**广西科技大学**

**项目设计报告**

**项目题目： 自动驾驶模拟：**

**基于AirSim无人驾驶平台的研究**

**课程名称： 计算机视觉**

**班 级： 数字媒体技术161班**

**学生姓名： 谢达笙**

**学 号： 201600404003**

**指导老师： 朱亚超**

**完成时间： 2019年6月14号**

**成绩评定：**

目录

[1. 课题介绍 4](#_Toc19628416)

[2. 背景和相关工作 4](#_Toc19628417)

[3. 设计思路 4](#_Toc19628418)

[4. 环境 4](#_Toc19628419)

[5. 制作步骤 6](#_Toc19628420)

[6. 实验和结果 14](#_Toc19628421)

[7. 结论和未来的工作 14](#_Toc19628422)

[8. 收获与鸣谢 14](#_Toc19628423)

[9. 参考文献 14](#_Toc19628424)

## 摘要： AirSim是用于无人机，汽车等的场景模拟器，基于虚幻引擎（Unreal Engine 4）。它是开源的，跨平台的，用于物理和视觉逼真的模拟。本文的目标是描述AirSim平台的搭建使用过程，以实验自动驾驶汽车的深度学习和计算机视觉。

## 关键词：人工智能；自动驾驶；Unreal Engine 4；计算机视觉

**Abstract:** AirSim is a scene simulator for drones, cars, etc., based on the Unreal Engine 4. It is open source, cross-platform, for physical and visual realistic simulation. The goal of this paper is to describe the construction of the AirSim platform to experiment with the deep learning and computer vision of autonomous vehicles.

**Keywords:** artificial intelligence; autonomous driving; Unreal Engine 4; computer vision

# 课题介绍

在1956年, 美国的JohnMcCarthy提出AI之后, 越来越多的应用引入了人工智能这一学科。它是研究用于模拟人的智能的理论、方法、技术及应用系统的一门新的技术科学, 是将人的思想运用于其它方面的重要开发领域。

自动驾驶：自动驾驶又被称为无人车、无人驾驶等 (autopilot, automatic driving, self-driving, driveless) 。对于自动驾驶的概念解释, 业界有着明确的等级划分, 可被它们分为两种模式:一种是NHSTAB (美国高速公路安全管理局) 制定的，一种是SAE International (国际汽车工程师协会) 制定的。【1】

# 背景和相关工作

## 2.1 软件部分

软件部分主要指模拟驾驶的虚拟幻境，这里简单介绍一下AirSim（本文主要使用的环境）和Udacity。

### 2.1.1 AirSim

Airsim是用于无人机，汽车等的场景模拟器，基于虚幻引擎，也支持unity（处于试验阶段，不稳定）。

### 2.1.2 Udacity

优达学城（Udacity）自动驾驶课程中提供的一个开源、类似Airsim的汽车模拟环境。优达学城（Udacity）拥有一整套完整的教程（<https://www.udacity.com/course/self-driving-car-engineer-nanodegree--nd013>）供大家学习使用。

## 2.2 硬件部分

### raspberry pi

看到很多国外的开发者使用树莓派开发无人驾驶模型。但这篇文章未涉及到基于树莓派的无人驾驶。

# 设计思路

搭建airsim—》安装python库—》测试库是否安装完整—》hello\_car测试—》数据探索与准备—》训练模型—》测试模型—》修改完善算法—》实现自动驾驶模拟

# 环境

软件：

OS：win10 专业版

Unreal Engine 4 4.18

Python 3.6.2

OpenCV 4.1.0

Keras 2.0.9

Numpy 1.17.1

TensorFlow 1.14.1

硬件：

CPU: 英特尔 Core i7-6700 @ 3.40GHz 四核

GPU: Nvidia GeForce GTX 1060 3GB

内存: 8 GB

\*如果您没有可用的GPU，则可以在Azure（<https://azuremarketplace.microsoft.com/en-us/marketplace/apps/microsoft-ads.dsvm-deep-learning>）上启动深度学习虚拟机，该虚拟机随附所有安装的依赖项和库（如果您使用此虚拟机，请使用提供的py35环境）。

# 制作步骤

## 5.1搭建环境

### 5.1.1搭建AirSim场景

引擎：想要使用 Unreal Engine，我们需要下载 Epic 开发的 Epic Games Launcher。然后，从 Epic Games Launcher 中再下载所需要版本的 Unreal Engine。

场景：选择 Epic Games Launcher 中左侧的 Learn，然后在右侧的页面中一直下拉找到 Landscape Mountains点进去。这个场景是官方教程使用的。选择 Create Project，然后选择一个路径存下创建工程时，一定要把工程与位置名称改为英文，默认是中文。下载好场景，Unreal Engine 4这一块的任务就基本完成。

