# Mobile Camera Application to Monitor Residential Society Vehicle Activity with Python Setup Instructions

## 1. Introduction

Objective:

The objective of this project is to develop an Automatic Number Plate Recognition (ANPR) system using machine learning models integrated with a Django web application. The system is designed to identify vehicle number plates and classify vehicles as either residents or non-residents.

Motivation:

The ANPR system enhances security by automating vehicle identification and access control in residential areas, facilitating efficient monitoring and improving community safety.

## 2. Project Setup

Python Version:

The project requires Python 3.8.5 due to specific library compatibility with Django and machine learning dependencies. If Python 3.8.5 is not automatically selected, follow the instructions below to set it up manually.

## 3. Configuring Python 3.8.5 in Visual Studio Code

1. Install Python 3.8.5

Start by downloading and installing Python 3.8.5 on your system.

2. Check Python Version in VS Code

- Open Visual Studio Code, press Ctrl + Shift + P, and select 'Python: Select Interpreter.'

- Look for Python 3.8.5 in the list and select it.

- Open a terminal in VS Code and type python --version to confirm that Python 3.8.5 is active.

3. Set Up Environment Variables (If Required)

If Python 3.8.5 does not appear in the VS Code interpreter list, follow these steps:

- Open 'Edit the system environment variables' on your computer.

- Select 'Environment Variables' and locate the 'Path' variable under 'System Variables.'

- Click 'Edit' and add the path where Python 3.8.5 is installed (e.g., C:\Python38).

- Restart VS Code and repeat the steps above to confirm that Python 3.8.5 is now available.

## 4. System Architecture and Workflow

The ANPR system follows a structured workflow to achieve its goal of automated vehicle identification and classification. The workflow consists of the following steps:

1. Step 1: Capture an image or video of vehicles entering the area.

2. Step 2: Detect license plates in the images or video frames using OpenCV.

3. Step 3: Extract text from the detected license plates using Tesseract OCR.

4. Step 4: Cross-reference the extracted data with a resident database to classify vehicles as either resident or non-resident.

## 5. Step-by-Step Implementation

1. Extract Project Folder

Extract the provided project folder and navigate to the main project directory.

2. Install Required Libraries

Run the following command in your terminal to install the necessary libraries:

pip install Django==3.1.4 numpy==1.18.5 opencv-python==4.4.0.46 pytesseract==0.3.7 scikit-image==0.17.2 Keras==2.4.3 tensorflow==2.3.1

3. Run the Django Project

- Migrate Database: Ensure that the database is set up by applying migrations:

python manage.py migrate

- Start Django Server: Once migrations are complete, start the Django server:

python manage.py runserver

- Access the application by navigating to http://127.0.0.1:8000 in your web browser.

4. Email Notifications

Configure email notifications by setting up SMTP credentials in settings.py:

EMAIL\_HOST\_USER = 'your-email@gmail.com'

EMAIL\_HOST\_PASSWORD = 'your-email-password'

This will enable the system to send notifications for non-resident vehicle entries.

## 6. Testing and Troubleshooting

Testing:

- Number Plate Detection: Test with sample images or video to ensure accurate license plate detection.

- Resident/Non-Resident Classification: Confirm the system correctly identifies and classifies vehicles as resident or non-resident.

- Email Notifications: Verify that email notifications are sent when a non-resident vehicle is detected.

Troubleshooting:

- Library Compatibility Issues: Ensure that all specified library versions are installed and compatible with Python 3.8.5.

- Email Configuration Issues: Double-check SMTP settings if email notifications are not functioning correctly.

## 7. Results and Conclusion

Outcome:

The ANPR system successfully automates the identification and classification of vehicles in real time, providing a robust and efficient method for managing vehicle access in residential areas. The system met project requirements for speed, accuracy, and scalability, enhancing community safety through automated security protocols.

Future Enhancements:

Potential improvements include expanding the training data set for increased accuracy, integrating real-time alert functionalities, and extending the system’s capabilities to support additional vehicle types and larger areas.