

CV Pro – AI Kit  
to build your  
own Self-  
Driving Car

CV Pro – The AI kit  
for Autonomous  
Navigation

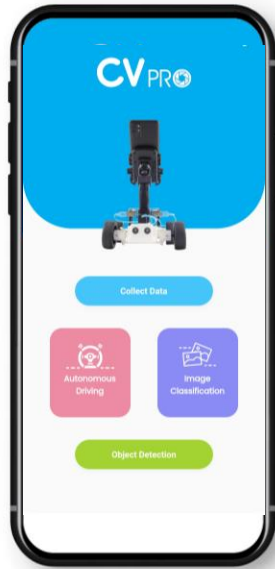


# Introduction to CV Pro

CV Pro is an AI-powered kit equipped with features such as:

- ❖ Simulate fully functional Self Driving Car using Computer Vision
- ❖ Achieve autonomous car driving in physical environment
- ❖ State of the art communication channels using smartphones
- ❖ Guided introduction to industry standard tools like OpenCV, Deep Learning, TensorFlow
- ❖ Learn to apply Computer Vision and deep learning techniques to identify lanes on a road





### Compatible Versions (App's)

- Android - 10 & above
- iOS - 11 & above

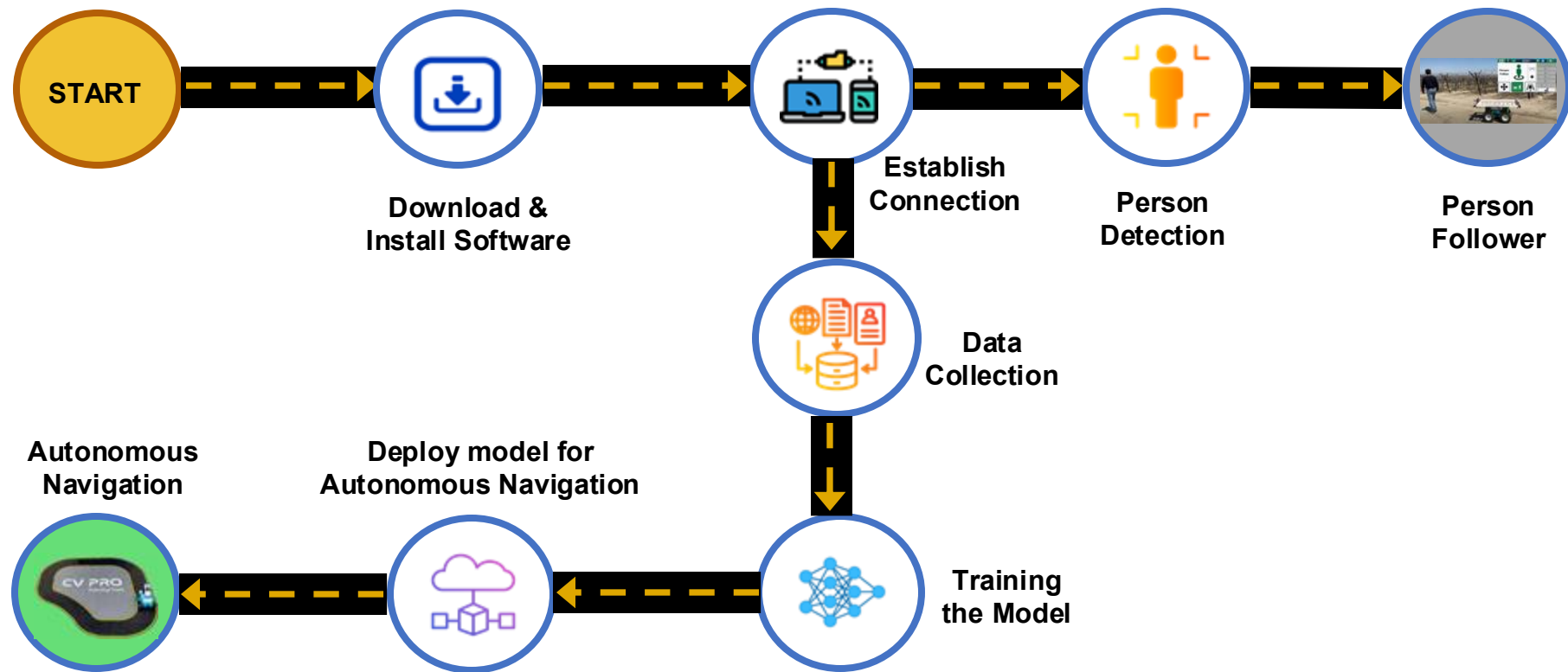


### Compatible Versions (Platforms)

- Windows OS - 10, 11
- Ubuntu OS - 21 & above
- macOS - 14 & above



# Software Configuration



# Downloading & Installing Software

Once the CV Pro kit has been assembled, it is imperative to download and install the requisite software in your computer and smartphone, to enable seamless operation with the kit. This step is vital for tasks such as data collection, training the machine learning model, and autonomous navigation of the CV Pro kit. Please adhere to the instructions provided to ensure the successful implementation of the CV Pro kit.





## Here is an overview of the process to be followed:

1. Acquire the '**Meritus-CVPro**' GitHub resource package by downloading it.
2. Establish '**Miniconda**' package in the default user directory.
3. Install the '**Mosquitto broker package**' in C:\Program Files, for Windows OS and in 'User Profile' path for the other operating systems. Refer to figure (1).

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\robot\Meritus-CVPRO-Windows>cvpro

Pre-Requisites:
-----
1. Download the Meritus-CVPRO from https://github.com/robotixdevteam/Meritus-CVPRO/tree/Windows and extract the same to the User-Profile Path
2. Miniconda should be installed in the User-Profile Path
3. Mosquitto should be installed in C:\Program Files

Hierarchy of Execution:
-----
Type 'my_conda' - Create or Activate the Conda Environment
Type 'install_cvpro' - Installation of required libraries
-----
Please ensure to connect the Bot with the System
Type 'launch_server' - Launch the MQTT Server
-----
Type 'run_cvpro' to move the Bot around for Data-Collection Process.
Type 'train_cvpro' to train the Bot for Autonomous Process.

C:\Users\robot\Meritus-CVPRO-Windows>
```

Figure 1

## Downloading files from GitHub

**Download Resource Package from the link provided below:** This is required to obtain and utilize relevant Python packages, for execution of the processes (data collection and model training).

- ❖ Visit <https://github.com/robotixdevteam/Meritus-CVPRO>
- ❖ Select the download resource based on your operating system. Click on the branch tree and choose the resource to download depending on your operating system.
- ❖ If you're using Windows, select the Windows download option, and for Ubuntu and macOS, choose the respective options. You can refer to figure (2).

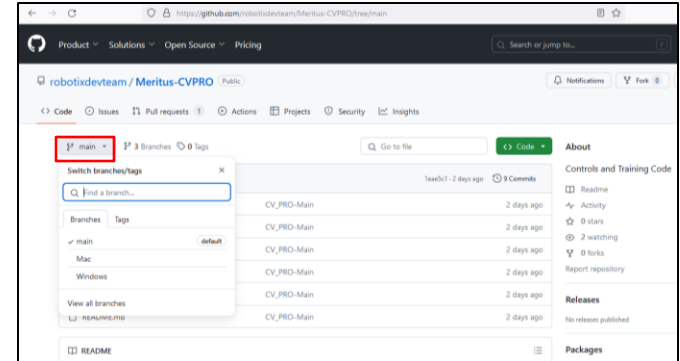


Figure 2

- ❖ So far, we've successfully downloaded the GitHub resource repository and incorporated it into our system.
- ❖ Next, we shall move on to installing the necessary software.
- ❖ To start, let us install '**Miniconda**' software.

## Installing the Miniconda software

**Objective of Miniconda** – This is required to control the bot and create machine learning model using the collected data.

- ❖ Visit the download link - <https://repo.anaconda.com/miniconda/>
- ❖ Select the Miniconda distribution package suitable for your operating system (e.g., *Windows, macOS, Linux*). Refer to figure (3).



Figure 3

## Installing the Miniconda software

- After the download is complete, locate the downloaded file and execute the installation process.
- The package should be installed only in the 'User Profile' path, irrespective of your operating system; whether it is Windows, Ubuntu or macOS.









## MQTT Broker Protocol Installation for Windows OS

- ❖ After installing the Miniconda software, our next step is to install the '**MQTT Broker protocol**' package.
- ❖ The MQTT Broker Protocol is essential for facilitating effective communication among various devices and clients.
- ❖ To install, it please visit the link provided in the subsequent slides for downloading. Ensure to select the appropriate package compatible with your operating system.

## MQTT Broker Protocol Installation for Windows OS

1. If you're using a Windows 64-bit OS, obtain the MQTT (Message Queuing Telemetry Transport) Broker installer suitable for Windows 64-bit from the provided link: [Windows 64-bit OS](#).
2. If your operating system is Windows 32-bit, download the MQTT Broker installer from this link: [Windows 32-bit OS](#).
3. After completing the download, find the installer file, '**mosquitto-2.0.15-install-windows-x64.exe**' from your downloaded location.
4. The package should be installed only in '**C:\Program Files**' and in no other location.



mosquitto-2.0.15-  
install-windows-x  
64.exe



## Mosquitto MQTT Broker installation for Ubuntu OS

- ❖ Open the Terminal Application
  - Begin by launching the terminal application.
- ❖ Update the Repository and Package List. Enter the following commands, in sequence:
  - **'sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa'**
  - **'sudo apt-get update'**
- ❖ Installation - After adding the Mosquitto repository and updating your package lists. Navigate to your **user profile path** and install Mosquitto broker with the following command: (Do not install in any other directory, apart from your User Profile path'.)
- ❖ **'sudo apt-get install mosquitto'**



## Mosquitto MQTT Broker installation for macOS

- ❖ Open the Terminal Application
  - Begin by launching the Terminal application on your Mac.
- ❖ Installation Using 'Homebrew'
  - To install the MQTT Broker, utilize the 'homebrew' project. Navigate to your **user profile path** and install Mosquitto broker with the following command: (Do not install in any other directory, apart from 'user profile' path.)
  - Execute the following command in the Terminal: **brew install mosquitto**.



- ❖ Having installed both Miniconda software and the MQTT broker protocol, we'll now proceed to create and activate the **Conda** environment, tailored to your operating system.
- ❖ Please follow the steps outlined in the upcoming slides:



# Commencing Operations on Diverse Operating Systems

## Windows:

- Open the Meritus-CVPRO-Windows (Extracted folder) and open the path in Windows-Command-Prompt, by typing 'cmd' in the Address Bar or Breadcrumb Bar.

(or)

- Press 'Win + R' which opens the "Run" dialog, then type 'cmd' and press 'Enter'. Now execute the command as follows:

**C:\xxxx\xxxx>cd Meritus-CVPRO-Windows**

- When you are in the command prompt window, enter the command '**cvpro**'. Please note that, throughout the process, this will be our ***working terminal***.

## Linux:

- Right Click the Meritus-CVPRO-Linux, (extracted folder) and select 'Open in Terminal'. This will open a new Terminal with folder path.

(or)

- Open a new Terminal, and type:

```
xxxx@xxxx: $ cd Meritus-CVPRO-Linux
```

- When you are in the terminal, enter the command 'source cvpro.sh' or '. cvpro.sh'. Please note that, throughout the process, this will be our ***working terminal***.

## macOS:

- Right-click the Meritus-CVPRO-Mac (extracted folder) and select 'New Terminal at Folder'.

(or)

- Open a Terminal app and type:

```
xxxx@xxxx: % cd Meritus-CVPRO-Mac
```

- Execute the command **source cvpro.zsh**. Please note that, throughout the process, this will be our ***working terminal***.

- ❖ The '**cvpro**' menu options are as follows:
- '**my\_conda**' – to create or activate the conda environment.
  - '**install\_cvpro**' – to install the required libraries.
  - '**launch\_server**' – to start MQTT server. (ensure that CV Pro kit is connected with your computer)
  - '**run\_cvpro**' – to collect the data by enabling the cv pro kit to navigate around the track..
  - '**train\_cvpro**' – to train the kit for generating the machine learning model.

Refer to the figure (4) given in the next slide.

## Menu options under 'cvpro':

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\robot\Meritus-CVPRO-Windows>cvpro

Pre-Requisites:
-----
1. Download the Meritus-CVPRO from https://github.com/robotixdevteam/Meritus-CVPRO/tree/Windows and extract the same to the User-Profile Path
2. Miniconda should be installed in the User-Profile Path
3. Mosquitto should be installed in C:\Program Files

Hierarchy of Execution:
-----
Type 'my_conda' - Create or Activate the Conda Environment
Type 'install_cvpro' - Installation of required libraries
-----
Please ensure to connect the Bot with the System
Type 'launch_server' - Launch the MQTT Server
-----
Type 'run_cvpro' to move the Bot around for Data-Collection Process.
Type 'train_cvpro' to train the Bot for Autonomous Process.

C:\Users\robot\Meritus-CVPRO-Windows>
```

Figure (4)

# Creating & Activating Environment

## Creating work environment - From Miniconda prompt – for Windows OS

- Begin by entering the following command, '**my\_conda**' in the working terminal, to create or activate the conda environment.
- When you enter the option, you will be prompted with three more options, '**build**', '**activate**' and '**deactivate**'.



## Creating work environment - From Miniconda prompt – for Windows OS

- **'build'** – This option will **create and activate** a virtual environment.
- **'activate'** – This will **activate** the environment created. ([when required](#))
- **'deactivate'** – This will **deactivate** the environment created. ([when required](#))

*Refer to the figures (5 & 6) given in the next slide.*

## Creating work environment - From Miniconda prompt – for Windows OS

**Step 1:** Enter the command '**my\_conda**',  
in the working terminal.

```
C:\Users\robot\Meritus-CVPRO-Windows>my_conda
Hi! Please Ensure Internet Connection!
Give me a command - 'build' or 'activate' or 'deactivate':
```

Figure 5

**Step 2:** Enter the command '**build**' to  
create and activate the environment.

```
C:\Users\robot\Meritus-CVPRO-Windows>my_conda
Hi! Please Ensure Internet Connection!
Give me a command - 'build' or 'activate' or 'deactivate': build
Creating a Virtual Environment:- cvpro ...
Retrieving notices: ...working... done
```

Figure 6

*If the environment is not activated follow the troubleshooting procedure given here. <<troubleshooting>>*

## Activating work environment

- Enter the command, '**my\_conda**' and from the options, displayed enter '**activate**'.
- This will activate the existing environment, allowing us to commence either the '**data collection**' or the '**training process**'. Subsequent instructions will be provided within the command prompt.

```
C:\Users\robot\Meritus-CVPRO-Windows>my_conda
Hi! Please Ensure Internet Connection!
Give me a command - 'build' or 'activate' or 'deactivate': activate
cvpro Environment is Activated...

For Data Collection Process:
-----
1. Please Disconnect the Internet, and connect the Bot with the System. Ensure the Connected IPv4 Address is: 192.168.4.2
2. Type 'launch_server' to Launch the MQTT Server to run the Bot!
3. Type 'run_cvpro' to begin the Data Collection Process!

For Training Process:
-----
1. Ensure your Bot is disconnected from the System as well as the Application.
2. Type 'train_cvpro' to begin the Training Process!

(cvpro) C:\Users\robot\Meritus-CVPRO-Windows>
```

Figure 7

*Refer to the figure (7) given.*

## Activating work environment – Required cases

Activate the environment under the following circumstances:

- **Scenario1:** You have given the 'build' command and subsequently '**cvpro**' environment is created successfully.
- **Scenario2:** You already have an environment and **installed** all the required **python libraries** in it.
- **Scenario3:** In the cases when you are **not satisfied** with the **existing model** and wish to **regenerate** the **model**.
- **Scenario4:** In the case where you wish to **redo** the **entire process** with the existing environment, '**cvpro**'.