\*官方文档特别强调需要使用 4.18 版本，否则可能不成功。低版本自然不行，高版本也可能带来问题。

### 5.1.2 搭建后端环境

克隆airsim：

这里介绍Visual Studio 2017克隆Github上项目的方法。

打开vs2017菜单里的“团队”，点击管理链接，进入右侧的团队资源管理器，克隆本地GIT存储库，第一行输入 https://github.com/Microsoft/AirSim

第二行选择存放地址，点击克隆。

克隆完成后，打开vs的命令行，进入AirSim的路径下，输入build.cmd，进行构建。一段时间后，关闭cmd窗口，打开AirSim的克隆路径下\AirSim\AirLib\deps\eigen3\Eigen\src\Core\arch\CUDA\Half.h 这个文件，找到“AS IS”这个引用符号，将它的改为英文引用符号，保存文件。再以同样的方法运行build.cmd，一段时间后，运行成功，airsim后端构建成功。

配置CUDA：

CUDA是NVIDIA推出的运算平台，CuDNN是专门针对Deep Learning框架设计的一套GPU计算加速方案。

在安装之前要查询下最新TensorFLow发行版支持到了哪个版本。笔者在安装TensorFLow时，CUDA已经到了10.1版本。另外，也要确认CUDA版本是否支持自己的显卡。基于以上两个条件，笔者选择了CUDA10.0，并下载了对应的CuDNN版本。

\*在CPU上运行TensorFlow则跳过此步。

\*相关链接如下：

1）显卡型号支持：https://developer.nvidia.com/cuda-gpus

2）CUDA下载地址：https://developer.nvidia.com/cuda-toolkit-archive

3）CuDNN下载地址：https://developer.nvidia.com/rdp/cudnn-download

安装Python库：

使用Python 3.5或更高版本安装Anaconda。

1. [安装CNTK](https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-CNTK-on-your-machine)或[安装Tensorflow](https://www.tensorflow.org/install/install_windows)（建议在GPU上运行TF）
2. [安装h5py](http://docs.h5py.org/en/latest/build.html)
3. [安装Keras](https://keras.io/#installation)并[配置Keras后端](https://keras.io/backend/)以使用TensorFlow（默认）或CNTK。
4. [安装AzCopy](https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy)。请务必将AzCopy可执行文件的位置添加到系统路径中。
5. 安装其他依赖项。在您的anaconda环境中，以root或管理员身份运行“InstallPackages.py”。

## 5.2 数据探索与准备

**概述**

   我们的目标是训练一个深度学习模型，能够让车辆根据拍摄照片和车辆的最后状态预测车辆行进角度。在这片文章中，我将准备端到端深度学习模型的数据。在此过程中，我们将对数据集做一些有用的监察，这将在训练模型的时候对我们有所帮助。

**端到端深度学习**

   端到端深度学习是一种响应深度神经网络成功的建模策略。不同于传统的方法，这种策略没有建立在特征工程上。相反的，它利用深度神经网络的强大，和近来硬件(GPU，FPGA等)的优势去利用大数据的惊人潜力。它比传统的深度学习跟接近于人类的学习方法，因为它让输入直接映射到输出端。这个方法的最大缺点是它需要大量的训练数据，使得他不适用于许多一般的应用程序。由于仿真软件能够(潜在地)生成无限大的数据，所以它们是端到端深度学习算法的完美数据源。

   自主驾驶是一个可以从端到端的深度学习中获益良多的领域。为了实现SAE标准中4级或5级自主性，汽车需要接受大量数据的训练(汽车制造商每周收集数百pb的数据并不罕见)，如果没有模拟器，这几乎是不可能的。【2】

有了像[AirSim](https://github.com/Microsoft/AirSim)这样逼真的仿真器，现在可以收集大量数据来训练你的自动驾驶模型，而不必使用实际的汽车。然而，这些模型可以使用较少的真是数据进行微调，并用于实际的汽车。这种技术被称作行为克隆。这份文档中，将训练一个模型，以学习只使用汽车上的一个前置摄像头作为视觉输入如何驾驶汽车通过AirSim的一部分景观地图。我们的策略是执行一些基本的数据分析来获得数据集的感觉，然后训练一个模型，通过摄像头获取到的帧和车的当前状态参数(速度、转向角、油门等)预测正确的驾驶控制信号(在本例中是转向角)。

这个数据集包含我们的标签，转向角。它还具有在记录转向角时拍摄的图像的名称。让我们来看一个示例图像 - 'img\_0.png’在’normal\_1’文件夹内。



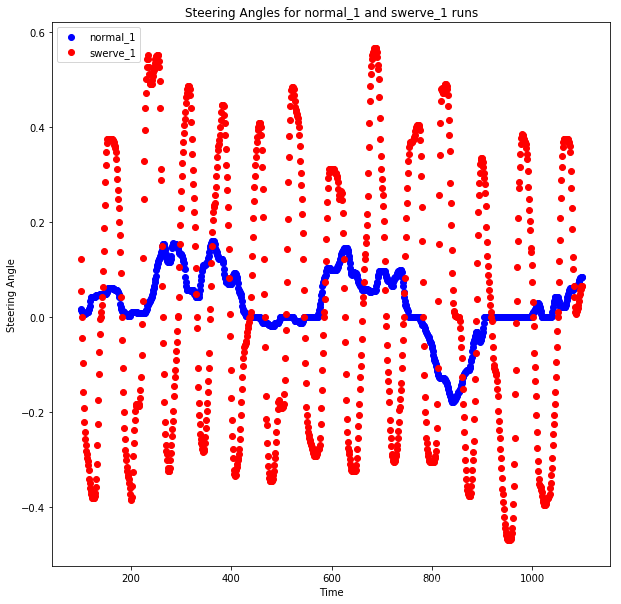
5.2.1（1）

我们可以直接观察到这幅图像只有一小部分是有趣的。例如，我们只需要关注车在下图红色的ROI(region of interest)如何驾驶。



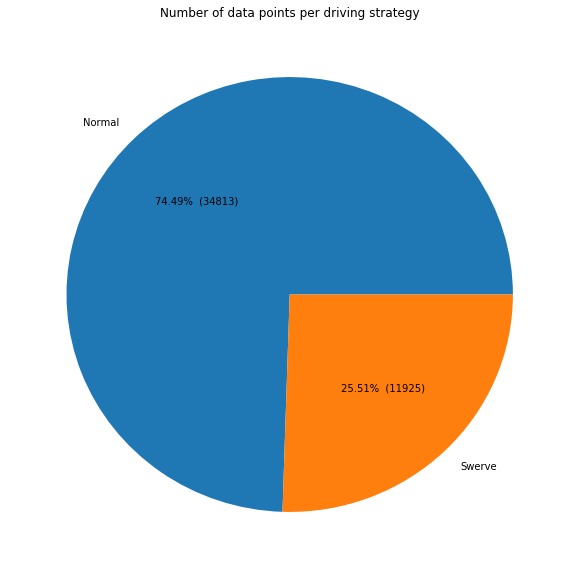
5.2.1（2）

   **提取ROI能减少训练时间和训练模型的数据量**。这也阻止了模型获取不相干环境特征(例如山，树等)而导致的混乱。  
   我们可以做的另一个观察是**数据集显示了垂直翻转公差**。换言之，我们得到一个有效的数据点，如果我们把图像绕Y轴翻转如果也翻转方向盘角度的符号。这很重要，因为它有效地将我们可用的数据点数量翻了一番。  
   此外，**训练后的模型应该对光照条件的变化保持不变**，因此我们就可以通过全局缩放图像的亮度来生成额外的数据点。



5.2.1（3）

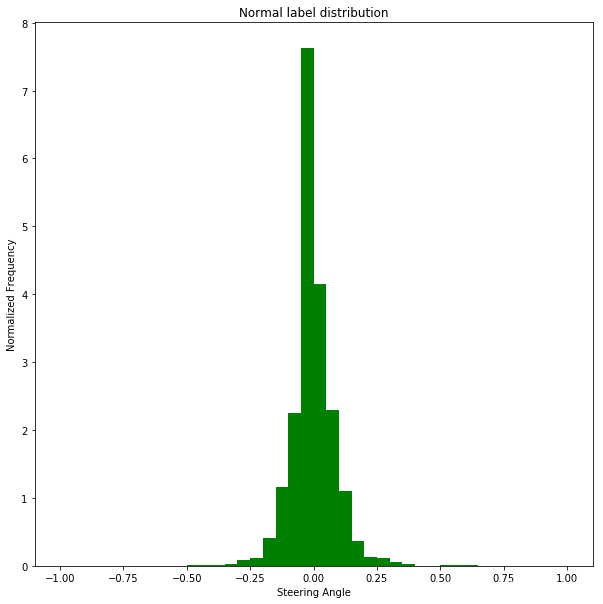
   在这儿我们能够清晰的观察到两种驾驶策略之间的不同。蓝点显示的是正常的驾驶策略，正如你所预料的那样，它会使你的方向盘角度或多或少接近于零，这使得你的车在路上基本上是直行的。  
   转向驾驶策略总是让车经过马路的时候从一边到另一边来回摆动。在训练端到端深度学习模型的时候需要记住这是一个很重要的阐述。由于我们没有进行任何功能工程，所以我们的模型几乎完全依赖于数据集来为它提供在收回期间所需的所有必要信息。因此，要考虑到模型可能遇到的任何急转弯，并让它在开始偏离道路时能够自我纠正，我们需要在训练时为它提供足够的此类例子。因此，我们创建了这些额外的数据集来关注这些场景。完成本教程后，您可以尝试仅使用 ‘normal’ 数据集重新运行所有内容，并观察汽车在路上行驶很长一段时间后失败。

