Lab 4 Tutorial

This tutorial will help you create special images (adversarial patches) that can trick a pretrained neural network, specifically yolov2, on a Raspberry Pi.

Step 1: Set up the project

- **Download the project files:**
  - Visit https://gitlab.com/EAVISE/adversarial-yolo and download the project. Use the instructions in the Readme file to learn how to train your patch on a GPU or CPU.

- **Install necessary libraries and get the yolov2 model weights**
  - pip install tensorboardX tensorboard
  - mkdir weights; curl https://pjreddie.com/media/files/yolov2.weights -o weights/yolo.weights

Step 2: Get the INRIA dataset:

- Use these commands to download and set up the dataset:
  - curl ftp://ftp.inrialpes.fr/pub/lear/douze/data/INRIAPerson.tar -o inria.tar
  - tar xf inria.tar
  - mv INRIAPerson inria
  - cp -r yolo-labels inria/Train/pos/

Step 3: Create your patch

- **Run the adversarial patch generation algorithm and obtain your own patch**: python train_patch.py paper_obj
- Make sure to read the code and the related paper to understand how it works.

Step 4: **Print the patch**

Please note that the patch must be printed in color. If the patch size is not enough, you can use two sheets combined (i.e., print half on each sheet) to see the performance of your own patch on yolov2 model.

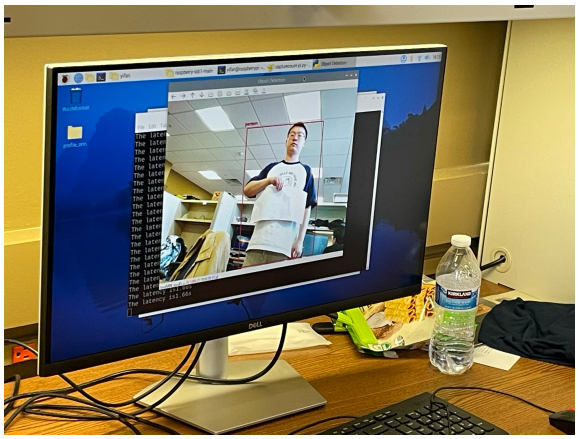
Step 5: **Follow the steps of lab1 to open the object detection project.**

- cd yolo
- curl https://pjreddie.com/media/files/yolov2.weights -o yolov2.weights
- curl https://raw.githubusercontent.com/pjreddie/darknet/master/cfg/yolov2.cfg -o yolov2.cfg
- cd ../
- python capturecount-pi.py

- Run the object detection script on the Pi with python capturecount-pi.py. Remember to adjust the model name and configuration in the capturecount-pi.py file as needed.

Step 6: **Test your patch.**

Without patch:



With patch: