AyeAI Internship Report

-By KUSH Jain

1st to 31st July 2019

**Week 1:**

Task 1: Installation of Ubuntu 18.04 LTS 64-bit

* This task was completed through the installation of VMWare Workstation, a software which allows you to run a virtual machine on your desktop.
* This was the primary Operating System for all the tasks performed throughout the internship
* About 20 - 25 GB of hard disk space was assigned to the virtual machine on which Ubuntu 18.04 was installed through the official ISO file
* Go to the following link, download the setup file for VMWare Workstation and follow the instructions on the screen for installation: https://www.vmware.com/in/products/workstation-player/workstation-player-evaluation.html

# Task 2: Literature review on the various open source autonomous vehicle platforms

* Understood the top 5 platforms used by various companies and wrote their reviews
* Understood the important components of these open source platforms
* Understood the 3 components used by some of the platforms:
  + The neural network model used
  + The datasets used to train the model
  + The framework used for the implementation of the model

The Caffe framework was used for the developing and testing of models in the internship, because it is widely used throughout the autonomous driving industry in various open source platforms, such as Apollo.

Task 3: Installation of Caffe on the virtual machine running on Ubuntu 18.04

* The CPU version was installed
* It is a good practice to run the commands ’sudo apt upgrade’ and ‘sudo apt update’ to regularly update packages in Ubuntu, before starting the Caffe installation.
* There are two methods of installing:
  1. **The build method**: This involves installing all the libraries involved through the ‘sudo apt install’ command first. The following libraries are required:
     + libprotobuf-dev, libleveldb-dev, libsnappy-dev, libopencv-dev, libboost-all-dev, libhdf5-serial-dev, libhdf5-10, libhdf5-dev, libhdf5-cpp-11, libopenblas-dev, libgflags-dev, libgoogle-glog-dev, liblmdb-dev, protobuf-compiler

Apart from this, the packages python-pip, git, scikit-image protobuf and vim were also installed.

After cloning the Caffe repository and making the changes in the Makefile.config according to the system on which Caffe is being installed, we run the 3 make commands: make all, make test and make runtest, for Caffe compilation.

**Link for Reference:** <https://medium.com/@atinesh/caffe-installation-on-ubuntu-18-04-lts-python-2-7-8e8c388ce51f>

* 1. **The package method:** Caffe provides packages which are pre-compiled. They can be installed by ‘sudo apt install caffe-cpu’ for CPU only version or ‘sudo apt install caffe-cuda’ for CUDA version. The dependencies can be installed by ‘sudo apt build-dep caffe-cpu’ or ‘sudo apt build-dep caffe-cuda’ depending on the version.

**Link for Reference:** <http://caffe.berkeleyvision.org/install_apt.html>

**Week 2:**

Task 4: Creating a BASH Script for Caffe Installation

The basics of bash scripting involve licensing, commenting lines to make the script user understand the commands and TODOs, which are pending things to be done.

This script is a package which needs to be installed and run by the user. It requires no other installations from the user’s side, because instructions to install all the libraries and packages are put in the script, making the script self-contained.

**Link for Reference:** <http://linfo.org/create_shell_1.html>

Task 5: Creating a new VM on the same OS (Ubuntu 18.04) for testing of the script

The script was tested for errors on the new Virtual Machine and changes were made accordingly. Various errors were encountered, such as missing libraries and packages, changes required in the Makefile etc. These changes were solved through a Root Cause analysis.

**Week 3**:

Task 6: Literature Review of Project Planning and Agile methods

* Understood the various project working methods used by organizations for efficiency and punctuality of delivering the project:
* Work Breakdown Structure
* DevOps
* Effort Estimation
* Agile Development
* Iterative Development v/s Waterfall Development Method
* Continuous Development / Continuous Integration
* Travis Cl

**Link for Reference:** https://github.com/KushRJ/ayeinterns-summer2019/blob/master/Kush/Review%20on%20Planning%20and%20Agile%20methods.pptx

Task 7: Look for Models and Datasets for autonomous driving

Two extensive autonomous driving datasets found were:

1. Nuscenes Dataset by APTIV ([www.nuscenes.org](http://www.nuscenes.org)): This dataset has various features:

* Full sensor suite (1x LIDAR, 5x RADAR, 6x camera, IMU, GPS)
* 1000 scenes of 20s each
* 1,400,000 camera images
* 390,000 lidar sweeps
* Two diverse cities: Boston and Singapore
* Left versus right hand traffic
* Detailed map information
* Manual annotations for 23 object classes
* 1.4M 3D bounding boxes annotated at 2Hz
* Attributes such as visibility, activity and pose

1. UC Berkeley DeepDrive Dataset (<https://bdd-data.berkeley.edu>): This dataset has various features:
   * Video Data: 100,000 HD video sequences of over 1,100-hour driving experience across many different times in the day, weather conditions, and driving scenarios
   * Road Object Detection: 2D Bounding Boxes annotated on 100,000 images for bus, traffic light, traffic sign, person, bike, truck, motor, car, train, and rider
   * Instance Segmentation: Explore over 10,000 diverse images with pixel-level and rich instance-level annotations
   * Driveable Area: Learn complicated drivable decision from 100,000 images
   * Lane Markings: Lane marking annotations on 100,000 images

Task 8: Running an Apollo model:

This task was not successful due to the errors in cloning the Apollo GitHub repository, so another model was used which is described below

The errors were faced in the cloning of the repository due to a bad network connection

Task 9: Running a simple model in Caffe:

* Went through basic tutorials on how to run a model in Caffe
* Ran the script provided for running an ImageNet model
* Various errors were encountered while running the script, which were solved through a Root Cause Analysis of each problem
* Enabled source code in the VM to execute the script for running the model

**Week 4:**

Task 9: Testing an image on the trained ImageNet model:

Installed and used gimp editor to make the test image have the same properties as the training dataset images and tested it on the model

Task 10: Train and test Lane Detection and Traffic Light Detection models in Caffe:

The current task is in process.