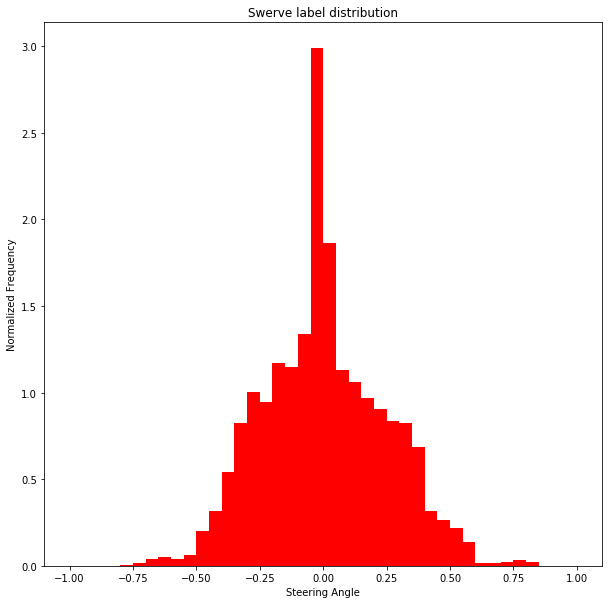


5.2.1（4）

   因此，大约四分之一的数据点是用转向驾驶策略收集的，其余的数据点是用正常策略收集的。我们也看到我们总共有47000个数据点可用。这些数据几乎是不够的，因此我们的网络不能太深。

   让我们看看这两种策略的标签分布情况。



5.2.1（5）  


5.2.1（6）

   我们可以从这些图中观察到一些数据：

* **正常驾驶汽车时，转向角几乎总是零**。有一个严重的不平衡，如果这部分数据没有向下采样，模型总是预测为零，汽车将无法转弯。
* 当使用转向策略驾驶汽车时，我们得到了在常规策略数据集中没有出现的急转弯的例子。**这验证了我们为什么要像前面解释的那样收集数据**。

   此时，我们需要将原始数据合并到适合培训的压缩数据文件中。在这里，我们将使用h5文件，因为这种格式完美支持大型数据集，而不需要一次性将所有内容读入内存。它还与Keras无缝合作。  
   处理数据集的代码很简单，但是很长。完成后，最终的数据集将有4个部分：

* **图像**：包含图像数据的numpy数组
* **状态**：包含汽车的最后已知状态的numpy数组。这是一个(转向，油门，刹车，速度)元组
* **label**：包含我们希望预测的转向角的numpy数组(在-1…1范围内归一化)
* **元数据**：包含文件元数据的numpy数组(它们来自哪个文件夹，等等)

   处理过程将花费一些时间。我们将把所有数据集合并到一起，然后将其拆分为训练/测试/验证数据集。【2】

## 5.3训练模型

现在我们已经对正在处理的数据有了一定的了解，可以开始设计模型了。在这篇文章中，我们定义网络结构并训练模型。我们还将讨论一些对数据的转换，以响应这篇文章的数据探索部分中所做的观察。

  让我们从探索阶段读取数据集。对于图像数据，将整个数据集加载到内存中开销太大。幸运的是，Keras有数据生成器的概念。DataGenerator只不过是一个迭代器，它将以块的形式从磁盘读取数据。这将保证了CPU和GPU的并行运算，增加吞吐量。  
 我们在探索阶段做了一点观察。现在，让我们想出一种能把它们包含进入我们的训练算法的策略：

* **只有一小部分图像是有用的** - 在批量生成时，我们可以删除不感兴趣的图像块。
* **数据集显示垂直翻转公差** - 在批量生成时，我们可以在Y轴上随意翻转一些图像和标签，这样模型就有了新的数据可以学习。
* **数据集应该不受光照变化的影响** - 我们可以从图像中随机添加或删除亮度，这样模型就可以知道应该忽略灯光的全局变化。
* **数据集具有较高的零值图像比例** - 我们可以随机丢弃一定比例的数据点，使转向角为零，这样模型在训练时可以看到一个平衡的数据集。
* **我们需要一些数据集中转向策略的例子，以便模型学习如何快速转向** - 我们在预处理阶段处理过这个。