## De-activating work environment – Required cases

- Deactivate the environment only if absolutely necessary. If you need to deactivate, enter the command, **'my\_conda'** and in the option, enter **'deactivate'**.

```
C:\Users\robot\Meritus-CVPRO-Windows>my_conda
Hi! Please Ensure Internet Connection!
Give me a command - 'build' or 'activate' or 'deactivate': deactivate

cvpro Environment is De-Activated

(base) C:\Users\robot\Meritus-CVPRO-Windows>
```

Figure 8

*Refer to the figure (8) given.*

# Installing Python Libraries

- ❖ We have now created and activated the conda environment, '**cvpro**', as shown in the figure (9).
- ❖ The next task is to install the dependencies.

```
Building the Environment...  
Build is Successful.  
  
Activating the Environment  
cvpro Environment is Built and Activated...  
  
Now you are ready for Installation Process.  
  
Type 'install_cvpro' to Install the Dependencies  
(cvpro) C:\Users\robot\Meritus-CVPRO-Windows>
```

**Figure 9**

- ❖ To install the dependencies, execute the command, '**install\_cvpro**', in the working terminal. A prompt will appear with options 'Y' or 'N'. Select 'Y', only in the presence of a stable internet connection. Refer to figure (10) given below.

```
Building the Environment...
Build is Successful.

Activating the Environment
cvpro Environment is Built and Activated...

Now you are ready for Installation Process.

Type 'install_cvpro' to Install the Dependencies

(cvpro) C:\Users\robot\Meritus-CVPRO-Windows>install_cvpro
Do you have a valid Internet Connection? - 'y' or 'n': y
Installing dependencies.....
Requirement already satisfied: pip in c:\users\robot\miniconda3\envs\cvpro\lib\site-packages (23.3.1)
Collecting pip
  Using cached pip-24.0-py3-none-any.whl.metadata (3.6 kB)
```

