Setting up compute environment for RVSS

Conda is a package management for Windows Mac and Linux. It is focused on python libraries, but it manages also packages in C/C++ and FORTRAN.
Nothing that can be done with conda cannot be achieved otherwise. However, with conda it is usually easier and cross-platform.
Conda handles dependencies seamlessly and makes it easy to set up different environments with different versions of libraries. And if an environment is ruined beyond repair, you can just remove it and start over with a clean one.

Guide for setting up a Conda environment.

```
## This is a brief how-to set up conda and basic libraries on your computer.

# indicates a comment.
## The parts not necessary at first are in blue:
## The parts strictly necessary to set up necessary libraries are in red:

## INSTALL CONDA
##
## go to https://conda.io/miniconda.html
Select the right Python 3 version appropriate to your machine (e.g Windows/Mac/Linux)

If asked for advanced options.  Do not select “add anaconda to the system path environment” or “register anaconda as the system Python 3.7”.

## windows:
## install the .exe
## Once installed on the "start" menu open the "anaconda prompt"

## linux and mac:
bash Miniconda3-*.sh

    # To know more about conda:
     #   https://conda.io/docs/user-guide/tasks/manage-environments.html

## to see your environments
conda env list

## remove an environment:
conda remove --name FAILED_ENVIRONMENT --all

## you should have just the base environment present at this point
## it is best not to use the base environment to install packages, rather you should make new environments for each major activity and install the required packages locally in order that you can delete a failed environment.

#####
## create a new conda environment for RVSS tutorials:
conda create --name rvss2019 python=3
```

```
## activate the environment you just created:
conda activate rvss2019

## when you are finished in an environment you can leave by
conda deactivate rvss2019

## make sure you have rvss_tutorials activated - check in the prompt (rvss_tutorials)

## Install packages needed for the tutorials inside the environment! Conda will sort out all the dependencies.
## install the jupyter notebook package.
conda install jupyter

## If asked to give permission to install files - say yes.

## install scipy and numpy for linear algebra libraries
conda install scipy numpy

## install pytorch and torchvision for implementation of learning algorithms and associated image processing modules .  Here we need to specify the channel since pytorch and torchvision don’t live in the standard conda universe.
conda install pytorch=0.4.1 torchvision -c pytorch

## install opencv
conda install opencv

## now lets start a jupyter notebook:
jupyter notebook

# click on "new" -> "python 3"

## for example, add in a cell:
a = 10
## "shift-Enter" will run the cell and moves to the next cell - creating a new cell if there isn’t one
## Add another line of code:
print('Hello world, the value of "a" is %d:' % (a))
## "ctl-Enter" will run the cell and keep that cell highlighted.

## to use a function from a package - for example computing the sin function using scipy
## First import the package in the notebook
import scipy, numpy
## shift-Enter
scipy.sin(2*numpy.pi)
## ctl-Enter

##############################
## If you want to get a much deeper introduction, some online tutorials that may be worth having a look at are

### Numpy quickstart:
# https://docs.scipy.org/doc/numpy-1.15.1/user/quickstart.html
```

### Pytorch quickstart:
## https://cs230-stanford.github.io/pytorch-getting-started.html

If you want to get ahead have a look at the following tutorial on PyTorch.

    - https://pytorch.org/tutorials/

Robotic Vision Summer School Workshop

Introduction

The workshop for this year aims at giving you the opportunity to deploy a deep neural network on a mobile robot. You will go through all the stages of this process: data collection and training data labelling, network architecture design and implementation, training and testing, deployment on the robot and task execution.

The task you are going to implement is an autonomous steering of a robot to follow a track. The input to the system is an RGB image from a camera onboard the robot and the output is a steering command to keep the robot on track.

The workshop task can be broken down into a number of sub-tasks, as shown below.

- Using your laptop to control the robot
- Collecting data with the robot
- Building a network
- Training your network
- Testing and refining your network

Visit the workshop website here: https://sites.google.com/view/rvss2019ws/overview

Setting up before you arrive

Setup the rvss2019 conda environment as described above.

- Alternatively, if you are setting up your own Python environment you will need:
    - Python 3.5 or above
    - Pytorch version 0.4 (not the latest 1.0) (select pip as your package and none for cuda version)
    - Scipy and numpy (use pip instructions)
    - OpenCV3 for python
    - torchvision
    - TensorboardX (optional for visualization during training only)

You will be building the neural network and training it on your laptop then perform the inference stage on the robot (no training on the robot).