DroneFlote Deployment Guide

### Foreword

This project utilizes packages and modules from several other independent projects such as; *Tensorflow* and *posenet* for image and pose detection.

*Docker and docker-compose* for containerization, deployment and isolating individual components for cohesion.

*Gazebo* for simulating our drones in a safe test environment and for testing purposes.

*CrazyS* and *Mavcomm* which are both git repositories that enable us to model and fly a simulated drone modeled after the drones used in practice.

Together with these we have integrated a system where the pose of a user can dynamically change how a simulated drone will fly.

# Installation requirements

Since we have built this project with the aim to separate each of its features for ease of deployment, dockers and containers are the main parts of the running system I will cover how to install Docker first.

## Git

This project resides on github where you will find all the relevant files required to deploy DroneFlote. As such we need to download and install *git* so we can retrieve a copy remotely to deploy.

On Windows we need to install git gui, which can be downloaded and installed here:

<https://git-scm.com/downloads>

The default components selected by the installer are fine but the option called “**configuring the line ending conversions”** must be set to *“checkout as-is, commit as-is”* or the project will not deploy on windows. This is the only change you are required to make during installation of git. All other options may be left at the default selected choice.

On linux, installing git from your package manager is more than enough.

## Docker and Compose

Docker is a solution for creating containers to house applications or parts therein to ease development and deployment, with containers being a low cost solution to deploying individual environments or operating systems as opposed to virtual machines.

### Windows and Mac

From what I understand, the installation for both systems is somewhat similar but this will guide you through setting it up on windows specifically.

1. First we need to install docker. The current version from windows can be downloaded from the page:

<https://www.docker.com/get-started>

1. Run the executable once downloaded and make sure all the configuration options are checked.
2. On Windows, Docker relies on a windows feature called wsl2 which must be installed for docker to run. Information about installing this can be found here:

<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

Steps 6 onward may not be necessary.

1. Once the install is complete, restart your PC before continuing.
2. Luckily Docker-Compose comes with Docker desktop so we can stop here.

### Linux

Both docker and compose can be either installed via the package manager or manually for many versions of linux.

For ease of use I suggest you either install from the package manager or follow the instructions on dockers website:

<https://docs.docker.com/engine/install/>

The same follows for docker-compose.

# Building

This is a two part process. First we need to get the project code, then build it into containers before we can then deploy the project as a whole.

Provided you are not updating anything and have already completed these first two steps, to deploy again we need to follow the deployment instructions.

If you do download an updated version you will have to build it again before you run it.

### Get the repository

First open up a terminal window or powershell window if you’re on windows and navigate to a directory you would like the source code to live.

*For windows users, the easiest way to do this is to open the file explorer and click through to the directory you choose, then click on the address bar (shown below), delete the text, type “powershell” without quotes and hit enter. This opens the powershell in the directory you were in.*



Now we will clone the repository into this folder, creating a new folder containing our project.

* git clone https://github.com/hupsuni/DroneFlote-Capstone.git

We now have a local copy of DroneFlote. Now enter that directory from the terminal and proceed.

* cd DroneFlote-Capstone

### Build the project

With a terminal open inside the newly downloaded directory we are going to have docker compose build all of our containers for us.

* docker-compose build

The compose file simplifies the build process extensively from a user perspective but it can take quite some time so let it finish building before proceeding.

# Deployment

Making use of the docker compose feature greatly uncomplicates deployment and running of all the involved containers.While each container can be run individually we can deploy the whole project from docker compose.

To deploy and run the project, once it has finished building, simply enter this command in the terminal:

* docker-compose up -d

It may take a minute or two to run all the containers and several minutes more the first time it is deployed.

To shut down the project and its containers use the command:

* docker-compose down

The project contains seleval docker images that will all run and communicate with each other.

One of these containers holds the pose recognition over camera application on a webserver accessible by going to the following address:

* <http://localhost:8888/>

There are two options to select from here.

The camera demo will use a live feed from your connected camera to map key points to coordinates.

The image demo uses preloaded static images to do the same and does not require a camera.

A gazebo simulator with a simulated drone is also run in another container that can be accessed through a secondary container running vnc by going to the following address:

* <http://localhost:8080/vnc.html>

This container simply displays the running simulation in real time through the web browser.