**Figure 10**



- 
- The screenshot shows a terminal window with the following content:
- ```
Installation is Completed!

For Data Collection Process:
-----

1. Please Disconnect the Internet, and connect the Bot with the System. Ensure the Connected IPv4 Address is: 192.168.4.2
2. Type 'launch_server' to Launch the MQTT Server to run the Bot!
3. Type 'run_cvpro' to begin the Data Collection Process!

For Training Process:
-----

1. Ensure your bot is disconnected from the System as well as the Application.
2. Type 'train_cvpro' to begin the Training Process!
```
- A red arrow points from the "Training Process" section to the bottom of the terminal window.
- ```
CV-Pro v0.1.0
Copyright © 2023 CV-Pro. All rights reserved.
CV-Pro is a free software distributed under the terms of the GNU General Public License v3.0.
See https://www.gnu.org/licenses/gpl-3.0.html for more details.

Usage: cvpro [options] command

Options:
  -h, --help            show this help message and exit
  -v, --version          show program version number and exit
  -c, --config FILENAME config file path
  -d, --device DEVICE_ID device ID (e.g., /dev/ttyUSB0)
  -i, --ip IP_ADDRESS    system IP address (e.g., 192.168.4.2)
  -p, --port PORT        system port (e.g., 8080)
  -m, --mode MODE        operation mode (e.g., launch_server, run_cvpro, train_cvpro)
  -s, --server SERVER_IP server IP address (e.g., 192.168.4.2)
  -S, --server_port SERVER_PORT server port (e.g., 8080)
  -t, --topic TOPIC       MQTT topic name (e.g., cvpro_data)
  -T, --topic_port TOPIC_PORT topic port (e.g., 1883)
  -u, --url URL           MQTT broker URL (e.g., mqtt://192.168.4.2:1883)
  -U, --username USERNAME MQTT username (e.g., user)
  -P, --password PASSWORD MQTT password (e.g., password)
  -k, --key KEY_PATH      private key file path (e.g., /etc/cvpro/key.pem)
  -K, --cert CERT_PATH    certificate file path (e.g., /etc/cvpro/cert.pem)
  -C, --ca CA_PATH         CA certificate file path (e.g., /etc/cvpro/ca.pem)
  -n, --name NAME         bot name (e.g., cvpro_bot)
  -N, --namespace NAMESPACE namespace name (e.g., cvpro)
  -o, --output OUTPUT_PATH output file path (e.g., /tmp/output.txt)
  -O, --output_format OUTPUT_FORMAT output format (e.g., json, csv)
  -f, --force             force to overwrite existing files
  -F, --file FILE_PATH    file path (e.g., /tmp/output.txt)
  -M, --mqtt_url MQTT_URL MQTT broker URL (e.g., mqtt://192.168.4.2:1883)
  -M_USERNAME, --mqtt_username MQTT_USERNAME MQTT username (e.g., user)
  -M_PASSWORD, --mqtt_password MQTT_PASSWORD MQTT password (e.g., password)
  -M_KEY, --mqtt_key MQTT_KEY_PATH private key file path (e.g., /etc/cvpro/key.pem)
  -M_CERT, --mqtt_cert MQTT_CERT_PATH certificate file path (e.g., /etc/cvpro/cert.pem)
