Docker Container Document

By: Parin Rajesh Jhaveri
Tejas Chheda
Sharmishtha Gupta

Docker Containerization:

- We are containerizing this model inference server in a Docker container.
- To be able to build this Docker container we first designed a Dockerfile that would build an image when executed. This Docker image is then leveraged to deploy the Docker Container when run.
- We used *python:3.7-slim* as the base image as it is lightweight as compared to *Buster* and more compatible than the *alpine* version.
- We have made */app* our working directory and copied our project there as well as the images directory
- We then install Flask and the CPU version of PyTorch.
- We expose the 5000 port on Docker as the HTTP Server runs on that port and to be able to accept requests from outside the container we need to expose that particular port
- The command that will be triggered when the docker container is *python3 main.py*. This will start the HTTP Server on port 5000.
- The docker shell script *run_docker.sh* contains the commands required to build the Dockerfile and then run the container with the image that is built - *flask-docker-base-app.*

How to Execute:

Steps to create and run docker containers (ensure that the docker daemon is running):

1. Run the shell script `run_docker.sh` using the command `sh run_docker.sh`. The shell script automates the build of the Docker Image and then Runs this image on a container by running the Docker Build command followed by the Docker Run Command.
2. Check to see if server is up and running at `0.0.0.0:5000/` or `localhost:5000/` by typing this in your browser

Steps to check inference on images (once docker is running and previous step is executed):
1. Download the image to run inference on (eg. dog.jpg) and store it in the */images* directory
2. Run the command `curl http://localhost:5000/inference/dog.jpg` to get the prediction (class of object in the image)