max_pooling，最后进行两层全连接层，并使用 SoftMax 进行多分类回归。

对优化后的模型，经过 100 次的迭代，初始的卷积神经网络最低训练误差值为 0.229 262，最低测试误差值为 0.222 37，最高训练准确率为 90.530 3%，最高测试准确率为 91.881 4%。

记录下每次迭代后的误差值与准确率，通过对比优化前与优化后的卷积神经网络，确定优化的方向与关键点。误差分析与准确率分析如图 11、图 12 所示。

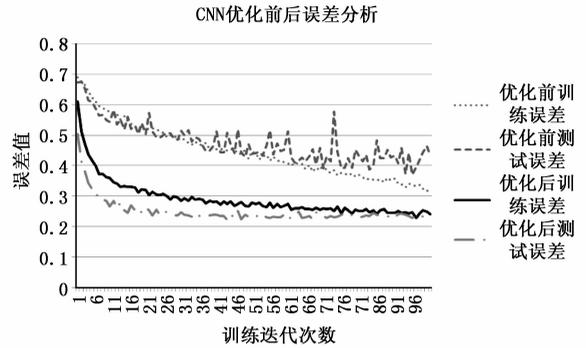


图 11 CNN 优化前后误差分析

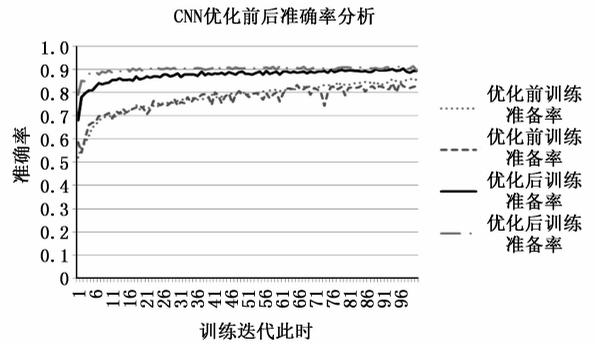


图 12 CNN 优化前后准确率分析

4 结束语

基于 Python 的机器视觉实验教学平台综合了机器视觉常见模型，结合 opencv - python 计算机视觉库与 TensorFlow 机器学习框架可以实现图像分类、识别，并且支持参数修改，为学生在学习机器视觉学科，提供一个交互式、可视化的实验环境，不仅能增强学生对机器视觉理论知识的理解，还能将理论与实际应用结合，解决机器视觉在工业与电子产品中的实际问题，有效提高学生的实践解决问题能力和创新能力，进一步推动国务院发布的《新一代人工智能发展规划》^[9]。

参考文献:

[1] TIOBE Index for September 2018 [EB/OL]. <https://www.tiobe.com/tiobe-index,2018>.