  -M_CA, --mqtt_ca MQTT_CA_PATH CA certificate file path (e.g., /etc/cvpro/ca.pem)
  -M_NAME, --mqtt_name MQTT_NAME bot name (e.g., cvpro_bot)
  -M_NAMESPACE, --mqtt_namespace MQTT_NAMESPACE namespace name (e.g., cvpro)
  -M_OUTPUT, --mqtt_output MQTT_OUTPUT_PATH output file path (e.g., /tmp/output.txt)
  -M_OUTPUT_FORMAT, --mqtt_output_format MQTT_OUTPUT_FORMAT output format (e.g., json, csv)
  -M_FORCE, --mqtt_force mqtt_force force to overwrite existing files
  -M_FILE, --mqtt_file MQTT_FILE_PATH file path (e.g., /tmp/output.txt)
  -M_MQTT_URL, --mqtt_mqtt_url MQTT_MQTT_URL MQTT broker URL (e.g., mqtt://192.168.4.2:1883)
  -M_MQTT_USERNAME, --mqtt_mqtt_username MQTT_MQTT_USERNAME MQTT username (e.g., user)
  -M_MQTT_PASSWORD, --mqtt_mqtt_password MQTT_MQTT_PASSWORD MQTT password (e.g., password)
  -M_MQTT_KEY, --mqtt_mqtt_key MQTT_MQTT_KEY_PATH private key file path (e.g., /etc/cvpro/key.pem)
  -M_MQTT_CERT, --mqtt_mqtt_cert MQTT_MQTT_CERT_PATH certificate file path (e.g., /etc/cvpro/cert.pem)
  -M_MQTT_CA, --mqtt_mqtt_ca MQTT_MQTT_CA_PATH CA certificate file path (e.g., /etc/cvpro/ca.pem)
  -M_MQTT_NAME, --mqtt_mqtt_name MQTT_MQTT_NAME bot name (e.g., cvpro_bot)
  -M_MQTT_NAMESPACE, --mqtt_mqtt_namespace MQTT_MQTT_NAMESPACE namespace name (e.g., cvpro)
  -M_MQTT_OUTPUT, --mqtt_mqtt_output MQTT_MQTT_OUTPUT_PATH output file path (e.g., /tmp/output.txt)
  -M_MQTT_OUTPUT_FORMAT, --mqtt_mqtt_output_format MQTT_MQTT_OUTPUT_FORMAT output format (e.g., json, csv)
  -M_MQTT_FORCE, --mqtt_mqtt_force mqtt_mqtt_force force to overwrite existing files
  -M_MQTT_FILE, --mqtt_mqtt_file MQTT_MQTT_FILE_PATH file path (e.g., /tmp/output.txt)

Commands:
  launch_server          Launch the MQTT Server to run the Bot!
  run_cvpro              Run the Data Collection Process!
  train_cvpro            Train the Model!

cvpro: I have created the CV-Pro based on the CV-Pro was disconnected status.
```

### Figure 11

# **Installing CV Pro Mobile Application**

## For Android Smartphone Users

- ❖ Scan QR-Code or Open the Google Play Store on your Android device.
- ❖ Search for the 'CV Pro' application.
- ❖ From the search results, select the 'CV Pro' app and tap on 'Install'.
- ❖ Review the requested permissions and tap 'Accept'.
- ❖ Once installed, locate the app icon either on your home screen or within the app drawer and launch it.
- ❖ Find the app icon on your home screen, and then launch the app.



## For iOS Phone Users

- ❖ Scan QR-Code or Access the App Store on your iOS device.
- ❖ Search for the 'CV Pro' application.
- ❖ Locate the 'CV Pro' app from the search results and tap 'Get'.
- ❖ If prompted, authenticate using your Apple ID or use Touch ID/Face ID to confirm.
- ❖ Allow the app to download.
- ❖ Find the app icon on your home screen, and then launch the app.



*Thank  
you!*