**MSD COOKBOOK 23-24**

**Table of Contents:**

1. Introduction

2. Requirements

3. Python Libraries Required

4. Steps to get started

a. Python

b. App Installation

c. Resolve Errors

**Introduction:**

In recent years, the escalating rate of vehicle thefts has emerged as a significant societal concern, posing economic losses and public safety threats on multiple fronts. This problem is particularly pronounced in urban areas, where dense populations and extensive surveillance systems coexist. Law enforcement agencies grapple with the daunting task of managing the overwhelming volume of surveillance data generated by Closed-Circuit Television (CCTV) systems. As vehicles go missing or are stolen, investigators face the daunting challenge of swiftly and accurately identifying, locating, and recovering these missing assets.

**Requirements:**

● Python 3.8 and above

**Python Libraries Required:**

The libraries can be installed in a Python virtual environment using the provided ***shell script file***.

The required libraries are:

● PyQt5

● lxml

● bcrypt

● psutil

● python-dateutil

● qt-material

● qtwidgets

● opencv-python

● numpy

● enchant

**Steps to get started:**

**1. Python**

Python can be downloaded and installed from here for Windows-based systems. In most cases it is already installed on Linux distros. If not:

**Python installation steps for Ubuntu:**

1. Open a terminal.

2. Update the packages list and install the packages necessary to build Python:

| $ sudo apt update  $ sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev libsqlite3-dev wget libbz2-dev |
| --- |

3. Download the latest release’s source code from the Python download page using wget :

| $ wget  https://www.python.org/ftp/python/3.8.12/Python-3.8.12.tgz |
| --- |

4. When the download finishes, extract the gzipped archive :

| $ tar -xf Python-3.8.12.tgz |
| --- |

5. Switch to the Python source directory and execute the configure script which performs a number of checks to make sure all of the dependencies on your system are present:

| $ cd Python-3.8.12  $ ./configure --enable-optimizations |
| --- |

6. Start the Python 3.8 build process:

| $ make |
| --- |

7. When the build process is complete, install the Python binaries by typing:

| $ sudo make altinstall |
| --- |

8. That’s it. Python 3.8 has been installed and ready to be used. Verify it by typing:

| $ python3.8 --version |
| --- |

9. The output should show the Python version: Python 3.8.12 4

**Python installation steps for CentOS:**

1. Open a terminal.

2. Install Python Dependencies

| sudo yum -y install epel-release  sudo yum -y update |
| --- |

3. Reboot after the upgrade

| sudo reboot |
| --- |

4. After the reboot, install build dependencies

| sudo yum -y groupinstall "Development Tools"  sudo yum -y install openssl-devel bzip2-devel libffi-devel xz-devel |
| --- |

5. Confirm gcc is available:

| gcc --version |
| --- |

6. Download latest Python 3.8 Archive

| sudo yum -y install wget  wget  https://www.python.org/ftp/python/3.8.12/Python-3.8.12.tgz |
| --- |

7. Extract the package.

| tar xvf Python-3.8.12.tgz |
| --- |

8. Change the created directory:

| cd Python-3.8\*/ |
| --- |

9. Setup installation by running the configure script.

5

| ./configure --enable-optimizations |
| --- |

10.Initiate compilation of Python 3.8 on CentOS 7.

| sudo make altinstall |
| --- |

11. If this was successful, you should get a message like below:

*Installing collected packages: setuptools, pip*

*Successfully installed pip-19.2.3 setuptools-41.2.0*

12.Confirm that the installation of Python 3.8 on CentOS 8 / CentOS 7 was successful.

| python3.8 --version |
| --- |

13.The output should show the Python version: Python 3.8.12

**2. App Installation**

We recommend creating a virtual environment to install the application. 1. Download ZIP file of code from this [link](https://drive.google.com/file/d/1bFM5wiZeImBquECUU2aW_1gqL2swz5pM/view?usp=drive_link).

2. Extract the downloaded ZIP file.

3. Open a terminal.

4. Install Python Virtual Environments if not installed.

| $ sudo apt install -y python3-venv |
| --- |

5. Create a virtual environment

$ python3 -m venv **msd\_env**

where **msd\_env** is the environment name

6. Activate the virtual environment

6

| $ source msd\_env/bin/activate |
| --- |

Your command prompt will now be prefixed with the name of your environment, in this case it is called **msd\_env**.

| Example:  (msd\_env) user@ubuntu:~/TIFR$ |
| --- |

7. Navigate to the directory where the application has been cloned. 8. Run the following command to install all requirements:

| $ bash ./ubuntu-install.sh **(for ubuntu)**  $ bash ./centos-install.sh **(for centos)** |
| --- |

9. Navigate to the directory “Inference+Library”.

10.Launch the application by executing the following command in the application's root directory:

| $ python3 main.py |
| --- |

**3. Resolve Errors**

If you get an error regarding protobuf, then try:

| $ pip3 install protobuf==3.20.\* |
| --- |

If you are unable to see the icons in Training gui, then:

1. Go to Annotation+Training/src/TrainingWindow.py

2. In \_createToolBarActions(), replace the image path with absolute image path.