Quick Startup

DASS Team-33

IIIT Hyderabad - April 18, 2024

Contents

1 Frontend

Follow the below steps for installing the Frontend on Operating System. It is recommended to use Ubuntu or any Debian Machine.

1.1 Setting Up Environment

Execute each steps in the given order to setup the environment for running the Frontend.

1.1.1 For Windows

1. Go to Node js Download Page and download MSI or any other package according to your system requirements(32 bit /64 bit).

2. Once Downloaded, go to the Downloaded location and open the file; Follow the instructions on the screen. Once it is done, you have to restart your PC.

3. Verify the installation by typing the following commands in the command prompt or PowerShell.

   ```
   node -v
   npm -v
   ```

   Both the commands should display the currently running version of node in your Local Machine. If it didn't, please re-install.

4. Go to Expo website SignUp and Register new account and using those credential login on command prompt.

5. Now go to "Frontend" directory. Search the file `bat.bat`.

6. Open command prompt or powershell and run `bat.bat`. Ensure you are in same directory.

7. After successful run the command prompt will show username and password for expo login. Enter the username and password generated earlier.

1.1.2 For Linux (Ubuntu or Debian)

1. Open terminal and run the command `sudo apt install nodejs npm`.

2. Verify the installation by typing the following commands in the command prompt or PowerShell.

   ```
   node -v
   npm -v
   ```

   Both the commands should display the currently running version of node in your Local Machine. If it didn't, please re-install.

3. Go to Expo website SignUp and Register new account and using those credential login on command prompt.

4. Now go to "Frontend" directory. Search the file `bat.sh`.

5. Open terminal and run `.\bat.sh`. Ensure you are in same directory.
6. After successful run the command prompt will show username and password for expo login. Enter the username and password generated earlier.

1.2 Files

- **App.js:** The central component in the code which collects all the individual screens in our app and render them through a stack Navigator.
- **Constants.js:** Contains all the environment variables for our app. All the changes should be performed only in this file. (host and port changes)
- **Home.js:** The home screen component for the app is implemented in this file. All the functionalities of the homepage like copy, delete, scan, etc are initiated from here.
- **Detected_Images.js and ImageView.js:** Both these files are responsible for rendering the screen containing the list of detected items in a scrollView.
- **Input_order.js:** This file is responsible for rendering the screen which takes the order input(item and quantity) from the user.
- **Settings.js:** This file is responsible for the "editable option functionality" which allows the user to change the ability of the user to edit the detected quantities.
- **Components and Colours.js:** All files for storing all the color codes used in the app.
- **newSummary.js:** This File is responsible for displaying the intermediate detections by using the data given by the backend model. The functionalities like delete, save, AddImage are also implemented here.
- **orderinfo.js:** This file is responsible for displaying the order details screen which shows up on clicking a particular order from the home Screen. Please note that the new scan functionality is implemented here in the app.
- **app.json, package.json, package-lock.json, babel.config.js and .expo:** All these files are responsible for compiling and monitoring the working of the app which are managed by the expo. Never delete these files.
- **Camera.js:** Displays Camera with Recording and Picturing option. To start Recording single tap the circle in the bottom and do the same to stop the recording. Long press the circle to capture a picture.
- **Processing.js:** Send the captured picture or video to the host mentioned in **Constants.js** to the route /imagedetect or /videodetect depending on the MIME type.
- **Detections.js:** Acts like a navigation page between the Camera and Processing page.
- **Network.js:** Checks the availability of the Internet while sending the request to the server.

2 Backend

2.1 Setting Up Environment

Note: Follow this Installation process only if you want to run the code in your local machine(localhost). You can skip this if you are using an AWS server refer to Contents

2.1.1 Python3 Installation

1. Update the package list:
   ```
   sudo apt update
   ```
2. Install necessary packages and add the DeadSnakes PPA:
   ```
   sudo apt install software-properties-common
   sudo add-apt-repository ppa:deadsnakes/ppa
   sudo apt update
   ```
3. Install Python3.8:
   ```
   sudo apt install python3.8
   ```
4. Verify the Python version:
   ```
   python --version
   ```

2.1.2 Virtual Environment Installation

1. Update the package list:
   ```
   sudo apt update
   ```
2. Install Python3 virtual environment:
   ```
   sudo apt install python3-venv
   ```

2.1.3 pip and pip3 Installation

1. Update the package list:
   ```
   sudo apt update
   ```
2. Install pip and pip3:
   ```
   sudo apt install python3-pip
   pip3 --version
   ```

2.1.4 Python Packages Installation

1. Create a virtual environment:
   ```
   python3 -m venv environmentname
   ```
2. Activate the environment:
   ```
   source environmentname/bin/activate
   ```
3. Install Flask:
   ```
   pip install Flask
   ```
4. Install pymongo:
   ```
   pip install pymongo
   ```

5. Install numpy:
   ```
   pip3 install numpy
   ```

6. Install opencv-python:
   ```
   pip install opencv-python
   ```

7. Install Pillow:
   ```
   pip install Pillow
   ```

8. Install sk-video:
   ```
   pip install sk-video
   ```

9. Install matplotlib:
   ```
   pip install matplotlib
   ```

10. Install torchvision:
    ```
    pip install --no-cache-dir torchvision
    ```

11. Install yacs:
    ```
    pip install yacs
    ```

12. Install scikit-learn, scipy, matplotlib:
    ```
    pip3 install -U scikit-learn scipy matplotlib
    ```

13. Install scikit-image:
    ```
    pip install scikit-image
    ```

14. Install Flask-Cors:
    ```
    pip install Flask-Cors
    ```

15. Install python3-opencv package:
    ```
    sudo apt-get install python3-opencv
    ```

16. Deactivate the environment:
    ```
    deactivate
    ```

2.1.5 MongoDB Installation

1. Follow the MongoDB official documentation to install MongoDB on your localhost.

2. Enable MongoDB on system startup:
   ```
   sudo systemctl enable mongod
   ```

3. Test MongoDB installation by running:
   ```
   sudo mongo
   ```

2.2 Files

- **data/demo**: Location of the folder where one needs to insert class images in the format of
  `raw_inputs/img1.png`,
  `raw_inputs/img2.png`,
  `raw_inputs/img1.png`, etc.