虽然Keras有一些针对图像的标准内置转换，但对于我们的目的来说，它们还不够。例如，当在标准ImageDataGenerator中使用horizontal\_flip = True时，labels的符号不会倒过来。幸运的是，我们只需要扩展ImageDataGenerator类并实现我们自己的转换逻辑。这部分代码是在Generator.py中完成的——它很简单，但是太长了，不能包含在这篇文章中。  
  在这里，我们使用以下参数初始化生成器：

* **Zero\_Drop\_Percentage: 0.9** - 也就是说，将随机的删除90%的label = 0的数据点
* **Brighten\_Range: 0.4** - 也就是说，将每张图像的亮度修改高达40%。为了计算 “brightness”，我们将图像从RGB转换为HSV空间，向上或向下缩放“V”坐标，然后再转换回RGB空间。
* **ROI: [76,135,0,255]** - 这是一个x1 x2 y1 y2表示的矩形，代表感兴趣的图像区域。

## 5.4 测试模型

在这个文档里，我们会使用我们在5.3中训练的模型在AirSim 中驱动一辆车。我们将对模型的性能进行一些观测，提出一些改进模型的潜在实验的建议。

然后，我们将载入模型，连接AirSim 模拟器到景观图环境。请确保在结束不同进程中*之前*模拟器一直在运行。

我们将设置汽车的初始状态和一些用于从模型存储输出的缓冲。我们将定义一些帮助函数来从AirSim 中读取RGB 图片，然后准备给模型使用。

最后，一个控制框来执行汽车运行。因为我们的模型没有预测速度，我们将试着保持我们的汽车运行在匀速5 m/s。执行下面这个框将使得模型驱动汽车！



5.4.1

# 实验和结果

汽车在道路上跑的很完美，在大部分时刻都保持在正确形式一侧，很谨慎的导航所有大转向时刻和所有它可能偏离道路的可能的时刻。然而，汽车移动很平缓，尤其在那些桥。然后，如果你让模型运行一会儿（超过5分钟一会），你将注意到汽车逐渐随机偏离道路造成碰撞。事实上，能够让汽车学会用非常小的数据集几乎完美地行驶，这件事本身是值得骄傲的！

# 结论和未来的工作

AirSim 打开了一个充满可能性的世界。即使你想训练更复杂的模型或者使用其他的学习策略都不会限制你尝试任何新鲜事情。这里是一些最直接的事情你能够去尝试的，可能需要你修改官方文档中提供的一些代码（包括帮助函数），但是不会要求一些不现实的条件。

未来看似遥不可及，其实已经慢慢进入我们的生活（高速公路上，一特斯拉车主在车里睡了十分钟）。本文只是在电脑上进行软件数据上的模拟仿真与测试，真正实现自动驾驶还有很长的路要走，这个是个多学科的技术，单从人工智能、计算机视觉是无法实现的，未来的工作是结合硬件做成简单的模型经行测试，实现无人驾驶物理模型。

# 收获与鸣谢

这次课程设计让我收获满满。首先，airsim目前为止是最棒的调试环境，美中不足的地方是资料只有官方说明文档，课题研究就跟开荒没什么区别，很痛苦，但成果都是很丰硕，从0到1的质变。然后就是，尽管很难（英文文档，英语教程都没有），但我坚持下来了，算是入门一门新技术。最后就是开拓了自己的视野，在外国的一些网站（比如谷歌、YouTube等）上学到很多前沿科技知识，中国在很多领域都有许多世界级的突破，但是在计算机领域还主要是欧美国家的世界，这是我们要学习的。

\*附上文章代码&数据：

链接：https://pan.baidu.com/s/1v2PHLWxYz9DWV4xoQn-pqg

提取码：zlul

历时一个多月终于学习并完成了这个课程设计，尽管完成的不是很完美，在此，我要感谢朱老师的教导，感谢数媒161选修《计算机视觉》这门课同学长达一个多月的陪伴，感谢贴吧、SCDN博客、交流群给我提供帮助的陌生人。

# 参考文献

[1]周璐雨,陈豪文,宁志豪.基于人工智能的自动驾驶技术[J].计算机产品与流通,2019(05):86.

[2] Autonomous Driving using End-to-End Deep Learning: an AirSim tutorial [EB/OL] <https://github.com/microsoft/AutonomousDrivingCookbook/tree/master/AirSimE2EDeepLearning>