  Also update in **Manual** and **Y_Train** set in both **demo_image.py** and **demo_video.py**. Ensure that for each train image we have manual image.

- **app.py**: Main Flask server file that contains all routes of the server.
- **demo.py**: A demo file for explaining the working of the model, inputs class images and file images, and outputs `0.png` in the root folder (i.e., backend), but this has nothing to do with **app.py** that contains routes that process the model. For this purpose, we designed two special files which will be discussed below.
- **demo_image.py**: A modified version of **demo.py** which contains a function for processing image files.
- **demo_video.py**: A modified version of **demo.py** which contains a function for processing video files.
- **osd/utils/visualization.py**: This contains some utility functions needed for the above mentioned **demo** files to run. We suggest not to edit it, but sometimes if you need to know about how the detected image is showing labels, ids, percentages, or need to change the labels this is the place to do it.

3 Running on Local Machine

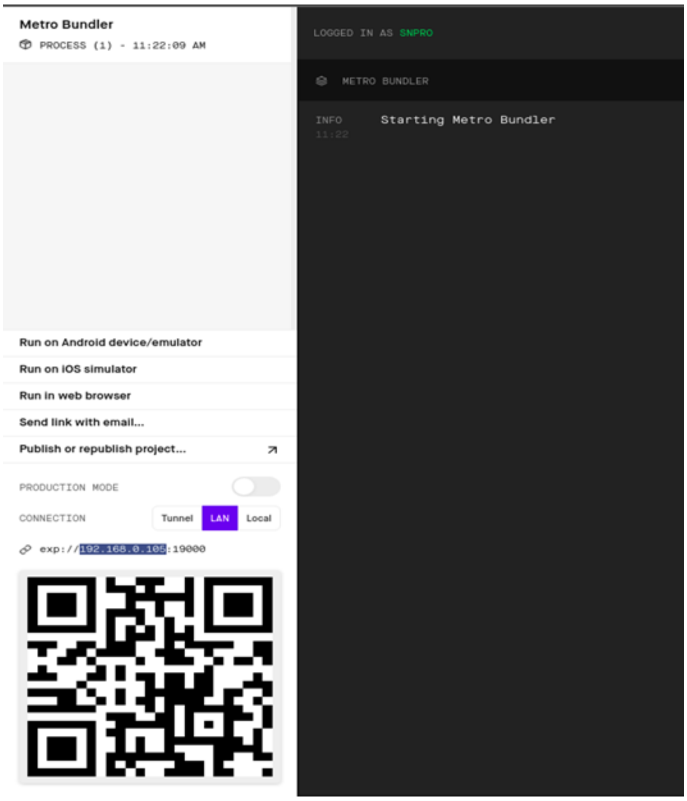
3.1 Frontend

- Navigate to the frontend directory and if you don't see node_modules execute steps from For Windows

  The node_modules folder in the frontend directory contains the code for all the dependencies. Follow above procedure every time one switches the system.
- Once all the dependencies are installed, you can start the metro bundler by running the command:

  ```
  expo start
  ```

  In a couple of seconds, this opens up the page shown in the below diagram.
- You can see the address 192.165.0.105 where your bundler is running. Make a note of this address which is the wifi ip address. This address will be different in your machine.
- Now install expo go in your android or iphone for play store or app store. Connect the phone to the same wifi as the computer and scan the QR code from the expo app (for android) and for iphone use Open Camera and click on link visible at bottom of frame.



Metro Bundler

PROCESS (1) - 11:22:09 AM

LOGGED IN AS SNPRO

METRO BUNDLER

```
INFO     Starting Metro Bundler
11:22
```

exp://192.168.0.105:19000

3.2 Backend

- Navigate to the backend directory and activate the virtual environment:

```
source <location_to_env>/bin/activate
```

- Connect both your mobile and laptop/PC to the same network.
- Once you have started the metro bundler in the frontend, you can see the host in which the expo app is running at the left bottom above the QR code in the metro bundler, as we have explained above in Fig1.
  - Copy that host.
- Now run the command on GPU:

```
flask run -h <copiedhost>
```

- Now run the command on CPU:

```
CUDA_VISIBLE_DEVICES="" flask run -h <copiedhost>
```

  That's it, now the backend server is running on the desired host (that host is nothing but the IPv4 of your network).

- Now open `Constants.js`, you can see some variables header, host, port. Set the host to the `<copiedhost>` and port to 5000.

4 Running Without Dependency

4.1 Frontend

- Navigate to the frontend directory and if you don't see node_modules execute steps from For Windows

  The node_modules folder in the frontend directory contains the code for all the dependencies. Follow above procedure every time one switches the system.

- Once all the dependencies are installed, you can start the metro bundler by running the command:

  ```
  npx expo start --tunnel
  ```

  In a couple of seconds, the qr code will be visible.

- Now install expo go in your android or iphone for play store or app store. Connect the phone to the same wifi as the computer and scan the QR code from the expo app (for android) and for iphone use Open Camera and click on link visible at bottom of frame.

4.2 Backend

Link the frontend to the deployed backend on AWS instead of the localhost backend, by define the host as the public IPv4 address of the deployed Amazon EC2 instance and the port as 80 in ./frontend/Constants.js

For activating the AWS backend server refer to **AWS Deployment Documentation**.